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
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 Diversify-Aggregate-Repeat training 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.

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

<table>
<thead>
<tr>
<th>Train Augmentation</th>
<th>No Aug.</th>
<th>Cutout</th>
<th>Cutmix</th>
<th>AutoAugment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pad+Crop+HFlip (PC)</td>
<td>78.51</td>
<td>67.04</td>
<td>56.52</td>
<td>58.33</td>
</tr>
<tr>
<td>Cutout (CO)</td>
<td>77.99</td>
<td>74.58</td>
<td>56.12</td>
<td>58.47</td>
</tr>
<tr>
<td>Cutmix (CM)</td>
<td>80.54</td>
<td>74.05</td>
<td>77.35</td>
<td>61.23</td>
</tr>
<tr>
<td>AutoAugment (AA)</td>
<td>79.18</td>
<td>71.26</td>
<td>60.97</td>
<td>73.91</td>
</tr>
<tr>
<td>Mixed-Training (MT)</td>
<td>81.43</td>
<td>77.31</td>
<td>73.20</td>
<td><strong>74.73</strong></td>
</tr>
<tr>
<td>Ensemble (CM+CO+AA)</td>
<td><strong>83.61</strong></td>
<td><strong>79.19</strong></td>
<td>73.19</td>
<td>73.90</td>
</tr>
</tbody>
</table>

![Figure 1. Schematic Diagram of the proposed method DART](image)

### 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 $\theta_1$ and $\theta_2$ is defined as the maximum loss attained by the models, $\theta = \alpha \cdot \theta_1 + (1 - \alpha) \cdot \theta_2 \forall \alpha \in [0,1]$.

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 $\ell_2$ norm ball of weights can result in a flatter minima thereby improving generalization. Gurlajani 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 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.
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 $\lambda/M$ 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. We assume that 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.

### Algorithm 1 Diversify-Aggregate-Repeat Training, DART

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

### 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 $(x_1, x_2) \in R^{d_1 \times 2}$, 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 $\epsilon \sim N(0, \sigma^2 I_d)$.

We consider a single layer convolutional neural network consisting of $C$ channels, with $w = (w_1, w_2, w_3, \ldots, w_C) \in R^{d_2 \times C}$. 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 $\phi$ is the activation function as defined by Shen et al. [57].
Weights learned by an ERM trained model: Let \( K_{\text{cut}} \) 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:

\[
w = \sum_{k=1}^{K_{\text{cut}}} v_k + \sum_{k > K_{\text{cut}}} y^{(k)} \epsilon^{(k)}
\]

(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):

\[
\forall k' \in [1, K], \quad T_k(v_{k'}) = v_{((k'+k-1) \mod K)+1}
\]

(2)

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

\[
D^{(\text{aug})}_{\text{train}} = D_{\text{train}} \cup T_1(D_{\text{train}}) \cup \ldots \cup T_{m-1}(D_{\text{train}})
\]

(3)

where \( D_{\text{train}} \) 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:

\[
w = \frac{1}{m} \sum_{j=1}^{m} \sum_{k=1}^{K_{\text{cut}}} v_{k_j} + \frac{1}{m} \sum_{j=1}^{m} \sum_{k > K_{\text{cut}}} y^{(k)} j^{(k)}
\]

(4)

Each branch is trained on the dataset \( D^{(\text{aug})}_{\text{train}} \) defined as:

\[
D^{(k)}_{\text{train}} = T_k(D_{\text{train}}), \quad k \in [1, 2, ..., m]
\]

(5)

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 \( f_\theta \) using the augmentations defined in Eq.5, is given by \( O \left( \frac{K}{\sigma^0_\epsilon} \right) \), if \( \frac{\sigma^0_\epsilon}{\epsilon} \ll \frac{1}{K}, \quad m = K \).

Proposition 2. If the noise patches learned by each \( f^c_\theta \) are i.i.d. Gaussian random variables \( \sim \mathcal{N}(0, \frac{\sigma^2}{\epsilon} I_d) \) 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 \( f_\theta \) is given by \( O \left( \frac{nm}{\sigma^2 + \sigma^0_\epsilon} \right) \), if \( d \gg n^2 \).

