Grounded Imagination Leads to Adaptation

Judah Goldfeder[1], Shreyes Kaliyur[1], Vaibhav Sourirajan[1], Patrick Minwan Puma[2], Philippe Martin Wyder[3], Yuhang Hu[1], Jiong Lin[1], Hod Lipson[1]

[1]Columbia University, New York, NY
[2]Harvard University, Cambridge, MA
[3]University of Washington, Seattle, WA
jag2396@columbia.edu

Abstract

Imagination — the capacity to simulate plausible unseen variations of the world — is a fundamental mechanism of biological adaptation. In data-scarce environments, organisms that can mentally rehearse novel scenarios gain decisive advantages over those limited to direct experience. We argue that adaptive imagination requires two components: a generative world model capable of generating plausible novel scenarios, and a selection process that keeps those scenarios grounded and relevant. Neither alone is sufficient. The most straightforward form of imagination is visual: picturing objects and scenes in ways we have not previously seen them. Thus, to test this hypothesis, we consider a few-shot computer vision setting. We present EvoAug, a computational model of this principle: an automated pipeline that uses generative models to synthesize task-relevant visual variations, and evolutionary search to ensure those simulations remain grounded and adaptive. Without evolutionary selection, unconstrained generative augmentation degrades performance — the computational analog of maladaptive fantasy. With it, the system discovers augmentation strategies that recover domain structure without explicit supervision, even from a single example per class. Results across six fine-grained classification datasets demonstrate that combining a generative world model with evolutionary grounding outperforms classical augmentation strategies, with gains largest precisely where data is scarcest — the conditions under which imagination matters most. We discuss implications for understanding imagination as a key adaptive mechanism in biological and artificial systems.

Data/Code available at: `https://github.com/Jgoldfeder/artificial_imagination`

Introduction

Imagination — the capacity to simulate plausible unseen variations of the world — is a fundamental mechanism of biological adaptation and planningHassabis et al. (2017). Ha and Schmidhuber termed the generative model that governs the realm of imagination a "world model"Ha and Schmidhuber (2018), a term that has since gained major traction in the AI community under various interpretationsHuang et al. (2026); LeCun et al. (2022). Yet the discussion has drifted away from the initial connection with imagination, and the role of mental world models and imagination from the perspective of artificial life remains largely unexplored, especially as it relates to adaptation, a key hallmark of intelligence Goldfeder et al. (2026).

We examine the role of imagination in adaptation through the lens of few-shot visual classification, which serves as an ideal testbed for several reasons. First, visual simulation is one of the most natural and direct forms of imaginationPearson (2019). Second, the low-data regime is precisely where imagination matters most: when direct experience is scarce, the ability to mentally rehearse unseen variations becomes a critical adaptive advantage. Third, the setting is sufficiently challenging that unconstrained imagination is actively harmful, as a world model that drifts too far from reality reinforces errors rather than correcting them. Finally, generative AI has now advanced to the point where visually convincing imaginative scenarios can be synthesized from minimal input.

A key challenge in leveraging synthetic data is the syn-to-real gap—the discrepancy between generated and real-world data. Poorly matched synthetic augmentations degrade performance rather than enhance it. For example, diffusion models still struggle with fine details such as realistic fingers (Narasimhaswamy et al., 2024). Thus, a model trained on data augmented by flawed synthetic images may reinforce errors. Similarly, a language model could amplify its own biases by training on text that it generated itself. This issue is particularly critical in tasks requiring fine-grained distinctions, such as image classification, or in low-data settings like few-shot learning. Addressing this gap is essential for generative augmentations to contribute meaningfully to AI training.

Thus, we argue that imagination never exists without constraint and grounding. We do not simply imagine the future; we also take note of how likely or realistic we deem a scenario, and update these priors based on our experience. This is why imagination is so useful for adaptation: it allows to

© 2026 [Judah Goldfeder, Shreyes Kaliyur, Vaibhav Sourirajan, Patrick Minwan Puma, Philippe Martin Wyder, Yuhang Hu, Jiong Lin, and Hod Lipson]. Published under a Creative Commons Attribution 4.0 International (CC BY 4.0) license.

Downloaded from direct.mit.edu/isal/proceedings-pdf/isal2026/38/124/2620452/isal.a.1050.pdf by guest on 31 October 2026

separate, as Aristotle would put it, essential from accidental propertiesRobertson Ishii and Atkins (2008). To illustrate this with an example: we can imagine a chair as purple or blue or brown or black. However, we struggle to imagine a chair without legs, or without a seat. This is because "color" is an accidental property of chairs, whereas "possessing a seat" is essential. Our world model is built from experience: we have encountered many chairs of various colors and shapes, but all of them have had legs and seats in some form. This allows for adaptation: we can now easily recognize new chairs, even if they are not in a color or shape we have seen before.

Motivated by this, we designed EvoAug, a computer vision system inspired by the role of imagination in biological cognition. The imagination "world model" consists of several powerful generative diffusion and NeRF models, that can be composed and constrained in a variety of ways. For each dataset, we learn an augmentation tree—a binary tree that applies a series of augmentation operators in accordance with learned branching probabilities. This is the grounding process described above: we discover via trial and error which types of synthetic data are useful for each specific task. Our approach provides promising results on fine-grained and few-shot classification tasks across multiple datasets.

Related Work

Data augmentation reduces model overfitting by applying image transformations that preserve the original semantics while introducing controlled diversity into the training set. Traditional augmentations include rotations, random cropping, mirroring, scaling, and other basic transformations. These straightforward techniques remain fundamental in state-of-the-art image augmentation pipelines. More advanced methods—such as erasing (Zhong et al., 2020; Chen et al., 2020; Li et al., 2020; DeVries and Taylor, 2017), copy-pasting (Ghiasi et al., 2021), image mixing (Zhang et al., 2017; Yun et al., 2019), and data-driven augmentations like AutoAugment (Cubuk et al., 2018) and its simplified variant RandAugment (Cubuk et al., 2020)—have expanded the augmentation toolbox.

