DiffAug: A Diffuse-and-Denoise Augmentation for Training Robust Classifiers

Chandramouli Shama Sastry, Sri Harsha Dumpala, Sageev Oore

NEURAL INFORMATION PROCESSING SYSTEMS

DALHOUSIE UNIVERSITY

Generative Data Augmentation with Diffusion Models

Recent works capitalize on the photorealistic image generation abilities of text-to-image diffusion models for data augmentation [1,3]. Successful diffusion-augmentation techniques require:

- **Extensive Compute**: for high-quality samples and hyperparameter fine-tuning (prompt, guidance-strength, etc). Longer sampling times necessitate offline data augmentation.
- **Massive training datasets** (e.g., LAION-5B) to create the diffusion model [2]. Previous works show ***limited gains from diffusion models trained with no extra data***.

These limitations give rise to the following motivations:

1. **Synthetic data-augmentation without extra data**: generative model and classifier share the same dataset (e.g., ImageNet).
2. **Efficient and Effective Augmentation**: efficient augmentation technique that can be performed within the train-loop but also remains effective.

Diffusion Models

Forward-diffusion stochastic process $\{\mathbf{x}_t\}_{t\in[0,T]}$ starts at data, $\mathbf{x}_0$, and ends at noise, $\mathbf{x}_T$:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\, dt + g(t)\, d\mathbf{w}, \tag{1}$$

where $\mathbf{w}$ denotes a standard Wiener process, $\mathbf{f}(\mathbf{x}(t), t)$ is a drift coefficient, and $g(t)$ is a diffusion coefficient.

Reverse-diffusion: To sample from $p_0(\mathbf{x})$ starting with samples from $p_T(\mathbf{x})$, we have to solve the reverse diffusion SDE (**?**):

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\, dt + g(t)\, d\bar{\mathbf{w}}, \tag{2}$$

where $d\bar{\mathbf{w}}$ is a standard Wiener process when time flows from T to 0, dt is an infinitesimal negative timestep, and the score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is estimated by $s_\theta(\mathbf{x}, t)$ trained using score-matching loss.

Highlights

- DiffAug is a simple and efficient diffusion-based augmentation technique.
- Has connections with MixUp and can be viewed as a type of vicinal-risk minimization.
- Enhances classifier robustness while preserving test accuracy and can be used at both train-time and test-time!
- No hyperparameter-tuning necessary.
- Both unconditional and conditional diffusion models can be used! Most of our experiments here are reported with unconditional diffusion models.
- Can be combined with popular augmentation-techniques such as AugMix and DeepAugment for extended robustness.
- We discover perceptually-aligned gradients (PAGs) when classifying DiffAug examples and offer theoretical explanations. Also, PAGs $\iff$ Robustness [4].
- We also achieve improvements in classifier-guided diffusion by training improved and robust classifiers for *qualitatively improved* guidance.

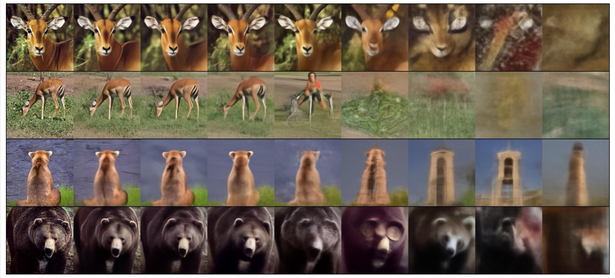
References

[1] Synthetic Data from Diffusion Models Improves ImageNet Classification. *TMLR 2023.*
[2] The Unmet Promise of Synthetic Training Images: Using Retrieved Real Images Performs Better. *NeurIPS 2024.*
[3] Leaving reality to imagination: Robust classification via generated datasets. *Trustworthy-ML workshop at ICLR 2023.*
[4] Do perceptually aligned gradients imply adversarial robustness? *ICML 2024.*
[5] Back to the Source: Diffusion-Driven Test-Time Adaptation. *CVPR 2023.*
[6] OpenOOD: Benchmarking Generalized OOD Detection. *NeurIPS D&B 2022.*
[7] (Certified!!) Adversarial Robustness for Free! *ICLR 2023.*

