DART: Diversify-Aggregate-Repeat Training Improves Generalization of Neural Networks

Samyak Jain * ◇ ‡ Sravanti Addepalli *
Pawan Kumar Sahu ∓ § ‡ Priyam Dey ∓ R.Venkatesh Babu
Vision and AI Lab, Indian Institute of Science, Bangalore

Abstract

Generalization of Neural Networks is crucial for deploying them safely in the real world. Common training strategies to improve generalization involve the use of data augmentations, ensembling and model averaging. In this work, we first establish a surprisingly simple but strong benchmark for generalization which utilizes diverse augmentations within a training minibatch, and show that this can learn a more balanced distribution of features. Further, we propose Diversify-Aggregate-Repeat Training (DART) strategy that first trains diverse models using different augmentations (or domains) to explore the loss basin, and further Aggregates their weights to combine their expertise and obtain improved generalization. We find that Repeating the step of Aggregation throughout training improves the overall optimization trajectory and also ensures that the individual models have sufficiently low loss barrier to obtain improved generalization on combining them. We theoretically justify the proposed approach and show that it indeed generalizes better. In addition to improvements in In-Domain generalization, we demonstrate SOTA performance on the Domain Generalization benchmarks in the popular DomainBed framework as well. Our method is generic and can easily be integrated with several base training algorithms to achieve performance gains. Our code is available here: `https://github.com/val-iisc/DART`.

1. Introduction

Deep Neural Networks have outperformed classical methods in several fields and applications owing to their remarkable generalization. Classical Machine Learning theory assumes that test data is sampled from the same distribution as train data. This is referred to as the problem of In-Domain (ID) generalization [15, 18, 29, 32, 48], where the goal of the model is to generalize to samples within same domain as the train dataset. This is often considered to be one of the most important requirements and criteria to evaluate models. However, in several cases, the test distribution may be different from the train distribution. For example, surveillance systems are expected to work well at all times of the day, under different lighting conditions and when there are occlusions, although it may not be possible to train models using data from all these distributions. It is thus crucial to train models that are robust to distribution shifts, i.e., with better Out-of-Domain (OOD) Generalization [25]. In this work, we consider the problems of In-Domain generalization and Out-of-Domain Generalization of Deep Networks. For the latter, we consider the popular setting of Domain Generalization [4, 23, 41], where the training data is composed of several source domains and the goal is to generalize to an unseen target domain.

The problem of generalization is closely related to the Simplicity Bias of Neural Networks, due to which models have a tendency to rely on simpler features that are often spurious correlations to the labels, when compared to the harder robust features [55]. For example, models tend to rely on weak features such as background, rather than more robust features such as shape, causing a drop in object classification accuracy when background changes [22, 72]. A common strategy to alleviate this is to use data augmentations [8–10, 27, 42, 53, 75, 77] or data from several domains during training [23], which can result in invariance to several spurious correlations, improving the generalization of models. Shen et al. [57] show that data augmentations enable the model to give higher importance to harder-to-learn robust features by delaying the learning of spurious features. We extend their observation by showing that training on a combination of several augmentation strategies (which we refer to as *Mixed* augmentation) can result in the learning of a balanced distribution of diverse features. Using this, we obtain a strong benchmark for ID generalization as shown in Table-1. However, as shown in prior works [1], the impact of augmentations in training is limited by the capacity of the network in being able to generalize well to

*Equal Contribution. ∓ Equal contribution second authors. Correspondence to Samyak Jain <samyakjain.cse18@itbhu.ac.in>, Sravanti Addepalli <sravantia@iisc.ac.in>. ◇ Indian Institute of Technology, Varanasi § Indian Institute of Technology, Dhanbad. ‡ Work done during internship at Vision and AI Lab, Indian Institute of Science, Bangalore.

Table 1. **Motivation:** Performance (%) on CIFAR100, ResNet-18 with ERM training for 200 epochs. Mixed-Training (MT) outperforms individual augmentations, and ensembles perform best.

	Test Augmentation			
Train Augmentation	No Aug.	Cutout	Cutmix	AutoAugment
Pad+Crop+HFlip (PC)	78.51	67.04	56.52	58.33
Cutout (CO)	77.99	74.58	56.12	58.47
Cutmix (CM)	80.54	74.05	**77.35**	61.23
AutoAugment (AA)	79.18	71.26	60.97	73.91
Mixed-Training (MT)	81.43	77.31	73.20	**74.73**
Ensemble (CM+CO+AA)	**83.61**	**79.19**	73.19	73.90

the diverse augmented data distribution. Therefore, increasing the diversity of training data demands the use of larger model capacities to achieve optimal performance. This demand for higher model capacity can be mitigated by training specialists on each kind of augmentation and ensembling their outputs [11,38,59,79], which results in improved performance as shown in Table-1. Another generic strategy that is known to improve generalization is model-weight averaging [31, 70, 71], which results in a flatter minima.

In this work, we aim to combine the benefits of the three strategies discussed above - diversification, specialization and model weight averaging, while also overcoming their individual shortcomings. We propose a **D**iversify-**A**ggregate-**R**epeat **T**raining strategy dubbed DART (Fig.1), that first trains M *Diverse* models after a few epochs of common training, and then *Aggregates* their weights to obtain a single generalized solution. The aggregated model is then used to reinitialize the M models which are further trained post aggregation. This process is *Repeated* over training to obtain improved generalization. The *Diversify* step allows models to explore the loss basin and specialize on a fixed set of features. The *Aggregate* (or Model Interpolation) step robustly combines these models, increasing the diversity of represented features while also suppressing spurious correlations. Repeating the *Diversify-Aggregate* steps over training ensures that the M diverse models remain in the same basin thereby permitting a fruitful combination of their weights. We justify our approach theoretically and empirically, and show that intermediate model aggregation also increases the learning time for spurious features, improving generalization. We present our key contributions below:

- We present a strong baseline termed Mixed-Training (MT) that uses a combination of diverse augmentations for different images in a training minibatch.
- We propose a novel algorithm DART, that learns specialized diverse models and aggregates their weights iteratively to improve generalization.
- We justify our method theoretically, and empirically on several In-Domain (CIFAR-10, CIFAR-100, ImageNet) and Domain Generalization (OfficeHome, PACS, VLCS, TerraIncognita, DomainNet) datasets.