Another approach involves generating synthetic data using generative models (Figueira and Vaz, 2022). Early work explored GANs (Besnier et al., 2020; Jahanian et al., 2021; Brock et al., 2018), VAEs (Razavi et al., 2019), and CLIP (Ramesh et al., 2022), achieving strong results (Engelsma et al., 2022; Skandarani et al., 2023). Recently, diffusion models, particularly for text-to-image synthesis, have surpassed GANs in producing photorealistic images (Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022b; Yang et al., 2025). Trained on large-scale internet data (Schuhmann et al., 2022), diffusion models have been used for augmentation (Azizi et al., 2023; Sarıyıldız et al., 2023; He et al., 2022; Shipard et al., 2023; Rombach et al., 2022; Islam et al., 2025, 2024), often relying on class names or simple class agnostics prompts to guide generation. Despite promising initial results, synthetic data remains inferior to real data, highlighting the persistent domain gap between the two (Yamaguchi and Fukuda, 2023).

To address this gap, recent approaches have incorporated conditioning the generative process on real data. Some popular methods involve projecting the original images to the diffusion latent space (Zhou et al., 2023), fine-tuning diffusion models on real data (Azizi et al., 2023), leveraging multi-modal LLMs to obtain detailed, custom image captions for high-quality text prompting(Yu et al., 2023), and employing image-to-image diffusion models that enable direct conditioning on a specific image (Saharia et al., 2022a; Meng et al., 2021; Zhang et al., 2023; He et al., 2022; Trabucco et al., 2025). Controlled diffusion, a subset of these methods, introduces a more powerful paradigm, furthering the efficient use of both text and image priors (Fang et al., 2024; Islam and Akhtar, 2025) with applications in segmentation (Trabucco et al., 2023) and classification (Goldfeder et al., 2024b) problems.

Given such a wide range of augmentation operators, an important problem is knowing which augmentations to use for a specific task, without the use of domain knowledge. This task, of automatically learning augmentation policies, falls under the class of meta learning and bi-level optimization problems, where we seek to learn a component of the learning algorithm itself (Hospedales et al., 2021). These algorithms generally fall under one of the following categories: gradient-based optimization, RL-based optimization, Bayesian optimization, and evolution-based optimization.

In the context of learning augmentation policies, all these methods have seen success (Yang et al., 2023). Differentiable methods often train a neural network to produce augmentations (Lemley et al., 2017), sometimes in a generative adversarial setup (Shrivastava et al., 2017; Tran et al., 2017). By far the most notable method, AutoAugment (Cubuk et al., 2018), employs reinforcement learning. While RL is traditionally sample inefficient, improvements upon vanilla RL strategies have leveraged Bayesian methods (Lim et al., 2019), evolutionary strategies (Ho et al., 2019; Wang et al., 2023), or approximate gradient estimation for first-order optimization (Hataya et al., 2020).

Learning augmentation policies is especially challenging in low data settings, as full data policies are usually not transferable to the few-shot case. Various approaches have been considered, including proposing K-fold validation as a method of retaining the data while still performing validation (Naghizadeh et al., 2021). However, this method does not scale to one-shot settings. Utilizing clustering as a label-efficient evaluation method, where augmentations are designed to stay within their corresponding class cluster, can address this limitation (Abavisani et al., 2020).

Downloaded from direct.mit.edu/isal/proceedings-pdf/isal2026/38/124/2620452/isal.a.1050.pdf by guest on 31 October 2026

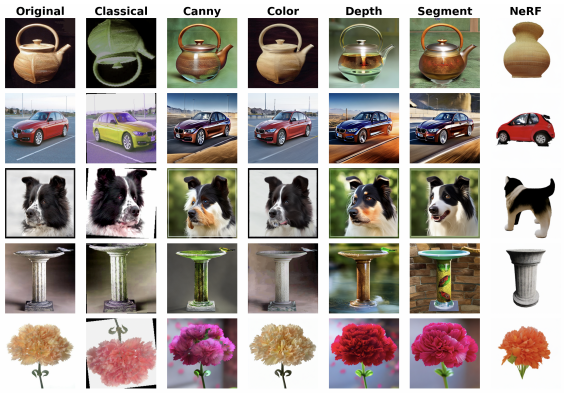


Figure 1: Example image augmentations using our pipeline. Classical augmentations include color jitter, rotation, and random cropping. Canny, color, depth, and segment use existing image information to steer a ControlNet diffusion model. NeRF uses a zero-shot NeRF to perform a 3D rotation.

Methods

Augmentation Operators

The generative augmentation operators are based on both diffusion and NeRFs. For diffusion-based operators, we use ControlNet (Zhang et al., 2023), an architecture which allows rapid customization of diffusion models without fine-tuning. To condition the model, we extract edges using Canny edge detection (Canny, 1986), segmentations using Segment Anything (Kirillov et al., 2023), depth maps using MiDaS (Ranftl et al., 2020), and color palettes by simply downsampling the image. This gives four diffusion-based augmentation operators, termed "Canny", "Segment", "Depth" and "Color". We use Zero123 (Liu et al., 2023b) for NeRF-based augmentation. This model creates a 3D reconstruction of an image from a single shot, allowing for 3D rotation. We then rotate 15 degrees left or right when performing an augmentation using this model. We term this operator "NeRF". Next, we include another augmentation operator, termed "Classical." This includes the full set of traditional augmentations: random crop, translation, scale, rotation, color jitter, and flip. This operator allows the evolution process to decide whether to include and build on the traditional classical augmentation pipeline or exclude it. Sometimes, all augmentations can be harmful, so we also included a "NoOp" operator that simply duplicates the existing image. Figure 1 gives examples of these operators.

Evolutionary Strategy

For our augmentation policy learning pipeline, we choose an evolutionary approach. This choice is motivated by practical considerations: diffusion and NeRF based augmentation is considerably more expensive to evaluate than traditional augmentations, so pipeline efficiency is crucial. Population-based evolutionary strategies have been shown to be as effective as RL approaches, with less than one percent of the computational effort (Ho et al., 2019). While gradient approximation methods have been shown to be even more efficient in some cases(Hataya et al., 2020), those results are for approximating gradients of simpler transformations, and do not translate to our pipeline, which can handle arbitrary generative modules. Further, recent work has shown evolution to be effective for searching for augmentation polices even in very complex augmentation spaces (Wang et al., 2023).