DiffAug: Diffuse-and-Denoise Augmentation

Given a sample $\mathbf{x}_0$, DiffAug first diffuses the sample to a random time $t \in [0, T]$ to generate $\mathbf{x}_t \sim p_t(\mathbf{x}_t|\mathbf{x}_0)$ and then applies a single reverse diffusion step (i.e., denoising):

$$\textsc{DiffAug}\,(\mathbf{x}_0, t) = \hat{\mathbf{x}}_t = \mathbf{x}_t + \sigma^2(t)\, s_\theta(\mathbf{x}_t, t) \tag{3}$$



For four original training examples ($\mathbf{x}_0$) in the leftmost column, we display 8 random augmentations ($\hat{\mathbf{x}}_t$) for each image between $t = 350$ and $t = 700$ in steps of size 50. Augmentations generated for $t < 350$ are *closer* to the input image while the augmentations for $t > 700$ are *farther* from the input image. We observe that DiffAug using larger values of t does not preserve the class label introducing noise in the training procedure. However, we find that this does not lead to lower classification accuracy but instead contributes to improved robustness.

DiffAug-Ensemble: DiffAug for Test-Time Adaptation

- Apply DiffAug at Test-Time to address covariate-shifts.
- Diffuse-and-Denoise can be viewed as projection towards the data manifold.
- Given classifier p_ϕ and a set of diffusion times $S = \{0, 50 \ldots 450\}$

$$p(y|\mathbf{x}_0) = \frac{1}{|\mathcal{S}|} \sum_{t\in\mathcal{S}} p_\phi(y|\hat{\mathbf{x}}_t)$$

- Diffusion-Driven Adaptation (DDA) is based on this idea [5].
- DiffAug-Ensemble is **10x faster** than DDA and is generally better or remains competitive.

Augmenting DiffAug!

- DiffAug remains effective even when high-quality synthetic data is available.
- For analysis, we consider a synthetic clone of ImageNet generated by Bansal & Grover [3].

	Diffusion Model	Training Data	Augmentation within Train-loop
+Synth	Stable-Diffusion	LAION-5B	✗
+DiffAug	Improved-DDPM	ImageNet-1.3M	✓

Table: Top-1 Accuracy (%) across different distribution shifts when additional high-quality synthetic data is available (+Synth). Net improvement with DiffAug training and DiffAug-Ensemble (DE) inference is shown.

Model	ImageNet-C (Severity=5)	ImageNet-R	ImageNet-S	ImageNet Sketch	ImageNet-A	ImageNet-D	Average
RN50	17.87	36.16	7.12	24.09	0.03	11.36	16.10
+DiffAug/DE	**32.22** (+14.85)	**41.61** (+5.45)	**12.52** (+5.40)	**26.67** (+2.56)	**1.09** (+1.06)	11.37 (+0.01)	**20.90**
RN50+Synth	17.58	49.28	7.68	35.45	0.63	17.52	21.35
+DiffAug/DE	**30.06** (+12.48)	**54.71** (+5.43)	**13.57** (+5.89)	**37.39** (+1.94)	**1.53** (+0.9)	**21.41** (+3.89)	**26.45**

Experiments: Classifier Robustness

Robustness to Covariate Shifts

We consider the following evaluation Modes:

(a) **Default**: Direct evaluation on test examples.
(b) **DDA**: A diffusion-based test-time image-adaptation technique.
(c) **DDA-SE**: Average of prediction on both DDA-adapted and original test example.
(d) **DiffAug-Ensemble (DE)**: Average of predictions on a set of test-time DiffAug augmentations.