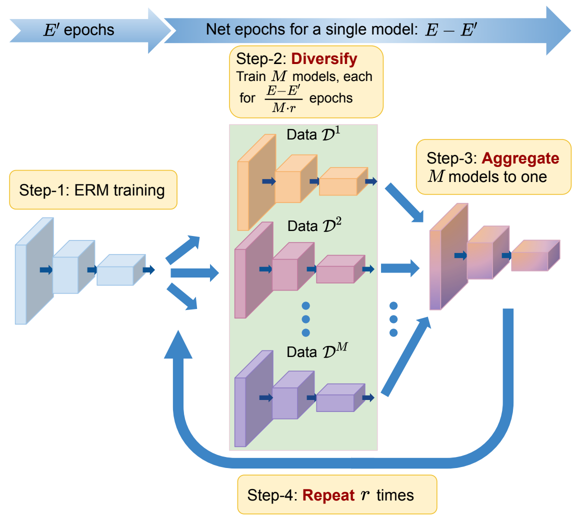

Figure 1. Schematic Diagram of the proposed method DART

2. Background: Mode Connectivity of Models

The overparameterization of Deep networks leads to the existence of multiple optimal solutions to any given loss function [33, 45, 76]. Prior works [14, 21, 46] have shown that all such solutions learned by SGD lie on a non-linear manifold, and are connected to each other by a path of low loss. Frankle *et al*. [19] further showed that converged models that share a common initial optimization path are linearly connected with a low loss barrier. This is referred to as the *linear mode connectivity* between the models. Several optimal solutions that are linearly connected to each other are said to belong to a common *basin* which is separated from other regions of the loss landscape with a higher *loss barrier*. Loss barrier between any two models θ_1 and θ_2 is defined as the maximum loss attained by the models, $\hat{\theta} = \alpha \cdot \theta_1 + (1 - \alpha) \cdot \theta_2 \quad \forall \quad \alpha \in [0, 1]$.

The linear mode connectivity of models facilitates the averaging of weights of different models in a common basin resulting in further gains. In this work, we leverage the linear mode connectivity of diverse models trained from a common initialization to improve generalization.

3. Related Works

3.1. Generalization of Deep Networks

Prior works aim to improve the generalization of Deep Networks by imposing invariances to several factors of variation. This is achieved by using data augmentations during training [8–10, 26, 42, 64, 75, 77], or by training on a combination of multiple domains in the Domain Generalization (DG) setting [7, 28, 30, 39, 41]. In DG, several works have focused on utilizing domain-specific features [3, 12], while others try to disentangle the features as domain-specific and domain-invariant for better generalization [6, 34, 39, 49, 67]. Data augmentation has also been exploited for Domain

Generalization [43, 51, 56, 58, 65, 66, 68, 73, 74, 81, 82] in order to increase the diversity of training data and simulate domain shift. Foret *et al.* [18] show that minimizing the maximum loss within an ℓ_2 norm ball of weights can result in a flatter minima thereby improving generalization. Gulrajani *et al.* [23] show that the simple strategy of ERM training on data from several source domains can indeed prove to be a very strong baseline for Domain Generalization. The authors also release DomainBed - which benchmarks several existing methods on some common datasets representing different types of distribution shifts. Recently, Cha *et al.* [5] propose MIRO, which introduces a Mutual-Information based regularizer to retain the superior generalization of the pre-trained initialization or Oracle, thereby demonstrating significant improvements on DG datasets. The proposed method DART achieves SOTA on the popular DG benchmarks and shows further improvements when used in conjunction with several other methods (Table-5) ascribing to its orthogonal nature.

3.2. Averaging model weights across training

Recent works have shown that converging to a flatter minima can lead to improved generalization [15, 18, 29, 32, 48, 60]. Exponential Moving Average (EMA) [50] and Stochastic Weight Averaging (SWA) [31] are often used to average the model weights across different training epochs so that the resulting model converges to a flatter minima, thus improving generalization at no extra training cost. Cha *et al.* [4] theoretically show that converging to a flatter minima results in a smaller domain generalization gap. The authors propose SWAD that overcomes the limitations of SWA in the Domain Generalization setting and combines several models in the optimal solution basin to obtain a flatter minima with better generalization. We demonstrate that our approach effectively integrates with EMA and SWAD for In-Domain and Domain Generalization settings respectively to obtain further performance gains (Tables-2, 4).

3.3. Averaging weights of fine-tuned models

While earlier works combined models generated from the same optimization trajectory, Tatro *et al.* [61] showed that for any two converged models with different random initializations, one can find a permutation of one of the models so that fine-tuning the interpolation of this with the second model leads to improved generalization. On a similar note, Zhao *et al.* [80] proposed to achieve robustness to backdoor attacks by fine-tuning the linear interpolation of pre-trained models. More recently, Wortsman *et al.* [71] proposed Model Soups and showed that in a transfer learning setup, fine-tuning and then averaging different models with same pre-trained initialization but with different hyperparameters such as learning rates, optimizers and augmentations can improve the generalization of the resulting model. The authors further note that this works best when

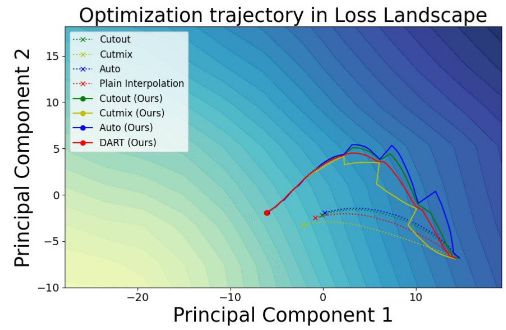

Figure 2. **Optimization trajectory** of the proposed approach DART when compared to independent ERM training on each augmentation. Axes represent the top two PCA directions obtained using the weights of DART training. The initial common point on the right represents the model obtained after 100 epochs of Mixed Training (MT). The trajectory shown is for an additional 100 epochs, with a total training budget of 200 epochs.

the pre-trained model is trained on a large heterogeneous dataset. While all these approaches work only in a fine-tuning setting, the proposed method incorporates the interpolation of differently trained models in the regime of *training from scratch*, allowing the learning of models for longer schedules and larger learning rates.