We define an augmentation tree as a binary tree, where each node represents an augmentation operator. The edges of our tree represent transition probabilities to each child node, summing to 1. This structure is chosen as it serves as a common genome for evolutionary algorithms.

Mutation Illustrated in Figure 2, mutation can occur at either the node level or the edge level. An edge mutation reassigns the probabilities of a transition between two child nodes. A node mutation switches the augmentation operator of that node (eg. Depth node becomes a Canny node).

Crossover Also illustrated in Figure 2, crossover is the other basic evolutionary operator. Two parents are selected, a child is created by splicing the branches of the parents together.

We thus define a population P of size n, of initial trees. In each generation, we use mutation and crossover to generate c children P_{new}, that are appended to P. Finally, the population is evaluated with a fitness function f, and the top n are kept for the next generation. Mutation and crossover probability are parameterized by p_m and p_c respectively. Algorithm 1 describes this process.

Fitness Functions

The goal of our augmentation strategy is to improve downstream model robustness, and thus the fitness function we choose to evaluate augmentation trees should either directly reflect what we seek to achieve or be a strong proxy. Note that in a full data setting, training data can be split into a train and validation. An augmentation tree can be evaluated by simply training a model with generated augmentations on the training data and measuring performance on the previously unseen evaluation data. We divide our discussion into two, more difficult, settings.

Low Data Setting In the low-data and few-shot case, the challenge becomes managing the noise of the evaluation function. We can no longer rely on a single train/val split to accurately measure the performance of a tree as low-data settings introduce high variability in splits. Thus, we use K-fold cross-validation.

In addition, directly using accuracy as our metric is no longer appropriate, as our validation set remains small enough that accuracy becomes coarse-grained and unstable. As a result, to align with the convention of higher fitness values corresponding to better candidates in the population,

Downloaded from direct.mit.edu/isal/proceedings-pdf/isal2026/38/124/2620452/isal_a_1050.pdf by guest on 31 October 2026

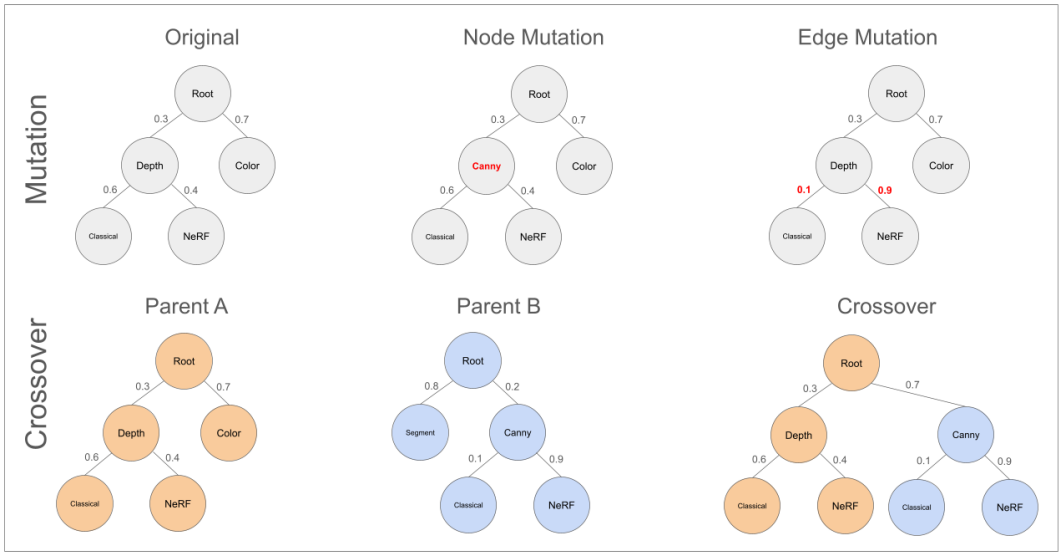


Figure 2: Mutation and Crossover for Augmentation Trees

we use the negative validation loss as the fitness function in these settings. Algorithm 2 describes this process. The pipeline can be seen in Figure 3a.

Algorithm 1 Evolutionary Search for Augmentation Trees

Require: Population size p, number of generations g, fitness function f, number of children c, mutation probability p_m, crossover probability p_c

$P \leftarrow \text{InitializePopulation}(p)$
for $i = 1$ to g **do**
  $P_{\text{new}} \leftarrow \text{MutateAndCrossover}(P, c, p_m, p_c)$
  $P \leftarrow P \cup P_{\text{new}}$
  Evaluate fitness $f(T)$ for each tree $T \in P$
  $P \leftarrow \text{SelectBest}(P, p)$ {Keep top p trees}
end for
$T_{\text{best}} \leftarrow \text{BestTree}(P)$
return T_{best}

One-Shot Setting In the most extreme case, we only have one image per class. Thus, proposed methods involving K-fold validation will not be able to span the full class range of the dataset (Naghizadeh et al., 2021). To address this problem, we devised the following strategies:

Label-Efficient Clustering Our goal is to find augmentations that preserve important class-specific characteristics while still providing novel data. Thus, when evaluating on a validation set is not possible, we can switch to a clustering approach. To find these novel, true-to-class augmentations, our intuition is to search for clusters that are wide, but still distinct from each other. Abavisani et al. proposed using this type of evaluation for augmentation pipelines in low-data

Algorithm 2 K-Fold Cross Validation Tree Fitness Function

Require: Dataset D, augmentation tree T, number of folds k

Split D into k folds: $D_1, D_2, \ldots, D_k$
Initialize $M \leftarrow 0$
for $i = 1$ to k **do**
  $D_{\text{val}} \leftarrow D_i$
  $D_{\text{train}} \leftarrow D \setminus D_i$
  $D_{\text{aug}} \leftarrow \text{ApplyAugmentationTree}(T, D_{\text{train}})$
  Train model M_i on D_{aug}
  $m_i \leftarrow \text{Evaluate}(M_i, D_{\text{val}})$
  $M \leftarrow M + m_i$
end for
$\bar{m} \leftarrow \frac{M}{k}$
return $\bar{m}$

and one-shot settings (Abavisani et al., 2020). They adopted Deep Subspace Clustering (Ji et al., 2017) and optimized the Silhouette coefficient as a measure of cluster quality. We improve upon this work in three ways:

1. We simplify the clustering process by using a pre-trained network to generate image embeddings which we then cluster, thus eliminating the need for a Deep Subspace Clustering network and requiring no training.

