TADA: Targeted Image Augmentation via Diffusion Models

Dang Nguyen[*1] Jiping Li[*1] Jinghao Zheng[*2] Baharan Mirzasoleiman[1]

[1]University of California, Los Angeles [2]EPFL

ICLR 2026

[*]Equal contribution

Code: `github.com/BigML-CS-UCLA/TADA`

Problem: Synthetic Data Augmentation Is Expensive

Diffusion-based augmentation has become the go-to tool for improving image classification accuracy beyond standard augmentations.

But current methods:

- Augment the *entire* training set
- Require **10×–30×** data inflation to yield gains

Prior works

Method	Data ratio
Azizi et al., 2023	10×
DreamDA (Fu et al., 2024)	30×
TADA (ours)	**0.3–0.6×**

Key Question

Does augmenting the *full* dataset yield optimal performance?

Can we augment **only part** of the data and *do better*?

Motivation: Not All Examples Are Equally Hard to Learn

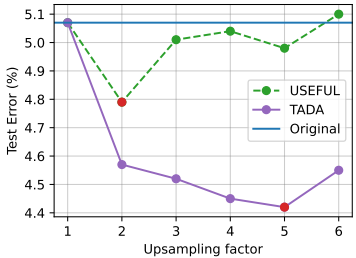
Insight from optimization (Nguyen et al., NeurIPS 2024):
SAM learns features at a *more uniform speed* than SGD, boosting generalisation by speeding up **slow-learnable** features.

- **Fast-learnable** examples: learned early (clear, large features)
- **Slow-learnable** examples: learned late (subtle, discriminative features)

Hypothesis: Synthetically augmenting *only* slow-learnable examples should replicate SAM's benefit — without a better optimizer.

Why not just upsample?
Upsampling amplifies *noise* by factor $k \Rightarrow$ hurts performance for $k > 2$.

Examples of slow- and fast-learnable images and faithful synthetic images. Features are preserved; noise is replaced.



USEFUL (upsampling) peaks at $k=2$; TADA (generation) improves up to $k=5$.

TADA (TArgeted Diffusion Augmentation): Three Steps

Step 1 — Identify Slow-Learnable Examples

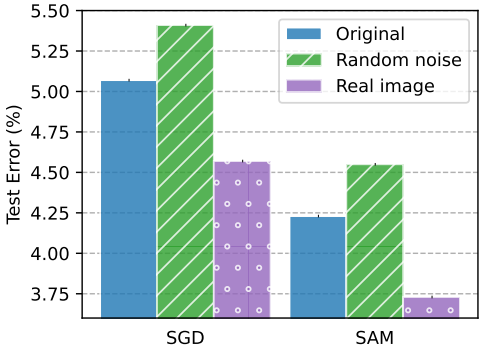
Cluster model outputs after a few epochs; select cluster with *higher average loss*.

Step 2 — Generate Faithful Synthetic Images

Add noise to real image, then denoise with a text prompt. Preserves semantic features; varies noise.

Step 3 — Replace & Retrain

Replace ($k = 2$–5) upsampled copies with synthetic images. Augments only **30–60%** of the dataset.



Real-image init vs. random-noise init denoising.

Theory: Why Generation Beats Upsampling

Analysis on a two-layer CNN — four key results:

Thm 1 — SAM Suppresses Noise

SAM's look-ahead reduces noise-direction alignment vs. SGD (**NoiseAlign** metric).

Thm 2 — Generation vs. Upsampling

Upsampling amplifies noise by factor k on repeated examples. Faithful generation keeps noise *independent* $\Rightarrow$ less overfitting.

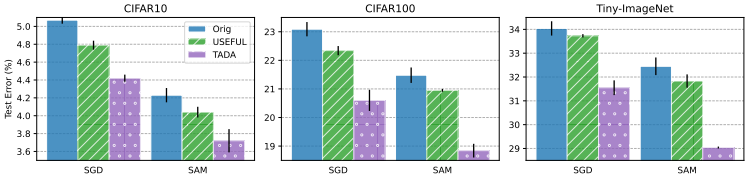
Thm 3 — Variance of mini-batch gradient

Upsampling inflates mini-batch gradient variance. Generation avoids this inflation entirely.

Cor 4 — Faster Convergence of Generation

Convergence of mini-batch SGD on synthetically augmented data is faster than on upsampled data.

Results: Consistent Gains Across Datasets

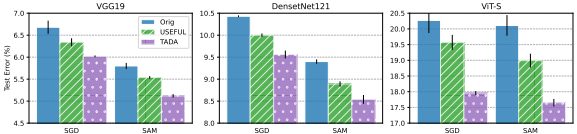


Test accuracy across datasets with ResNet18.

- Beats Original & USEFUL on **all datasets**
- **SGD + TADA > SAM** on CIFAR-100 & TinyImageNet
- SOTA on CIFAR-100 & TinyImageNet
- Up to **2.8%** gain on TinyImageNet
- Augments only **30–60%** of data

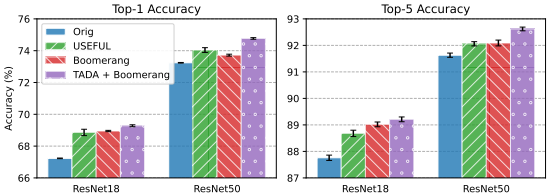
Results: Diverse Architectures & ImageNet Scale

Multiple architectures on CIFAR-10:



VGG19, DenseNet121, ViT-S — TADA consistently wins.

ImageNet (ResNet18 & ResNet50):



Top-1/Top-5 accuracy. Outperforms Boomerang while augmenting only **65%** of data vs. 100%.

Take-away

Augmenting the full dataset is suboptimal.
Targeted augmentation of slow-learnable examples with faithful synthetic images is both *more accurate* and *more efficient*.

TADA in a nutshell:

1. Identify slow-learnable examples via clustering
2. Generate faithful images with real-image guided diffusion
3. Replace ($k = 2$–5) upsampled copies with synthetic images

Results:

- Up to **2.8%** gain across CNNs & Transformers
- SGD + TADA > SAM on CIFAR-100 & TinyImageNet — SOTA
- Scales to ImageNet; works for transfer learning and object detection
- **0.3–0.6**× generation cost

Code: `github.com/BigML-CS-UCLA/TADA`