3.4. Averaging weights of differently trained models

Wortsman *et al.* [70] propose to average the weights of multiple models trained simultaneously with different random initializations by considering the loss of a combined model for optimization, while performing gradient updates on the individual models. Additionally, they minimize the cosine similarity between model weights to ensure that the models learned are diverse. While this training formulation does learn diverse connected models, it leads to individual models having sub-optimal accuracy (Table-2) since their loss is not optimized directly. DART overcomes such issues since the individual models are trained directly to optimize their respective classification losses. Moreover, the step of intermediate interpolation ensures that the individual models also have better performance when compared to the baseline of standard ERM training on the respective augmentations (Fig.4 in the Supplementary).

4. Proposed Method: DART

A series of observations from prior works [14, 19, 21, 46] have led to the conjecture that models trained independently with different initializations could be linearly connected with a low loss barrier, when different permutations of their weights are considered, suggesting that *all solutions effectively lie in a common basin* [16]. Motivated by these observations, we aim at designing an algorithm that explores the basin of solutions effectively with a robust optimization path and combines the expertise of several diverse models

Algorithm 1 Diversify-Aggregate-Repeat Training, DART

1: **Input:** M networks f_{θ^k} where $0 < k \leq M$, whose weights are aggregated every λ epochs. Training Dataset for each network f_{θ^k} is represented by $D^k = \{(x_i^k, y_i^k)\}$. The union of all datasets is denoted as D^*. Number of training epochs E, Maximum Learning Rate LR_{max}, Cross-entropy loss ℓ_{CE}. Model is trained using ERM for $E^{'}$ epochs initially.
2: **for** $epoch = 1$ **to** E **do**
3: $\quad \text{LR} = 0.5 \cdot \text{LR}_{max} \cdot (1 + cos((epoch - 1)/E \cdot \pi))$
4: $\quad$ **if** $epoch < E'$ **then**
5: $\quad\quad \theta = min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell_{CE}(\theta, D^*)$
6: $\quad$ **else**
7: $\quad\quad$ **if** $epoch = E'$ **then**
8: $\quad\quad\quad \theta^k \leftarrow \theta \;\; \forall k \in [1, M]$
9: $\quad\quad$ **end if**
10: $\quad\quad \theta^k = min_{\theta^k} \frac{1}{n} \sum_{i=1}^{n} \ell_{CE}(\theta, D^k) \; \forall k \in [1, M]$
11: $\quad\quad$ **if** $epoch \;\%\; \lambda = 0$ **then**
12: $\quad\quad\quad \theta = \frac{1}{M} \sum_{k=1}^{M} \theta^k$
13: $\quad\quad\quad \theta^k \leftarrow \theta \;\; \forall k \in [1, M]$
14: $\quad\quad$ **end if**
15: $\quad$ **end if**
16: **end for**

to obtain a single generalized solution.

We show an outline of the proposed approach - *Diversify-Aggregate-Repeat Training*, dubbed DART, in Fig.1. Broadly, the proposed approach is implemented in four steps - i) ERM training for E' epochs in the beginning, followed by ii) Training M *Diverse* models for λ/M epochs each, iii) *Aggregating* their weights, and finally iv) *Repeating* the steps *Diversify-Aggregate* for $E - E'$ epochs.

A cosine learning rate schedule is used for training the model for a total of E epochs with a maximum learning rate of LR_{max}. We present the implementation of DART in Algorithm-1, and discuss each step in detail below:

1. **Traversing to the *Basin* of optimal solutions:** Since the goal of the proposed approach is to explore the *basin* of optimal solutions, the first step is to traverse from a randomly initialized model upto the periphery of this basin. Towards this, the proposed *Mixed-Training* strategy discussed in Section-1 is performed on a combination of several augmentations D^* for the initial E' epochs (L4-L5 in Alg.1).
2. **Diversify - Exploring the *Basin*:** In this step, M diverse models f_{θ^k} initialized from the Mixed-Training model (L8 in Alg.1), are trained using the respective datasets D^k (L10 in Alg.1). These are generated using diverse augmentations in the In-Domain setting, and from a combination of different domains in the Domain Generalization setting. We set $|D^k| = |D|/M$ where D is the original dataset.
3. **Aggregate - Combining diverse experts:** Owing to the initial common training for E' epochs, the k diverse models lie in the same basin, enabling an effective aggregation of their weights using simple averaging (L12 in Alg.1) to obtain a more generalized solution θ. Aggregation is done after every λ epochs.
4. **Repeat:** Next, all k models are reinitialized using the common model θ (L13 of Alg.1), after which the individual models are trained for λ epochs on their respective datasets D^k as discussed in Step-2, and the process continues for a total of $E - E'$ epochs.

Visualizing the Optimization Trajectory: We compare the optimization trajectory of the proposed approach DART with independent training on the same augmentations in Fig.2 after a common training of $E' = 100$ epochs on Mixed augmentations. The models explore more in the initial phase of training, and lesser thereafter, which is a result of the cosine learning rate schedule and reducing gradient magnitudes over training. The exploration in the initial phase helps in increasing the diversity of models, thereby improving the robustness to spurious features (as shown in Proposition-3) leading to a better optimization trajectory, while the smaller steps towards the end help in retaining the flatter optima obtained after Aggregation. The process of repeated aggregation also ensures that the models remain close to each other, allowing longer training regimes.

5. Theoretical Results

We use the theoretical setup from Shen *et al*. [57] to show that the proposed approach DART achieves robustness to spurious features, thereby improving generalization.