2. Prior work employed k-means to form clusters (Douzas et al., 2018), adding computational complexity. We simplify this by directly using known class labels as clusters. This allows us to evaluate explicitly whether augmentations form meaningful, class-based clusters rather than merely measuring separability.

Downloaded from direct.mit.edu/isal/proceedings-pdf/isal2026/38/124/2620452/isal.a.1050.pdf by guest on 31 October 2026

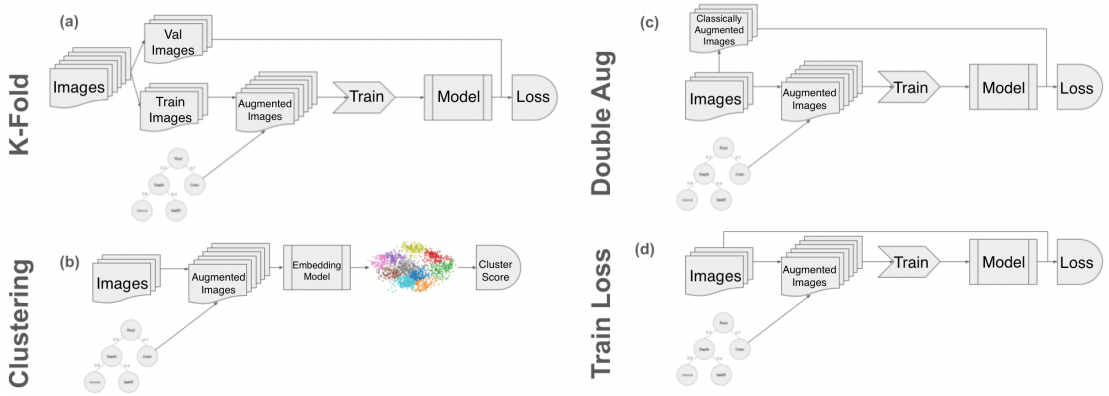


Figure 3: Tree Learning Pipelines. (a) K-Fold applies when there is more than one example per class. (b) We can measure cluster quality for the 1-shot case. (c) We can duplicate the image and assume the problem to be 2-shot instead of 1-shot (d) We can simply use training loss, though it is risky to assume that lower train loss equates to better performance.

3. When evaluating augmentation quality via clustering, traditional metrics like the Silhouette coefficient reward cohesion but do not penalize small or redundant clusters. This can cause the evolutionary algorithm to favor augmentation trees that produce minimal or trivial variations, which lack diversity and generalization potential. To avoid this pitfall, we introduce an additional penalty term based on average cluster radius, balancing cohesion with cluster size and separability. This modified metric thus encourages the formation of clusters that are both cohesive and sufficiently distinct, promoting better generalization.

This process is given in Algorithm 3. The pipeline can be seen in Figure 3b.

Double Augmentation This strategy is simple yet effective. We apply classical augmentations—which reliably introduce meaningful variations—to expand the original one-shot dataset. The augmented dataset is then divided into k splits, and the negative validation losses are averaged across splits, as detailed in Algorithm 4 and illustrated in Figure 3c. This approach allows us to increase augmentations while minimizing the risk of degrading dataset quality or relevance through unintended variations introduced by generative models.

Algorithm 4 1-Shot Double Augmentation Fitness Function

Require: One-shot dataset D, augmentation tree T, number of folds k

$D' \leftarrow \emptyset$
for each image $x \in D$ **do**
  $A(x) = \{\text{ClassicAug}(x)_1, \ldots, \text{ClassicAug}(x)_k\}$
  $D' \leftarrow D' \cup A(x)$
end for
return KFoldFitness(D', T, k) {Refer to Alg. 2}

Training Loss We can also simply use training loss as a proxy in the one-shot case. We augment all the images, and train a model. We then evaluate trees based on how low the training loss is after a fixed number of epochs. While this should encourage minor augmentations, and also makes use of train loss to estimate eval loss, a very erroneous assumption, it still works well in practice. The pipeline can be seen in Figure 3d.

Algorithm 3 1-Shot Clustering Fitness Function

Require: Image dataset D, augmentation tree T, embedding model E

$D_{\text{aug}} \leftarrow \text{ApplyAugmentationTree}(T, D)$
Initialize embedding list $L \leftarrow \emptyset$
for each image $x \in D_{\text{aug}}$ **do**
  $e \leftarrow E(x)$
  Append e to L
end for
$C \leftarrow \text{Cluster}(L)$
$S \leftarrow \text{ComputeSilhouetteScore}(C)$
$d \leftarrow \text{ComputeMeanClusterDistance}(C)$
$s \leftarrow \alpha S - \frac{1-\alpha}{d}$
return s

Results

Experiment Setup

We perform our experiments on six datasets: Caltech256 (Griffin et al., 2007), Oxford IIIT-Pets (Parkhi et al., 2012), Oxford 102 Flowers (Nilsback and Zisserman, 2008), Stanford Cars (Krause et al., 2013), Stanford Dogs (Khosla et al., 2011), and Food101 (Bossard et al., 2014). To highlight how powerful our method is, even in few-shot settings when the fine-grained semantic distinctions are minor, we *deliberately searched* for few-shot images and classes that were the most challenging.

For an n-way k-shot classification task, we do this as follows. First, we randomly select n classes from the original

Downloaded from direct.mit.edu/isal/proceedings-pdf/isal2026/38/124/2620452/isal_a_1050.pdf by guest on 31 October 2026

Table 1: 5-way, 2-shot classification accuracy (%) with standard deviation across 6 datasets and 3 downstream image classification architectures. Bolded values indicate the best performance per row.