Train Augmentations	ImageNet-C (severity = 5) DDA	DDA (SE)	DE	Def.	**Avg**	ImageNet-Test DDA	DDA (SE)	DE	Def.	**Avg**
AugMix (AM)	33.18	36.54	34.08	26.72	32.63	62.23	75.98	73.8	77.53	72.39
AM+DiffAug	34.64	38.61	38.58	29.47	**35.33**	63.53	76.09	75.88	77.34	**73.21**
DeepAugment (DA)	35.41	39.06	37.08	31.93	35.87	63.63	75.39	74.28	76.65	72.49
DA+DiffAug	37.61	41.31	40.42	33.78	**38.28**	65.47	75.54	75.43	76.51	**73.24**
DA+AM (DAM)	40.36	44.81	41.86	39.52	41.64	65.54	74.41	73.54	75.81	72.33
DAM+DiffAug	41.91	46.35	44.77	41.24	**43.57**	66.83	74.64	74.39	75.66	**72.88**
RN50	28.35	30.62	27.12	17.87	25.99	58.09	74.38	71.43	76.15	70.01
RN50+DiffAug	31.15	33.51	32.22	20.87	**29.44**	61.04	74.87	75.07	75.95	**71.73**
ViT-B/16	43.6	52.9	48.25	50.75	48.88	67.4	81.72	80.43	83.71	78.32
ViT-B/16+DiffAug	45.05	53.54	51.87	52.78	**50.81**	70.05	81.85	82.59	83.59	**79.52**
Avg	37.13	41.73	39.63	34.49	38.24	64.38	76.49	75.68	77.89	73.61
Avg (No-DiffAug)	36.18	40.79	37.68	33.36	37.00	63.38	76.38	74.70	77.97	73.11
Avg (DiffAug)	38.07	42.66	41.57	35.63	39.48	65.38	76.60	76.67	77.81	74.12

Table: Top-1 Accuracy on Imagenet-C and Imagenet-Test.

(a) No degradation in clean test accuracy despite training directly on partially synthesized examples.
(b) DiffAug improves performance across all evaluation modes.
(c) DiffAug models with DE improve over Non-DiffAug models with DDA-SE, in terms of both accuracy and wallclock time.

OOD Detection

Train Augmentation	ASH	MSP	ReAct	Scale	**Avg.**
AugMix(AM)	82.16	77.49	79.94	83.61	80.8
AM+DiffAug	83.62	78.35	81.29	84.81	**82.02**
RN50	78.17	76.02	77.38	81.36	78.23
RN50+DiffAug	79.86	76.86	78.76	82.81	**79.57**

Table: AUROC on Imagenet Near-OOD Detection.

(a) Amongst existing augmentation techniques, AugMix offers the best OOD detection rate and AugMix/ASH is placed 3rd out of 73 methods in the OpenOOD leaderboard [6].
(b) DiffAug introduces further improvements!
(c) AM+DiffAug/Scale is placed second on the leaderboard and comparable to the top AUROC of 84.87.
(d) Augmentation combinations that improve covariate shift robustness do not necessarily improve OOD detection (e.g., DAM).
(e) DiffAug improves OOD detection *and* covariate shift robustness for all tested augmentations.

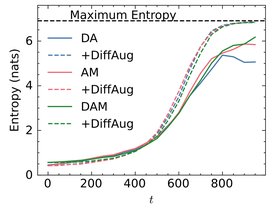


Figure: Average prediction entropy on DiffAug samples vs diffusion time measured with Imagenet-Test. We observe that the models trained with DiffAug correctly yield predictions with higher entropies for images containing imperceptible details (i.e. larger t). Surprisingly, the classifiers trained without DiffAug do not also assign uniform label distribution for DiffAug samples at $t = 999$, which have no class-related information by construction.