Preliminaries and Setup: We consider a binary classification problem with two classes $\{-1, 1\}$. We assume that the dataset contains n inputs and K orthonormal robust features which are important for classification and are represented as $v_1, v_2, v_3, \ldots, v_K$, in decreasing order of their frequency in the dataset. Let each input example x be composed of two patches denoted as $(x_1, x_2) \in R^{d \times 2}$, where each patch is characterized as follows: i) **Feature patch**: $x_1 = yv_{k^*}$ where y is the target label of x and $k^* \in [1, K]$, ii) **Noisy patch**: $x_2 = \epsilon$ where $\epsilon \sim \mathcal{N}\left(0, \frac{\sigma^2}{d} I_d\right)$.

We consider a single layer convolutional neural network consisting of C channels, with $w = (w_1, w_2, w_3, \ldots, w_C) \in R^{d \times C}$. The function learned by the neural network (F) is given by $F(w, x) = \sum_{c=1}^{C} \sum_{p=1}^{2} \phi(w_c, x_p)$, where ϕ is the activation function as defined by Shen *et al*. [57].

Weights learned by an ERM trained model: Let K_{cut} denote the number of robust features learned by the model. Following Shen *et al*. [57], we assume the learned weights to be a linear combination of the two types of features present in the dataset as shown below:

$$w = \sum_{k=1}^{K_{cut}} v_k + \sum_{k>K_{cut}} y^{(k)} \epsilon^{(k)} \quad (1)$$

Data Augmentations: As defined by Shen *et al*. [57], an augmentation T_k can be defined as follows (K denotes the number of different robust patches in the dataset):

$$\forall\ k' \in [1, K], \quad \mathcal{T}_k(v_{k'}) = v_{((k'+k-1)\ mod\ K)+1} \quad (2)$$

Assuming unique augmentations for each of the m branches, the augmented data is defined as follows:

$$D_{train}^{(aug)} = D_{train} \cup \mathcal{T}_1(D_{train}) .. \cup \mathcal{T}_{m-1}(D_{train}) \quad (3)$$

where D_{train} is the training dataset. If $m = K$, each feature patch v_i appears n times in the dataset, thus making the distribution of all the feature patches uniform.

Weight Averaging in DART: In the proposed method, we consider that m models are being independently trained after which their weights are averaged as shown below:

$$w = \frac{1}{m} \sum_{j=1}^{m} \sum_{k=1}^{K_{cut_j}} v_{k_j} + \frac{1}{m} \sum_{j=1}^{m} \sum_{k>K_{cut_j}} y_j^{(k)} \epsilon_j^{(k)} \quad (4)$$

Each branch is trained on the dataset $D_{train}^{(k)}$ defined as:

$$D_{train}^{(k)} = \mathcal{T}_k(D_{train}), \quad k \in [1, 2, ..., m] \quad (5)$$

Propositions: In the following propositions, we derive the convergence time for learning robust and noisy features, and compare the same with the bounds derived by Shen *et al*. [57] in Section-6. The proofs of all propositions are presented in Section-1 of the Supplementary.

Notation: Let f_θ denote a neural network obtained by averaging the weights of m individual models f_θ^k, $k \in [1, m]$ which are represented as shown in Eq.1. n is the total number of data samples in the original dataset D_{train}. K is the number of orthonormal robust features in the dataset. The weights $w_1, w_2, \ldots, w_C$ of each model f_θ^k are initialized as $w_c \sim \mathcal{N}\left(0, \sigma_0^2 I_d\right)\ \forall\ c \in [1, C]$, where C is the number of channels in a single layer of the model. $\frac{\sigma}{\sqrt{d}}$ is the standard deviation of the noise in noisy patches, q is a hyperparameter used to define the activation (Details in Section-1 of the Supplementary), where $q \geq 3$ and d is the dimension of each feature patch and weight channel w_c.

Proposition 1. *The convergence time for learning any feature patch v_i $\forall i \in [1, K]$ in at least one channel $c \in C$ of the weight averaged model f_θ using the augmentations defined in Eq.5, is given by $O\left(\frac{K}{\sigma_0^{q-2}}\right)$, if $\frac{\sigma^q}{\sqrt{d}} \ll \frac{1}{K}$, $m = K$.*

Proposition 2. *If the noise patches learned by each f_θ^k are $i.i.d.$ Gaussian random variables $\sim \mathcal{N}(0, \frac{\sigma^2}{d} I_d)$ then with high probability, convergence time of learning a noisy patch $\epsilon^{(j)}$ in at least one channels $c \in [1, C]$ of the weight averaged model f_θ is given by $O\left(\frac{nm}{\sigma_0^{q-2}\sigma^q}\right)$, if $d \gg n^2$.*

Proposition 3. *If the noise learned by each f_θ^k are $i.i.d.$ Gaussian random variables $\sim \mathcal{N}\left(0, \frac{\sigma^2}{d} I_d\right)$, and model weight averaging is performed at epoch T, the convergence time of learning a noisy patch $\epsilon^{(j)}$ in at least one channels $c \in [1, C]$ of the weight averaged model f_θ is given by $T + O\left(\frac{nm^{(q-2)} d^{(q-2)/2}}{\sigma^{(2q-2)}}\right)$, if $d \gg n^2$.*

6. Analysis on the Theoretical Results

In this section, we present the implications of the theoretical results discussed above. While the setup in Section-5 discussed the existence of only two kinds of patches (feature and noisy), in practice, a combination of these two kinds of patches - termed as Spurious features - could also exist, whose convergence can be derived from the above results.

6.1. Learning Diverse Robust Features

We first show that *using sufficiently diverse data augmentations during training generates a uniform distribution of feature patches, encouraging the learning of diverse and robust features by the network.* We consider the use of m unique augmentations in Eq.3 which transform each feature patch into a different one using a unique mapping as shown in Eq.2. The mapping in Eq.2 can transform a skewed feature distribution to a more uniform distribution after performing augmentations. This results in K_{cut} being sufficiently large in Eq.1, which depends on the number of high frequency robust features, thereby encouraging the learning of a more balanced distribution of robust features. While Proposition-1 assumes that $m = K$, we show in Corollary 1.1 of the Supplementary that even when $m \neq K$, the learning of hard features is enhanced.