Dataset	Model	Naive Baseline	Random Tree	RandAugment	AutoAugment	Learned
Caltech256	ResNet50	79.78 ± 0.73	81.42 ± 7.64	84.71 ± 0.01	86.78 ± 0.00	**88.28 ± 1.75**
	MobileNet	80.95 ± 1.08	-	**86.04 ± 0.01**	84.45 ± 0.01	84.18 ± 0.75
	ViT-Small	73.30 ± 6.61	-	81.85 ± 0.01	**81.74 ± 0.02**	79.83 ± 9.90
Flowers102	ResNet50	70.49 ± 1.08	78.59 ± 2.11	83.65 ± 0.00	**86.60 ± 0.00**	73.73 ± 3.70
	MobileNet	77.00 ± 1.83	-	**79.54 ± 0.01**	81.75 ± 0.01	73.21 ± 4.75
	ViT-Small	97.89 ± 1.43	-	**99.37 ± 0.00**	98.63 ± 0.00	94.30 ± 3.85
Stanford Dogs	ResNet50	78.44 ± 0.13	83.76 ± 6.42	80.76 ± 0.01	82.23 ± 0.00	**85.15 ± 2.73**
	MobileNet	75.34 ± 0.99	-	73.72 ± 0.02	75.26 ± 0.00	**77.51 ± 0.63**
	ViT-Small	83.67 ± 1.38	-	**86.95 ± 0.02**	83.97 ± 0.02	80.61 ± 2.01
Stanford Cars	ResNet50	30.90 ± 1.68	36.94 ± 6.48	37.20 ± 0.01	34.68 ± 0.01	**40.40 ± 3.07**
	MobileNet	35.35 ± 1.05	-	36.30 ± 0.01	36.95 ± 0.01	**37.36 ± 0.15**
	ViT-Small	40.64 ± 5.43	-	43.83 ± 0.01	42.74 ± 0.02	**46.32 ± 1.43**
Oxford-IIIT Pet	ResNet50	86.57 ± 0.60	84.97 ± 3.08	85.25 ± 0.01	86.57 ± 0.00	**88.34 ± 1.72**
	MobileNet	84.41 ± 0.53	-	87.27 ± 0.01	86.08 ± 0.00	**89.21 ± 2.93**
	ViT-Small	88.52 ± 0.55	-	91.28 ± 0.01	90.60 ± 0.01	**91.44 ± 2.06**
Food101	ResNet50	47.82 ± 0.57	42.61 ± 4.56	46.23 ± 0.00	**51.32 ± 0.00**	49.78 ± 4.21
	MobileNet	39.93 ± 1.46	-	**43.49 ± 0.00**	44.82 ± 0.00	42.97 ± 2.61
	ViT-Small	55.66 ± 3.22	-	62.20 ± 0.02	64.14 ± 0.02	**64.18 ± 2.13**

dataset. Then we randomly selected k images from each class. We fine-tune a pretrained Resnet50 model (He et al., 2016) on these images and record the accuracy. We repeat this procedure 10 times, gathering 10 different subsets of the classes with different images for each dataset. Afterwards, we note which subset of classes from the dataset had the lowest baseline test accuracy, and we choose this subset as the setting for our augmentation benchmarks.

For our genetic algorithm, we initialize a population of 14. For each of the seven augmentation operators, we initialize two trees whose root nodes use that operator, creating a balanced population. This broadens the solution space exploration and avoids the pitfalls of random initialization on a small population. We set the mutation probability to 10% and include 6 crossovers per generation. We restrict tree depth to 2, allowing the composition of at most 2 operations per augmentation. For each of the 10 generations, we generate 8 children. In the 2 and 5-shot cases, we use K-fold fitness, choosing folds such that the classes remained balanced. To evaluate augmentation trees, we train the models for 20 epochs and observe the corresponding loss. In the one-shot case, we examine the three other fitness functions (double augmentation, training loss, and clustering) proposed above.

Once the best tree is chosen, we generate augmentations and evaluate the downstream classification accuracy against several baselines:

1. Naive Baseline: We randomly apply classical augmentations (cropping, scaling, translation, horizontal/vertical flipping, color jitter, rotation)

2. RandAugment: We perform a grid search over the number of operations (num_ops) and magnitude parameters, selecting the configuration with the lowest validation loss using cross-validation on a train/validation split; this best-performing configuration is then evaluated on the full test set.

3. AutoAugment: We apply the ImageNet-learned AutoAugment policy to our datasets.

In all downstream classification tasks, training proceeds for 200 epochs. In the 1-shot setting, we augment each image in the original dataset 2 times, in the 2-shot setting 5 times, and in the 5-shot setting 2 times. We also evaluate our methods and baselines against augmentations generated from random trees in the ResNet experiments to ensure that our evolutionary search was an important part of creating true-to-class augmentations. Each experiment is performed at least three times with varying seeds, and the average and standard deviation are reported. We evaluate using a pre-trained ResNet50, ViT-Small (Dosovitskiy et al., 2021), or MobileNetV2 (Sandler et al., 2019) model. Models are fine-tuned using Adam (Kingma and Ba, 2017), with a learning rate of 1e-3. We use NVIDIA GeForce RTX 4090 chips with 24 GB of memory. Each experiment took between 2 and 24 hours to complete, depending on the number of ways and shots.

Downloaded from direct.mit.edu/isal/proceedings-pdf/isal2026/38/124/2620452/isal.a.1050.pdf by guest on 31 October 2026

Table 2: 5-way, 5-shot classification accuracy (%) with standard deviation across 6 datasets and 3 downstream image classification architectures. Bolded values indicate the best performance per row.

