Self-Improving Diffusion Models with Synthetic Data

Sina Alemohammad Ahmed Imtiaz Humayun Shruti Agarwal
John Collomosse Richard Baraniuk

Rice University, Adobe Research

ICLR 2025
Presenter: Haotian Liu

August 23, 2025

Outline

Outline

Background

Task

- Using synthetic data to improve the performance of generative models, particularly diffusion models.

Limitations

- Over many generations of training, the quality and/or diversity of synthetic data will decrease, resulting in **Model Autophagy Disorder (MAD)** and **Model Collapse**.
- MADness arises because synthetic data, regardless of how accurately it is modeled and generated, is still an approximation of samples from the real data distribution.
- An autophagous loop causes any approximation errors to be compounded, ultimately resulting in performance deterioration and bias amplification.

Model Authophagy Disorder

$\mathcal{A}(\cdot)$ is an algorithm that, given a training dataset $\mathcal{D}$ as input, constructs a generative model with distribution $\mathcal{G}$, i.e., $\mathcal{G} = \mathcal{A}(\mathcal{D})$. Consider a sequence of generative models $\mathcal{G}^t = \mathcal{A}(\mathcal{D}^t)$ for $t \in \mathbb{N}$.

Let $\mathrm{dist}(\cdot, \cdot)$ denote a distance metric on distributions. A MAD generative process is a sequence of distributions $\left(\mathcal{G}^t\right)_{t \in \mathbb{N}}$ such that $\mathbb{E}\left[\mathrm{dist}\left(\mathcal{G}^t, p_{\mathrm{r}}\right)\right]$ increases with t. There are some main loop types:

- **Case 1: Fully Synthetic Loop.** Training data is purely synthetic: $\mathcal{D}_t = \mathcal{D}_s^{t-1}$
- **Case 2: Synthetic Augmentation.** Training data mixes a fixed real dataset $\mathcal{D}_r$ with synthetic data: $\mathcal{D}_t = \mathcal{D}_r \cup \mathcal{D}_s^{t-1}$

In particular, for the fully synthetic loop, it has been shown theoretically and experimentally that $\mathbb{E}\left[\mathrm{dist}\left(\mathcal{G}^\infty, p_{\mathrm{r}}\right)\right] \to \infty$.

Model Authophagy Disorder

Mitigating MAD

Goal: Ensure performance does not diverge.

$$\mathbb{E}\left[\operatorname{dist}\left(\mathcal{G}^{\infty}, p_{\mathrm{r}}\right)\right] \leq C$$

However, performance is still worse than the initial model:

$$\mathbb{E}\left[\operatorname{dist}\left(\mathcal{G}^{\infty}, p_{\mathrm{r}}\right)\right] > \mathbb{E}\left[\operatorname{dist}\left(\mathcal{G}^{1}, p_{\mathrm{r}}\right)\right]$$

Achieved by: Synthetic augmentation, accumulating past data.

Preventing MAD

Goal: Maintain or improve initial performance.

$$\mathbb{E}\left[\operatorname{dist}\left(\mathcal{G}^{\infty}, p_{\mathrm{r}}\right)\right] \leq \mathbb{E}\left[\operatorname{dist}\left(\mathcal{G}^{1}, p_{\mathrm{r}}\right)\right]$$

Existing approaches that prevent MAD are not **closed-loop**. They rely on **new external information** at each step, such as a data verifier or filter, external guidance during generation, or a fresh stream of real data.

Background

Open Question

- How can we best exploit synthetic data in generative model training to improve real data modeling and synthesis?
- How can we exploit synthetic data in generative model training in a way that does not lead to MADness in the future?

Solution

- Develop Self-IMproving diffusion models with Synthetic data (SIMS), a new learning framework for generative models that addresses both of the above issues simultaneously.

Diffusion Models Overview

Forward Process:

Start with a data instance $x_0 \sim p(x)$ and gradually add noise.
The distribution of the noisy sample x_t at time t given x_0 is a Gaussian:

$$q_t(x_t|x_0) = \mathcal{N}(x_t \mid \mu = a_t x_0, \Sigma = \sigma_t^2 I) \tag{1}$$

where a_t and σ_t are predefined scaling and noise schedules.
This process can also be described by a Stochastic Differential Equation (SDE):

$$dx = f(x, t)\, dt + g(t)\, dw \tag{2}$$

where w is the standard Wiener process (i.e., Brownian motion).