Shen *et al*. [57] show that the time for learning any feature patch v_k by at least one weight channel $c \in C$ is given by $O\left(\frac{1}{\sigma_0^{q-2}\rho_k}\right)$ if $\frac{\sigma^q}{\sqrt{d}} \ll \rho_k$, where ρ_k is the fraction of the frequency of occurrence of feature patch v_k divided by the total number of occurrences of all the feature patches in the dataset. The convergence time for learning feature patches is thus limited by the one that is least frequent in the input data. Therefore, by making the frequency of occurrence of all feature patches uniform, this convergence time reduces. In Proposition-1 we show that the same holds true even for the proposed method DART, where several branches are trained using diverse augmentations and their weights are finally averaged to obtain the final model. This justifies the improvements obtained in Mixed-Training (MT) (Eq.1) and in the proposed approach DART (Eq.4) as shown in Table-2.

6.2. Robustness to Noisy Features

Firstly, the use of diverse augmentations in both Mixed-Training (MT) and DART results in better robustness to noisy features since the value of K_{cut} in Eq.1 and Eq.4 would be higher, resulting in the learning of more feature patches and suppressing the learning of noisy patches. *The proposed method DART indeed suppresses the learning of noisy patches further, and also increases the convergence time for learning noisy features as shown in Proposition-2*. When the augmentations used in each of the m individual branches of DART are diverse, the noise learned by each of them can be assumed to be $i.i.d.$ Under this assumption, averaging model weights at the end of training results in a reduction of noise variance, as shown in Eq.4. More formally, we show in Proposition-2 that the *convergence time of noisy patches increases by a factor of m when compared to ERM training*. We note that this does not hold in the case of averaging model weights obtained during a single optimization trajectory as in SWA [31], EMA [50] or SWAD [4], since the noise learned by models that are close to each other in the optimization trajectory cannot be assumed to be $i.i.d.$

6.3. Impact of Intermediate Interpolations

We next analyse the impact of averaging the weights of the models at an intermediate epoch T in addition to the interpolation at the end of training. The individual models are further reinitialized using the weights of the interpolated model as discussed in Algorithm-1. As shown in Proposition-3, averaging the weights of all branches at the intermediate epoch T helps in increasing the convergence time of noisy patches by a factor $O\left(\frac{\sigma_0^{q-2} m^{q-3} d^{(q-2)/2}}{\sigma^{q-2}}\right)$ when compared to the case where models are interpolated only at the end of training as shown in Proposition-2. By assuming that $q > 3$ and $d \gg n^2$ similar to Shen *et al.* [57], the lower bound on this can be written as $O\left(\frac{\sigma_0 n}{\sigma}\right)$. We note that in a practical scenario this factor would be greater than 1, demonstrating the increase in convergence time for noisy patches when intermediate interpolation is done.

7. Experiments and Results

In this section, we empirically demonstrate the performance gains obtained using the proposed approach DART on In-Domain (ID) and Domain Generalization (DG) datasets. We further attempt to understand the various factors that contribute to the success of DART.

Dataset Details: To demonstrate In-Domain generalization, we present results on CIFAR-10 and CIFAR-100 [37], while for DG, we present results on the 5 real-world datasets on the DomainBed [23] benchmark - VLCS [17], PACS [39], OfficeHome [63], Terra Incognita [2] and DomainNet [47], which represent several types of domain shifts with different levels of dataset and task complexities.

Table 2. **In-Domain Generalization:** Performance (%) of DART when compared to baselines on WideResNet-28-10 model. Standard deviation for DART and MT is reported across 5 reruns.

Method	CIFAR-10	CIFAR-100
ERM+EMA (Pad+Crop+HFlip)	96.41	81.67
ERM+EMA (AutoAugment)	97.50	84.20
ERM+EMA (Cutout)	97.43	82.33
ERM+EMA (Cutmix)	97.11	84.05
Learning Subspaces [70]	97.46	83.91
ERM+EMA (Mixed Training-MT)	97.69 $\pm$ 0.19	85.57 $\pm$ 0.13
DART (Ours)	**97.96** $\pm$ 0.06	**86.46** $\pm$ 0.12

Table 3. **DART on ImageNet-1K and finegrained datasets:** Performance (%) of DART when compared to ERM+EMA Mixed Training baseline on ResNet-50. In the first row, a *Single Augmentation (SA)* is used in all branches (RandAugment [9] for ImageNet-1K, and Pad-Crop for finegrained datasets). In the second row, *Mixed Augmentations (MA)* - Pad-Crop, RandAugment [9] and Cutout [10] are used in different branches. AutoAugment [8] is used instead of RandAugment for finegrained datasets in the latter case of Mixed Augmentations (MA).

	Stanford-CARS		CUB-200		Imagenet-1K	
	ERM + EMA	DART	ERM + EMA	DART	ERM + EMA	DART
SA	88.11	**90.42**	78.55	**79.75**	78.55	**78.96**
MA	90.88	**91.95**	81.72	**82.83**	79.06	**79.20**

Training Details (ID): The training epochs are set to 600 for the In-Domain experiments on CIFAR-10 and CIFAR-100. To enable a fair comparison, the best performing configuration amongst 200, 400 and 600 total training epochs is used for the ERM baselines and Mixed-Training, since they may be prone to overfitting. We use SGD optimizer with momentum of 0.9, weight decay of 5e-4 and a cosine learning rate schedule with a maximum learning rate of 0.1. Interpolation frequency (λ) is set to 50 epochs for CIFAR-100 and 40 epochs for CIFAR-10. As shown in Fig-3(b), accuracy is stable when $\lambda \in [10, 80]$. We present results on ResNet-18 and WideResNet-28-10 architectures.

Training Details (DG): Following the setting in DomainBed [23], we use Adam [35] optimizer with a fixed learning rate of 5e-5. The number of training iterations are set to 15k for DomainNet (due to its higher complexity) and 10k for all other datasets with the interpolation frequency being set to 1k iterations. ResNet-50 [24] was used as the backbone, initialized with Imagenet [54] pre-trained weights. Best-model selection across training checkpoints was done based on validation results from the train domains itself, and no subset of the test domain was used. We use fixed values of hyperparameters for all datasets in the DG setting. As shown in Fig.6 (a) of the Supplementary, ID and OOD accuracies are correlated, showing that hyperparameter tuning based on ID validation accuracy as suggested by Gulrajani *et al.* [23] can indeed improve our results further. We present further details in Section-4 of Supplementary.