Dataset	Model	Naive Baseline	Random Tree	RandAugment	AutoAugment	Learned
Caltech256	ResNet50	88.15 ± 0.25	92.22 ± 0.93	92.55 ± 0.00	**93.31 ± 0.00**	91.48 ± 1.11
	MobileNet	88.80 ± 0.19	-	89.51 ± 0.00	**91.41 ± 0.00**	90.05 ± 0.98
	ViT-Small	85.75 ± 1.04	-	90.32 ± 0.01	90.70 ± 0.01	**92.33 ± 1.27**
Flowers102	ResNet50	82.61 ± 0.96	79.51 ± 2.69	89.04 ± 0.01	**89.92 ± 0.01**	84.95 ± 1.15
	MobileNet	88.82 ± 0.69	-	**91.47 ± 0.00**	89.26 ± 0.01	86.60 ± 1.07
	ViT-Small	99.78 ± 0.19	-	99.89 ± 0.00	**99.89 ± 0.00**	99.34 ± 0.33
Stanford Dogs	ResNet50	88.69 ± 0.65	90.81 ± 1.08	89.21 ± 0.00	91.24 ± 0.00	**91.35 ± 0.72**
	MobileNet	82.88 ± 0.72	-	81.49 ± 0.00	83.10 ± 0.00	**83.40 ± 0.27**
	ViT-Small	88.73 ± 0.46	-	**89.73 ± 0.00**	87.99 ± 0.00	84.66 ± 0.85
Stanford Cars	ResNet50	52.97 ± 0.80	53.75 ± 1.44	54.63 ± 0.00	51.48 ± 0.01	**57.98 ± 3.20**
	MobileNet	50.79 ± 1.39	-	55.41 ± 0.01	**55.93 ± 0.01**	48.87 ± 1.71
	ViT-Small	58.12 ± 2.73	-	63.00 ± 0.02	**66.67 ± 0.01**	59.25 ± 2.23
Oxford-IIIT Pet	ResNet50	92.07 ± 0.44	93.12 ± 0.74	92.91 ± 0.01	93.61 ± 0.00	**93.63 ± 0.43**
	MobileNet	88.56 ± 0.85	-	90.32 ± 0.00	90.14 ± 0.00	**90.53 ± 0.76**
	ViT-Small	94.95 ± 0.56	-	**95.44 ± 0.01**	95.37 ± 0.01	93.54 ± 0.53
Food101	ResNet50	54.09 ± 0.58	56.88 ± 2.60	56.72 ± 0.00	58.28 ± 0.00	**58.75 ± 1.60**
	MobileNet	51.88 ± 0.59	-	52.00 ± 0.00	54.05 ± 0.00	**54.19 ± 1.36**
	ViT-Small	75.40 ± 2.22	-	**79.59 ± 0.00**	78.25 ± 0.00	76.49 ± 0.75

Table 3: 5-way, 1-shot classification accuracy (%) with standard deviation across 6 datasets and 3 downstream image classification architectures. Bolded values indicate the best performance per row.

Dataset	Model	Naive Baseline	NoOp / Classical Tree	Random Tree	RandAugment	AutoAugment	Learned (Clustering)
Caltech256	ResNet50	65.77 ± 1.29	78.67 ± 2.00	81.57 ± 6.44	81.63 ± 0.01	82.92 ± 0.01	**83.65 ± 4.92**
	MobileNet	67.28 ± 2.56	-	-	71.97 ± 0.01	71.47 ± 0.01	**80.09 ± 4.73**
	ViT-Small	66.72 ± 3.53	-	-	75.60 ± 0.03	75.38 ± 0.03	**82.12 ± 3.64**
Flowers102	ResNet50	61.48 ± 0.78	66.15 ± 0.90	63.03 ± 3.95	63.97 ± 0.00	65.84 ± 0.01	**66.75 ± 2.34**
	MobileNet	57.11 ± 1.30	-	-	53.17 ± 0.01	58.46 ± 0.01	**60.78 ± 2.58**
	ViT-Small	94.60 ± 1.88	-	-	**96.26 ± 0.03**	93.98 ± 0.02	95.47 ± 2.19
Stanford Dogs	ResNet50	70.30 ± 0.58	75.79 ± 0.29	76.58 ± 3.84	75.86 ± 0.01	77.22 ± 0.02	**78.86 ± 3.21**
	MobileNet	60.58 ± 2.66	-	-	65.19 ± 0.02	67.73 ± 0.01	**69.70** ± 2.37
	ViT-Small	75.55 ± 1.61	-	-	78.47 ± 0.01	77.83 ± 0.02	**79.70 ± 3.12**
Stanford Cars	ResNet50	21.31 ± 0.80	28.11 ± 0.43	29.77 ± 1.83	**32.84 ± 0.01**	31.84 ± 0.03	29.66 ± 2.62
	MobileNet	30.43 ± 1.12	-	-	30.18 ± 0.01	**30.35 ± 0.02**	29.05 ± 3.18
	ViT-Small	31.10 ± 5.56	-	-	36.15 ± 0.02	34.66 ± 0.01	**37.35 ± 3.37**
Oxford-IIIT Pet	ResNet50	79.68 ± 1.50	82.44 ± 0.55	81.47 ± 6.34	78.17 ± 0.01	82.71 ± 0.00	**86.16 ± 1.19**
	MobileNet	72.18 ± 1.45	-	-	76.31 ± 0.00	74.17 ± 0.01	**80.43 ± 1.87**
	ViT-Small	76.10 ± 5.44	-	-	83.88 ± 0.02	79.61 ± 0.04	**84.58 ± 3.22**
Food101	ResNet50	30.90 ± 0.57	30.06 ± 0.31	32.83 ± 2.42	30.38 ± 0.00	30.46 ± 0.00	**34.28 ± 0.83**
	MobileNet	28.78 ± 0.63	-	-	25.52 ± 0.01	26.15 ± 0.01	**34.61 ± 1.74**
	ViT-Small	43.74 ± 2.51	-	-	**48.17 ± 0.02**	45.44 ± 0.01	44.89 ± 2.30

Few-Shot Results

The few-shot results are shown in Tables 1 and 2. We measure the accuracy on the test set for models trained using the baseline strategies, random augmentation trees, and the augmentation trees learned from our pipeline. While EvoAug consistently outperforms the Naive Baseline, results are mixed when evaluated against AutoAugment and RandAugment. Notably, EvoAug is much better on the Stanford Dogs and Oxford-IIIT Pets datasets, but marginally worse on Flowers102.

One-Shot Results

Our 1-shot results are shown in Table 3. Here, we include our clustering-based fitness function learning strategy. EvoAug consistently outperforms the Naive Baseline, and often outperforms RandAugment and AutoAugment, achieving strong performance in scarce data settings. We also run our pipeline restricting nodes to just classical or NoOp transformations and find that these restricted trees perform worse than our normal trees. This supports the conclusion that generative augmentation operators are an important part of performance.