Proposition 3. If the noise learned by each \( f^c_\theta \) are i.i.d. Gaussian random variables \( \sim \mathcal{N}(0, \frac{\sigma^2}{\epsilon} I_d) \), 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 \( f_\theta \) is given by \( T + O \left( \frac{2 m (n-2)^d (\epsilon^{(j)} - 2)^2}{\sigma^2} \right) \), 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 \( K_{\text{cut}} \) 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 \( v_i \) by at least one weight channel \( c \in C \) is given by \( O \left( \frac{1}{\sigma^0_\epsilon} \right) \) if \( \frac{\sigma^0_\epsilon}{\epsilon} \ll \rho_k \), where \( \rho_k \) is the fraction of the frequency of occurrence of feature patch \( v_i \) 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 $K_{ctu}$ 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 $O \left( \frac{\sigma_q^2 \sigma_d^2}{\sigma_0^2} \right)$ 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 $O \left( \frac{\sigma_0^2}{\sigma_0^2} \right)$. 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM+EMA (Pad+Crop+HFlip)</td>
<td>96.41</td>
<td>81.67</td>
</tr>
<tr>
<td>ERM+EMA (AutoAugment)</td>
<td>97.50</td>
<td>84.20</td>
</tr>
<tr>
<td>ERM+EMA (Cutout)</td>
<td>97.43</td>
<td>82.33</td>
</tr>
<tr>
<td>ERM+EMA (Cutmix)</td>
<td>97.11</td>
<td>84.05</td>
</tr>
<tr>
<td>Learning Subspaces [70]</td>
<td>97.46</td>
<td>83.91</td>
</tr>
<tr>
<td>ERM+EMA (Mixed Training-MT)</td>
<td>97.69 ± 0.13</td>
<td>85.57 ± 0.13</td>
</tr>
<tr>
<td>DART (Ours)</td>
<td>97.96 ± 0.06</td>
<td>86.46 ± 0.12</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Method</th>
<th>Stanford-CARS</th>
<th>CUB-200</th>
<th>Imagenet-1K</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM + EMA</td>
<td>ME</td>
<td>ME</td>
<td>ME</td>
</tr>
<tr>
<td>DART</td>
<td>90.88</td>
<td>91.95</td>
<td>79.06</td>
</tr>
<tr>
<td>ERM + EMA</td>
<td>91.38</td>
<td>82.83</td>
<td>79.06</td>
</tr>
<tr>
<td>Mixed Aug.</td>
<td>89.08</td>
<td>91.95</td>
<td>82.83</td>
</tr>
<tr>
<td>DART</td>
<td>97.96 ± 0.06</td>
<td>86.46 ± 0.12</td>
<td></td>
</tr>
</tbody>
</table>

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 ($\lambda$) 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.
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 mini-batch 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 ~ 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 pre-trained 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 4. Domain Generalization:** OOD accuracy(%) of DART when compared to the respective baselines on DomainBed datasets with ResNet-50 model. Standard dev. across 3 runs is reported.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Vanilla</th>
<th>DART (w/o SW AD)</th>
<th>SWAD</th>
<th>DART (+ SWAD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM [62]</td>
<td>66.5</td>
<td>70.31</td>
<td>70.60</td>
<td>72.28</td>
</tr>
<tr>
<td>ARM [78]</td>
<td>64.8</td>
<td>69.24</td>
<td>69.75</td>
<td>71.31</td>
</tr>
<tr>
<td>SAM† [18]</td>
<td>67.4</td>
<td>70.39</td>
<td>70.26</td>
<td>71.55</td>
</tr>
<tr>
<td>Cutmix† [75]</td>
<td>67.3</td>
<td>70.07</td>
<td>71.08</td>
<td>71.49</td>
</tr>
<tr>
<td>Mixup [68]</td>
<td>68.1</td>
<td>71.14</td>
<td>71.15</td>
<td>72.38</td>
</tr>
<tr>
<td>DANN [20]</td>
<td>65.9</td>
<td>70.32</td>
<td>69.46</td>
<td>70.85</td>
</tr>
<tr>
<td>CDANN [49]</td>
<td>65.8</td>
<td>70.75</td>
<td>69.70</td>
<td>71.69</td>
</tr>
<tr>
<td>SagNet [44]</td>
<td>68.1</td>
<td>70.19</td>
<td>70.84</td>
<td>71.96</td>
</tr>
<tr>
<td>MIRO [5]</td>
<td>70.5</td>
<td>72.54</td>
<td>72.40</td>
<td>72.71</td>
</tr>
<tr>
<td>MIRO (CLIP)†</td>
<td>83.3</td>
<td>86.14</td>
<td>84.80</td>
<td>87.37</td>
</tr>
</tbody>
</table>

**Table 5. Combining DART with other DG methods (Office-Home):** 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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Vanilla</th>
<th>DART (w/o SW AD)</th>
<th>SWAD</th>
<th>DART (+ SWAD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM [62]</td>
<td>71.96</td>
<td>72.28</td>
<td>71.55</td>
<td>71.49</td>
</tr>
<tr>
<td>ARM [78]</td>
<td>67.4</td>
<td>70.31</td>
<td>70.26</td>
<td>71.55</td>
</tr>
<tr>
<td>SAM† [18]</td>
<td>71.31</td>
<td>71.14</td>
<td>71.15</td>
<td>72.38</td>
</tr>
<tr>
<td>Cutmix† [75]</td>
<td>71.3</td>
<td>70.07</td>
<td>71.08</td>
<td>71.49</td>
</tr>
<tr>
<td>Mixup [68]</td>
<td>51.3</td>
<td>53.04</td>
<td>53.03</td>
<td>53.02</td>
</tr>
<tr>
<td>DANN [20]</td>
<td>65.9</td>
<td>70.32</td>
<td>69.46</td>
<td>70.85</td>
</tr>
<tr>
<td>CDANN [49]</td>
<td>65.8</td>
<td>70.75</td>
<td>69.70</td>
<td>71.69</td>
</tr>
<tr>
<td>SagNet [44]</td>
<td>68.1</td>
<td>70.19</td>
<td>70.84</td>
<td>71.96</td>
</tr>
<tr>
<td>MIRO [5]</td>
<td>70.5</td>
<td>72.54</td>
<td>72.40</td>
<td>72.71</td>
</tr>
<tr>
<td>MIRO (CLIP)†</td>
<td>83.3</td>
<td>86.14</td>
<td>84.80</td>
<td>87.37</td>
</tr>
</tbody>
</table>

**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
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 $\lambda$ 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


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


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