Table 4. **Domain Generalization:** OOD accuracy(%) of DART when compared to the respective baselines on DomainBed datasets with ResNet-50 model. Standard dev. across 3 reruns is reported.

Algorithm	VLCS	PACS	OfficeHome	TerraInc	DomainNet	Avg
ERM [62]	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	63.3
+ DART (Ours)	78.5 ± 0.7	87.3 ± 0.5	70.1 ± 0.2	48.7 ± 0.8	45.8 ± 0.0	66.1
SWAD [4]	79.1 ± 0.1	88.1 ± 0.1	70.6 ± 0.2	50.0 ± 0.3	46.5 ± 0.1	66.9
+ DART (Ours)	**80.3** ± 0.2	**88.9** ± 0.1	**71.9** ± 0.1	**51.3** ± 0.2	**47.1** ± 0.0	**67.9**

Table 5. **Combining DART with other DG methods (OfficeHome):** OOD performance (%) of the proposed method DART coupled with different algorithms against their vanilla and SWAD counterparts. Numbers represented with † were reproduced while others are from Domainbed [23]. All models except the last row are trained on a ResNet-50 Imagenet pretrained model. The last row shows results on a CLIP initialized ViT-B/16 model.

Algorithm	Vanilla	DART (w/o SWAD)	SWAD	DART (+ SWAD)
ERM [62]	66.5	70.31	70.60	**72.28**
ARM [78]	64.8	69.24	69.75	**71.31**
SAM† [18]	67.4	70.39	70.26	**71.55**
Cutmix† [75]	67.3	70.07	71.08	**71.49**
Mixup [68]	68.1	71.14	71.15	**72.38**
DANN [20]	65.9	70.32	69.46	**70.85**
CDANN [40]	65.8	70.75	69.70	**71.69**
SagNet [44]	68.1	70.19	70.84	**71.96**
MIRO [5]	70.5	72.54	72.40	**72.71**
MIRO (CLIP)†	83.3	86.14	84.80	**87.37**

In Domain (ID) Generalization: In Table-2, we compare our method against ERM training with several augmentations, and also the strong Mixed-Training benchmark (MT) obtained by using either AutoAugment [8], Cutout [10] or Cutmix [75] for every image in the training minibatch uniformly at random. We use the same augmentations in DART as well, with each of the 3 branches being trained on one of the augmentations. As discussed in Section-3, the method proposed by Wortsman *et al.* [70] is closest to our approach, and hence we compare with it as well. We utilize Exponential Moving Averaging (EMA) [50] of weights for the ERM baselines and the proposed approach for a fair comparison. On CIFAR-10, we observe gains of 0.19% on using ERM-EMA (Mixed) and an additional 0.27% on using DART. On CIFAR-100, 1.37% improvement is observed with ERM-EMA (Mixed) and an additional 0.89% with the proposed method DART. We also incorporate DART with SAM [18] and obtain $\sim 0.2\%$ gains over ERM + SAM with Mixed Augmentations as shown in Table-2 of the Supplementary. The comparison of DART with the Mixed Training benchmark (ERM+EMA on mixed augmentations) on ImageNet-1K and fine-grained datasets, Stanford-Cars [36] and CUB-200 [69] on an ImageNet pretrained model is shown in Table-3. On ImageNet-1K, we obtain 0.41% gains on using RandAugment [9] across all the branches, and 0.14% gains on using Pad-Crop, RandAugment and Cutout for different branches. We obtain gains of upto 1.5% on fine-grained datasets.

Table 6. **DART using same augmentation across all branches:** Performance (%) of DART when compared to baselines across different augmentations on CIFAR-100 using WideResNet-28-10 architecture. DART is better than baselines in all cases.

Method	Pad+Crop+HFlip	AutoAug.	Cutout	Cutmix	Mixed-Train.
ERM	81.48	83.93	82.01	83.02	85.54
ERM + EMA	81.67	84.20	82.33	84.05	85.57
DART (Ours)	**82.31**	**85.02**	**84.15**	**84.72**	**86.13**

SOTA comparison - Domain Generalization: We present results on the DomainBed [23] datasets in Table-4. We compare only with ERM training (performed on data from a mix of all domains) and SWAD [4] in the main paper due to lack of space, and present a thorough comparison across all other baselines in Section-4.3 of the Supplementary. For the DG experiments, we consider 4 branches ($M = 4$), with 3 branches being specialists on a given domain and the fourth being trained on a combination of all domains in equal proportion. For the DomainNet dataset, we consider 6 branches due to the presence of more domains. On average, we obtain 2.8% improvements over the ERM baseline without integrating with SWAD, and 1% higher accuracy when compared to SWAD by integrating our approach with it. We further note from Table-5 that the DART can be integrated with several base approaches - with and without SWAD, while obtaining substantial gains across the respective baselines. The proposed approach therefore is generic, and can be integrated effectively with several algorithms. As shown in the last row, we obtain substantial gains of 2.6% on integrating DART with SWAD and a recent work MIRO [5] using CLIP initialization [52] on a ViT-B/16 model [13].

Evaluation without imposing diversity across branches: While the proposed approach imposes diversity across branches by using different augmentations, we show in Table-6 that it works even without explicitly introducing diversity, by virtue of the randomness introduced by SGD and different ordering of input samples across models. We obtain an average improvement of 0.9% over the respective baselines, and maximum improvement of 1.82% using Cutout. This shows that the performance of DART is not dependent on data augmentations, although it achieves further improvements on using them.