Downloaded from direct.mit.edu/isal/proceedings-pdf/isal2026/38/124/2620452/isal_a_1050.pdf by guest on 31 October 2026

Table 4: Accuracy on full datasets for various models

Dataset	Setting	RN50	RN101	VGG19	EN	Visformer	Swin	MN	DN
Caltech	Baseline	72.37	73.62	67.40	71.79	68.83	63.95	66.48	75.74
	Conditioned	**76.49**	**77.64**	**70.82**	**73.85**	**73.15**	**69.55**	**68.33**	**78.10**
Cars	Baseline	86.78	88.16	87.22	86.75	83.37	75.43	80.80	91.08
	Conditioned	**91.02**	**90.95**	**89.61**	**88.56**	**87.40**	**82.32**	**82.70**	**92.20**
Aircraft	Baseline	75.23	75.91	**88.80**	81.25	72.61	60.88	70.24	80.53
	Conditioned	**82.33**	**81.10**	88.20	**81.76**	**74.67**	**71.74**	**74.17**	**83.29**
Dogs	Baseline	66.49	70.15	**68.63**	**64.17**	**64.65**	52.10	**58.60**	**70.44**
	Conditioned	**68.74**	**70.40**	66.05	62.45	64.36	**56.50**	58.30	70.21
Pets	Baseline	69.22	70.72	**83.17**	73.59	73.02	58.54	67.35	**80.16**
	Conditioned	**71.07**	**74.03**	81.28	**74.41**	**76.24**	**61.00**	**68.46**	79.34

Full Dataset Results

While the main body of our work focuses on the few shot setting, there are also experiments done which have indicated that conditioned generation is beneficial in the full dataset settingGoldfeder et al. (2024a). The method used in these experiments employs LLaVa2 (Liu et al., 2023a) generated captions to condition the augmentation of images in the dataset. We believe that with more intelligent conditioning (by learning augmentation trees which match the dataset), we can achieve better performance.

We reproduce the relevant summary statistics below in Table 4 for completeness. The results show that conditioned generation consistently achieves higher accuracy than a classically augmented baseline across six datasets: Caltech256 (Griffin et al., 2007), Stanford Cars (Krause et al., 2013), FGVC Aircraft (Maji et al., 2013), Stanford Dogs (Khosla et al., 2011), Oxford IIIT-Pets (Parkhi et al., 2012); and eight model architectures: ResNet (He et al., 2016), VGG (Simonyan and Zisserman, 2014), EfficientNet (Tan and Le, 2019), Visformer (Chen et al., 2021), Swin Transformer (Liu et al., 2021), MobileNet (Howard et al., 2017), DenseNet (Iandola et al., 2014), and ViT (Dosovitskiy et al., 2021).

Conclusion

We have argued that imagination is a fundamental mechanism of biological adaptation, and that its computational analog requires two components: a generative world model and a selection process that keeps simulations grounded. EvoAug instantiates this principle in the visual domain, pairing controlled diffusion and NeRF operators with evolutionary search to discover task-specific augmentation strategies in few-shot and one-shot settings. The results support the thesis. Generative augmentation without evolutionary grounding is unreliable, as random trees demonstrate: unguided imagination is as likely to harm as to help. Evolutionary search alone, operating over classical augmentations only, produces weaker results than the full pipeline. It is their combination that consistently improves adaptation. Most tellingly, the system recovers domain-relevant structure without supervision: in the Flowers102 analysis, evolution discovers that color is an essential property of flowers and viewpoint is accidental, a distinction that aligns with biological intuition but was never explicitly provided. These findings suggest that the imagination framing is not merely a rhetorical device but a productive lens for understanding when and why generative augmentation works. We hope this perspective encourages the artificial life community to examine world models not only as engineering tools but as computational models of a capacity that has been central to biological adaptation for far longer than AI has existed.

Acknowledgments We thank the NSF AI Institute in Dynamic Systems for funding our work (grant #2112085).

Downloaded from direct.mit.edu/isal/proceedings-pdf/isal2026/38/124/2620452/isal.a.1050.pdf by guest on 31 October 2026

References

Abavisani, M., Naghizadeh, A., Metaxas, D., and Patel, V. (2020). Deep subspace clustering with data augmentation. *Advances in Neural Information Processing Systems*, 33:10360–10370.

Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., and Fleet, D. J. (2023). Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*.

Besnier, V., Jain, H., Bursuc, A., Cord, M., and Pérez, P. (2020). This dataset does not exist: training models from generated images. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Bossard, L., Guillaumin, M., and Van Gool, L. (2014). Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer.

Brock, A., Donahue, J., and Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698.

Chen, P., Liu, S., Zhao, H., and Jia, J. (2020). Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*.

Chen, Z., Xie, L., Niu, J., Liu, X., Wei, L., and Tian, Q. (2021). Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 589–598.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2018). Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. (2020). Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 702–703.

DeVries, T. and Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

Douzas, G., Bacao, F., and Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. *Information Sciences*, 465:1–20.

Engelsma, J. J., Grosz, S., and Jain, A. K. (2022). Printsgan: synthetic fingerprint generator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6111–6124.

Fang, H., Han, B., Zhang, S., Zhou, S., Hu, C., and Ye, W.-M. (2024). Data augmentation for object detection via controllable diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1257–1266.

Figueira, A. and Vaz, B. (2022). Survey on synthetic data generation, evaluation methods and gans. *Mathematics*, 10(15):2733.

Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.-Y., Cubuk, E. D., Le, Q. V., and Zoph, B. (2021). Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2918–2928.

Goldfeder, J., Wyder, P., LeCun, Y., and Ziv, R. S. (2026). Ai must embrace specialization via superhuman adaptable intelligence. *arXiv preprint arXiv:2602.23643*.

Goldfeder, J. A., Puma, P. M., Guo, G., Trigo, G. G., and Lipson, H. (2024a). Learning via imagination: Controlled diffusion image augmentation. In *NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward*.

Goldfeder, J. A., Puma, P. M., Guo, G., Trigo, G. G., and Lipson, H. (2024b). Self supervised learning using controlled diffusion image augmentation. In *NeurIPS 2024 Workshop: Self-Supervised Learning-Theory and Practice*.

Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset.

Ha, D. and Schmidhuber, J. (2018). World models. *arXiv preprint arXiv:1803.10122*, 2(3):440.

Hassabis, D., Kumaran, D., Summerfield, C., and Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258.