Certified Adversarial Accuracy with DDS and PAGs

	Certified Accuracy (%) at l_2 radius. 0.5	1.0	1.5	2.0	2.5	3.0
ViT	36.30	25.50	16.72	14.10	10.70	8.10
ViT+DiffAug	**40.30**	**32.50**	**23.62**	**19.40**	**15.20**	**11.00**

Table: Certified Accuracy with DDS [7] for different l_2 perturbation radius. As is standard in the literature, we consider $\sigma_t \in \{0.25, 0.5, 1.0\}$ and select the best σ_t for each l_2 radius.

Figure: Perceptually Aligned Gradients (PAGs) with ViT+DiffAug.

Theoretical Investigation of PAGs

To understand the perceptual alignment, we analyse the derivative of classifier log-likelihood and trace improvements to backpropagation through the score network:

$$\frac{d \log p_\phi(y|\hat{\mathbf{x}}_t)}{d\mathbf{x}} = \frac{d \log p_\phi(y|\hat{\mathbf{x}}_t)}{d\hat{\mathbf{x}}_t} \frac{d\hat{\mathbf{x}}_t}{d\mathbf{x}}$$

Theorem 1. *Consider a forward-diffusion SDE defined as* $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\, dt + g(t)\, d\mathbf{w}$ *such that* $p_t(\mathbf{x}|\mathbf{x}_0) = \mathcal{N}(\mathbf{x} \mid \mathbf{m}_t,\ \sigma^2(t)I)$ *where* $\mathbf{m}_t = \mu(\mathbf{x}_0, t)$. *If* $\mathbf{x} \sim p_t(\mathbf{x})$ *and* $\hat{\mathbf{x}}_t = \mathbf{x} + \sigma^2(t) s_\theta(\mathbf{x}, t)$, *for optimal parameters* θ, *the derivative of* $\hat{\mathbf{x}}_t$ *w.r.t.* $\mathbf{x}$ *is proportional to the covariance matrix of the conditional distribution* $p(\mathbf{m}_t|\mathbf{x})$.

$$\frac{\partial \hat{\mathbf{x}}_t}{\partial \mathbf{x}} = J = \frac{1}{\sigma^2(t)} Cov[\mathbf{m}_t|\mathbf{x}]$$

- A transformation by the covariance matrix stretches a vector along the principal components of the conditional distribution $p(\mathbf{m}_t|\mathbf{x})$.
- Since the conditional distribution $p(\mathbf{m}_t|\mathbf{x})$ corresponds to the distribution of *candidate* denoised images, the principal directions of variation are also perceptually aligned! This explains perceptually aligned gradients.

Improving Classifier-Guidance with DiffAug

(a) Guidance classifiers are trained on noisy forward-diffused examples to produce classifier-gradients for conditional generation with unconditional diffusion models.
(b) Observing perceptually aligned gradients, we use denoised examples to improve gradient quality in classifier-guided generation.
(c) We introduce **Denoise-Augmented Classifiers** (DA-Classifiers) wherein we train guidance classifiers using both noisy image and its corresponding denoised image as simultaneous inputs. Noisy Classifiers are trained with noisy inputs only.
(d) Noisy-classifiers achieve lower accuracy due to underfitting and the additional denoised input helps improve DA-Classifier. Previous works interpret noisy examples to lie in the ambient space while denoised examples lie in the data manifold.
(e) We observe improvements in (i) generalization, (ii) perceptual gradient alignment and (iii) image generation performance.

Method	CIFAR10 FID↓	IS ↑	$\bar{D}$ ↑	$\bar{C}$ ↑	Acc ↑	Imagenet FID ↓	sFID ↓	IS ↑	P ↑	R ↑	Acc-1 ↑
Noisy Classifier	2.81	9.59	0.78	0.71	54.79	5.44	**5.32**	194.48	0.81	0.49	33.79
DA-Classifier	**2.34**	**9.88**	**0.92**	**0.77**	**57.16**	**5.24**	5.37	**201.72**	0.81	0.49	**36.11**

Qualitative Evaluation

(a) PAG: Noisy-classifer vs DA-Classifier

(b) Generated Samples: Noisy-Classifier vs DA-Classifier