Accuracy across training epochs: We show the accuracy across training epochs for the individual branches and the combined model in Fig.4 for two cases - (a) performing interpolations from the beginning, and (b) performing interpolations after half the training epochs, as done in DART. It can be noted from (a) that the interpolations in the initial few epochs have poor accuracy since the models are not in a common basin. Further, as seen in initial epochs of (a), when the learning rate is high, SGD training on an interpolated model cannot retain the flat solution due to its implicit

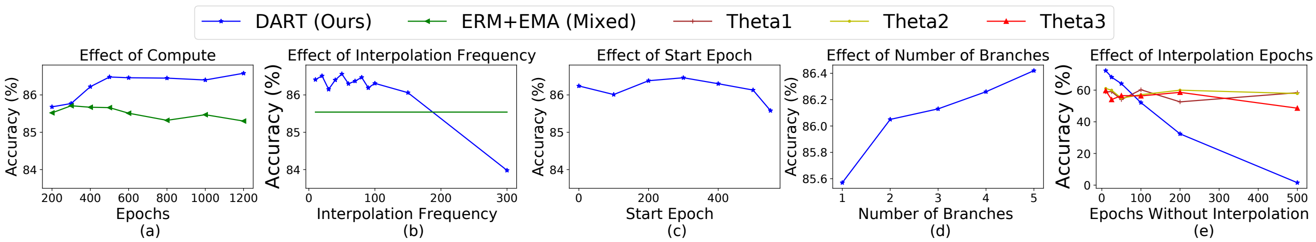

Figure 3. **Ablations on CIFAR-100, WideResNet-28-10**: (a-d) Experiments comparing DART with the Mixed-Training baseline using the standard training settings. (e) Varying the interpolation epoch after 50 epochs of common training using a fixed learning rate of 0.1.

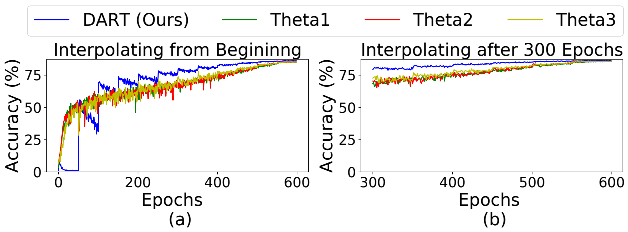

Figure 4. **Accuracy of DART across training epochs** for CIFAR-100 on WideResNet-28-10 model: Each branch is trained on different augmentations, whose accuracy is also plotted. Model Interpolation is done (a) from the beginning, (b) after 300 epochs. Although model interpolation and reinitialization happens every 50 epochs, interpolated model accuracy is plotted every epoch.

bias of moving towards solutions that minimize train loss alone. Whereas, in the later epochs as seen in (b), the improvement obtained after every interpolation is retained. We therefore propose a common training strategy for the initial half of epochs, and split training after that.

Ablation experiments: We note the following observations from the plots in Fig.3 (a-e):

(a) **Effect of Compute:** Using DART, we obtain higher (or similar) performance gains as the number of training epochs increases, whereas the accuracy of ERM+EMA (Mixed) benchmark starts reducing after 300 epochs of training. This can be attributed to the increase in convergence time for learning noisy (or spurious) features due to the intermediate aggregations as shown in Proposition-3, which prevents overfitting.

(b) **Effect of Interpolation Frequency:** We note that an optimal range of λ or the number of epochs between interpolations is 10 - 80, and we set this value to 50. If there is no interpolation for longer epochs, the models drift apart too much, causing a drop in accuracy.

(c) **Effect of Start Epoch:** We note that although the proposed approach works well even if interpolations are done from the beginning, by performing ERM training on mixed augmentations for 300 epochs, we obtain 0.22% improvement. Moreover, since interpolations do not help in the initial part of training as seen in Fig.4 (a), we propose to start this only in the second half.

(d) **Effect of Number of branches:** As the number of branches increases, we note an improvement in performance due to higher diversity across branches, leading to more robustness to spurious features and better generalization as shown in Proposition-2.

(e) **Effect of Interpolation epochs:** We perform an experiment with 50 epochs of common training followed by a single interpolation. We use a fixed learning rate and plot the accuracy by varying the interpolation epoch. As this value increases, models drift far apart, reducing the accuracy after interpolation. At epoch-500, the accuracy even reaches 0, highlighting the importance of having a low loss barrier between models.

8. Conclusion

In this work, we first show that ERM training using a combination of *diverse* augmentations within a training minibatch can be a strong benchmark for ID generalization, which is outperformed only by ensembling the outputs of individual experts. Motivated by this observation, we present DART - Diversify-Aggregate-Repeat Training, to achieve the benefits of training diverse experts and combining their expertise throughout training. The proposed algorithm first trains several models on different augmentations (or domains) to learn a *diverse* set of features, and further *aggregates* their weights to obtain better generalization. We repeat the steps Diversify-Aggregate several times over training, and show that this makes the optimization trajectory more robust by suppressing the learning of noisy features, while also ensuring a low loss barrier between the individual models to enable their effective aggregation. We justify our approach both theoretically and empirically on several benchmark In-Domain and Domain Generalization datasets, and show that it integrates effectively with several base algorithms as well. We hope our work motivates further research on leveraging the linear mode connectivity of models for better generalization.

9. Acknowledgments

This work was supported by the research grant CRG/2021/005925 from SERB, DST, Govt. of India. Sravanti Addepalli is supported by Google PhD Fellowship.

References

[1] Sravanti Addepalli, Samyak Jain, et al. Efficient and effective augmentation strategy for adversarial training. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:1488–1501, 2022. 1

[2] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018. 6

[3] Manh-Ha Bui, Toan Tran, Anh Tran, and Dinh Phung. Exploiting domain-specific features to enhance domain generalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:21189–21201, 2021. 2

[4] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:22405–22418, 2021. 1, 3, 6, 7

[5] Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain generalization by mutual-information regularization with pre-trained models. *European Conference on Computer Vision (ECCV)*, 2022. 3, 7

[6] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *European Conference on Computer Vision (ECCV)*, pages 301–318. Springer, 2020. 2

[7] Ching-Yao Chuang, Antonio Torralba, and Stefanie Jegelka. Estimating generalization under distribution shifts via domain-invariant representations. *arXiv preprint arXiv:2007.03511*, 2020. 2

[8] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 6, 7

[9] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 18613–18624, 2020. 1, 2, 6, 7