Hataya, R., Zdenek, J., Yoshizoe, K., and Nakayama, H. (2020). Faster autoaugment: Learning augmentation strategies using backpropagation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 1–16. Springer.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, R., Sun, S., Yu, X., Xue, C., Zhang, W., Torr, P., Bai, S., and Qi, X. (2022). Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*.

Ho, D., Liang, E., Chen, X., Stoica, I., and Abbeel, P. (2019). Population based augmentation: Efficient learning of augmentation policy schedules. In *International conference on machine learning*, pages 2731–2741. PMLR.

Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. (2021). Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Downloaded from direct.mit.edu/isal/proceedings-pdf/isal2026/38/124/2620452/isal.a.1050.pdf by guest on 31 October 2026

Huang, W., Chao, Y.-W., Mousavian, A., Liu, M.-Y., Fox, D., Mo, K., and Fei-Fei, L. (2026). Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*.

Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., and Keutzer, K. (2014). Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*.

Islam, K. and Akhtar, N. (2025). Context-guided responsible data augmentation with diffusion models.

Islam, K., Zaheer, M. Z., Mahmood, A., and Nandakumar, K. (2024). Diffusemix: Label-preserving data augmentation with diffusion models.

Islam, K., Zaheer, M. Z., Mahmood, A., Nandakumar, K., and Akhtar, N. (2025). Genmix: Effective data augmentation with generative diffusion model image editing.

Jahanian, A., Puig, X., Tian, Y., and Isola, P. (2021). Generative models as a data source for multiview representation learning. *arXiv preprint arXiv:2106.05258*.

Ji, P., Zhang, T., Li, H., Salzmann, M., and Reid, I. (2017). Deep subspace clustering networks. *Advances in neural information processing systems*, 30.

Khosla, A., Jayadevaprakash, N., Yao, B., and Li, F.-F. (2011). Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2. Citeseer.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. (2023). Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. (2013). 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.

LeCun, Y. et al. (2022). A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62.

Lemley, J., Bazrafkan, S., and Corcoran, P. (2017). Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, 5:5858–5869.

Li, P., Li, X., and Long, X. (2020). Fencemask: a data augmentation approach for pre-extracted image features. *arXiv preprint arXiv:2006.07877*.

Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. (2019). Fast autoaugment. *Advances in neural information processing systems*, 32.

Liu, H., Li, C., Li, Y., and Lee, Y. J. (2023a). Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.

Liu, R., Wu, R., Hoorick, B. V., Tokmakov, P., Zakharov, S., and Vondrick, C. (2023b). Zero-1-to-3: Zero-shot one image to 3d object.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. (2013). Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. (2021). Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*.

Naghizadeh, A., Metaxas, D. N., and Liu, D. (2021). Greedy autoaugmentation for n-shot learning using deep neural networks. *Neural Networks*, 135:68–77.

Narasimhaswamy, S., Bhattacharya, U., Chen, X., Dasgupta, I., Mitra, S., and Hoai, M. (2024). Handiffuser: Text-to-image generation with realistic hand appearances. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 2468–2479. IEEE.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (2021). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Nilsback, M.-E. and Zisserman, A. (2008). Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. (2012). Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE.

Pearson, J. (2019). The human imagination: the cognitive neuroscience of visual mental imagery. *Nature reviews neuroscience*, 20(10):624–634.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. arxiv 2022. *arXiv preprint arXiv:2204.06125*.

Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., and Koltun, V. (2020). Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637.

Razavi, A., Van den Oord, A., and Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32.

Robertson Ishii, T. and Atkins, P. (2008). Essential vs. accidental properties.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Downloaded from direct.mit.edu/isal/proceedings-pdf/isal2026/38/124/2620452/isal.a.1050.pdf by guest on 31 October 2026

Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. (2022a). Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. (2022b). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2019). Mobilenetv2: Inverted residuals and linear bottlenecks.

Sarıyıldız, M. B., Alahari, K., Larlus, D., and Kalantidis, Y. (2023). Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8011–8021.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. (2022). Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294.

Shipard, J., Wiliem, A., Thanh, K. N., Xiang, W., and Fookes, C. (2023). Diversity is definitely needed: Improving model-agnostic zero-shot classification via stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 769–778.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. (2017). Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Skandarani, Y., Jodoin, P.-M., and Lalande, A. (2023). Gans for medical image synthesis: An empirical study. *Journal of Imaging*, 9(3):69.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

Trabucco, B., Doherty, K., Gurinas, M., and Salakhutdinov, R. (2023). Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*.

Trabucco, B., Doherty, K., Gurinas, M., and Salakhutdinov, R. (2025). Effective data augmentation with diffusion models.

Tran, T., Pham, T., Carneiro, G., Palmer, L., and Reid, I. (2017). A bayesian data augmentation approach for learning deep models. *Advances in neural information processing systems*, 30.

Wang, G., Cubuk, E. D., Rosenberg, A., Cheng, S., Weiss, R. J., Ramabhadran, B., Moreno, P. J., Le, Q. V., and Park, D. S. (2023). G-augment: Searching for the meta-structure of data augmentation policies for asr. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 23–30. IEEE.

Yamaguchi, S. and Fukuda, T. (2023). On the limitation of diffusion models for synthesizing training datasets. *arXiv preprint arXiv:2311.13090*.

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., and Yang, M.-H. (2025). Diffusion models: A comprehensive survey of methods and applications.

Yang, Z., Sinnott, R. O., Bailey, J., and Ke, Q. (2023). A survey of automated data augmentation algorithms for deep learning-based image classification tasks. *Knowledge and Information Systems*, 65(7):2805–2861.

Yu, Z., Zhu, C., Culatana, S., Krishnamoorthi, R., Xiao, F., and Lee, Y. J. (2023). Diversify, don't fine-tune: Scaling up visual recognition training with synthetic images. *arXiv preprint arXiv:2312.02253*.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, L., Rao, A., and Agrawala, M. (2023). Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847.

Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. (2020). Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008.

Zhou, Y., Sahak, H., and Ba, J. (2023). Training on thin air: Improve image classification with generated data. *arXiv preprint arXiv:2305.15316*.

Downloaded from direct.mit.edu/isal/proceedings-pdf/isal2026/38/124/2620452/isal.a.1050.pdf by guest on 31 October 2026