[10] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1, 2, 6, 7

[11] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000. 2

[12] Zhengming Ding and Yun Fu. Deep domain generalization with structured low-rank constraint. *IEEE Transactions on Image Processing (TIP)*, 27(1):304–313, 2018. 2

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 7

[14] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning (ICML)*, pages 1309–1318. PMLR, 2018. 2, 3

[15] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Uncertainty in Artificial Intelligence (UAI)*. PMLR, 2016. 1, 3

[16] Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[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 *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 1657–1664, 2013. 6

[18] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 3, 7

[19] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning (ICML)*, pages 3259–3269. PMLR, 2020. 2, 3

[20] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research (JMLR)*, 17(59):1–35, 2016. 7

[21] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018. 2, 3

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

[23] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 3, 6, 7

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[25] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019. 1

[26] Dan Hendrycks*, Norman Mu*, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Aug-

mix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations (ICLR)*, 2020. 2

[27] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8126–8135, 2020. 1

[28] Shoubo Hu, Kun Zhang, Zhitang Chen, and Laiwan Chan. Domain generalization via multidomain discriminant analysis. In *Uncertainty in Artificial Intelligence (UAI)*, pages 292–302. PMLR, 2020. 2

[29] W Ronny Huang, Zeyad Ali Sami Emam, Micah Goldblum, Liam H Fowl, Justin K Terry, Furong Huang, and Tom Goldstein. Understanding generalization through visualizations. In *"I Can't Believe It's Not Better!" NeurIPS 2020 workshop*, 2020. 1, 3

[30] Maximilian Ilse, Jakub M. Tomczak, Christos Louizos, and Max Welling. Diva: Domain invariant variational autoencoders. In *Proceedings of the Third Conference on Medical Imaging with Deep Learning (MIDL)*, pages 322–348, 2020. 2

[31] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 876–885, 2018. 2, 3, 6

[32] Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations (ICLR)*, 2020. 1, 3

[33] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[34] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, pages 158–171, 2012. 2

[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015. 6

[36] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 7

[37] Alex Krizhevsky. Learning multiple layers of features from tiny images. *https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf*, 2009. 6

[38] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems (NeurIPS)*, 30, 2017. 2

[39] 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 (ICCV)*, Oct 2017. 2, 6

[40] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 2018. 7

[41] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 2

[42] Soon Hoe Lim, N. Benjamin Erichson, Francisco Utrera, Winnie Xu, and Michael W. Mahoney. Noisy feature mixup. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2

[43] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *European Conference on Computer Vision (ECCV)*, pages 466–483. Springer, 2020. 3

[44] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8690–8699, 2021. 7

[45] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems (NeurIPS)*, 30, 2017. 2

[46] Quynh Nguyen. On connected sublevel sets in deep learning. In *International conference on machine learning (ICML)*, pages 4790–4799. PMLR, 2019. 2, 3

[47] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1406–1415, 2019. 6

[48] Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative flatness and generalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:18420–18432, 2021. 1, 3

[49] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *International Conference on Machine Learning (ICML)*, pages 7728–7738. PMLR, 2020. 2

[50] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 3, 6, 7

[51] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12556–12565, 2020. 3

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 7

[53] Alexandre Ramé, Rémy Sun, and Matthieu Cord. Mixmo: Mixing multiple inputs for multiple outputs via deep subnet-

works. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 823–833, 2021. 1

[54] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision (IJCV)*, 115(3):211–252, 2015. 6

[55] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:9573–9585, 2020. 1

[56] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018. 3

[57] Ruoqi Shen, Sébastien Bubeck, and Suriya Gunasekar. Data augmentation as feature manipulation: a story of desert cows and grass cows. In *International Conference on Machine Learning (ICML)*, 2022. 1, 4, 5, 6

[58] Yichun Shi, Xiang Yu, Kihyuk Sohn, Manmohan Chandraker, and Anil K Jain. Towards universal representation learning for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6817–6826, 2020. 3

[59] Saurabh Singh, Derek Hoiem, and David Forsyth. Swapout: Learning an ensemble of deep architectures. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016. 2

[60] David Stutz, Matthias Hein, and Bernt Schiele. Relating adversarially robust generalization to flat minima. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7807–7817, 2021. 3

[61] Norman Tatro, Pin-Yu Chen, Payel Das, Igor Melnyk, Prasanna Sattigeri, and Rongjie Lai. Optimizing mode connectivity via neuron alignment. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:15300–15311, 2020. 3

[62] Vladimir Vapnik. Statistical learning theory wiley. *New York*, 1998. 7

[63] 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 (CVPR)*, pages 5018–5027, 2017. 6

[64] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning (ICML)*, pages 6438–6447. PMLR, 2019. 2

[65] Riccardo Volpi and Vittorio Murino. Addressing model vulnerability to distributional shifts over image transformation sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3

[66] 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 (NeurIPS)*, 31, 2018. 3

[67] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 04 2020. 2

[68] Yufei Wang, Haoliang Li, and Alex C Kot. Heterogeneous domain generalization via domain mixup. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3622–3626. IEEE, 2020. 3, 7

[69] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-201, Caltech, 2010. 7

[70] Mitchell Wortsman, Maxwell C Horton, Carlos Guestrin, Ali Farhadi, and Mohammad Rastegari. Learning neural network subspaces. In *International Conference on Machine Learning (ICML)*, pages 11217–11227. PMLR, 2021. 2, 3, 6, 7

[71] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, pages 23965–23998. PMLR, 2022. 2, 3

[72] Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[73] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *International Conference on Learning Representations (ICLR)*, 2021. 3

[74] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3

[75] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 6023–6032, 2019. 1, 2, 7

[76] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, 2021. 2

[77] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 2

[78] Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk

minimization: Learning to adapt to domain shift. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:23664–23678, 2021. 7

[79] Shaofeng Zhang, Meng Liu, and Junchi Yan. The diversified ensemble neural network. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:16001–16011, 2020. 2

[80] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In *International Conference on Learning Representations (ICLR)*, 2020. 3

[81] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European conference on computer vision (ECCV)*, pages 561–578. Springer, 2020. 3

[82] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations (ICLR)*, 2021. 3