Note: For more details, see this link.

Diffusion Models Overview

Reverse Process: Generating Data

To generate data, we need to solve the reverse-time SDE, which starts from noise x_T and evolves towards a clean sample x_0. The reverse SDE is given by:

$$dx = \left[f(x,t) - g^2(t)\nabla_{x_t} \log q_t(x_t)\right] dt + g(t)\, d\bar{w} \tag{3}$$

Train a neural network $s_\theta(x_t, t)$ to approximate the unknown score function:

$$s_\theta(x_t, t) \approx \nabla_{x_t} \log q_t(x_t)$$

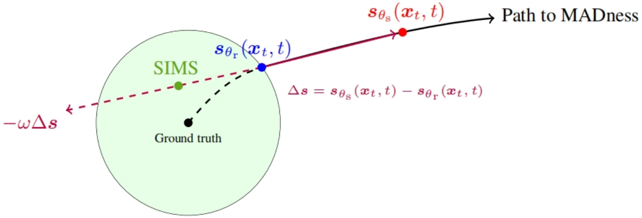
The model is trained by minimizing the following score-matching objective:

$$\min_\theta \frac{1}{|D|} \sum_{x_0 \in D} \mathbb{E}_{t, x_t \sim q_t(x_t|x_0)} \left[\lambda(t) \| s_\theta(x_t, t) - \nabla_{x_t} \log q_t(x_t) \|^2\right] \tag{4}$$

where D is the training set and $\lambda(t)$ is a weighting function.

Outline

Method



- Training a first-generation base diffusion model on exclusively real data results in a score function in the vicinity of the ground truth.
- Naïvely fine-tuning on synthetic data creates an auxiliary model (s_{θ_s}) that drifts **further away** from the Ground Truth, following the trajectory towards MADness.
- Instead of accepting degradation, SIMS uses the drift vector ($s_{\theta_s} - s_{\theta_r}$) to **linearly extrapolate backwards**. This pushes the final model into the inaccessible region, creating samples closer to the truth than the original base model.

Method

Algorithm 1 SIMS Procedure

Input: Training dataset $\mathcal{D}$
Hyperparameters: Synthetic dataset size n_s, guidance strength ω, training budget $\mathcal{B}$

1: **Train base diffusion model**: Use dataset $\mathcal{D}$ to train the diffusion model using standard training, resulting in the score function $\boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)$.
2: **Generate auxiliary synthetic data**: Create an internal synthetic dataset $\mathcal{S}$ by generating $n_\mathrm{s} = |\mathcal{S}|$ samples from the base diffusion model.
3: **Train auxiliary diffusion model**: Fine-tune the base model using only $\mathcal{S}$ within the training budget $\mathcal{B}$ to obtain $\boldsymbol{s}_{\theta_\mathrm{s}}(\boldsymbol{x}_t, t)$. Discard $\mathcal{S}$.
4: **Extrapolate the score function**: Use $\boldsymbol{s}_{\theta_\mathrm{s}}(\boldsymbol{x}_t, t)$ to extrapolate backwards from $\boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)$ to the SIMS score function

$$\boldsymbol{s}_{\theta}(\boldsymbol{x}_t, t) = \boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t) - \omega(\boldsymbol{s}_{\theta_\mathrm{s}}(\boldsymbol{x}_t, t) - \boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)) = (1+\omega)\boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t) - \omega \boldsymbol{s}_{\theta_\mathrm{s}}(\boldsymbol{x}_t, t).$$

Synthesize: Generate synthetic data from the model using the SIMS score function $\boldsymbol{s}_{\theta}(\boldsymbol{x}_t, t)$.

- **Synthetic Dataset Size (n_s)**: Too large implies No guidance. Too small implies a Poor estimate of drift, which means Ineffective guidance. Match the real dataset size.
- **Auxiliary Training Budget ($\mathcal{B}$)**: The goal is a score function that is "not too different, not too similar". Must find the **optimal stopping point**.
- **Inference Computational Cost**: SIMS requires **twice the function evaluations** at inference time. Apply guidance only within a limited time interval.

Outline

Main Experiment

- **Dataset**: 32$\times$32 resolution CIFAR-10 (50k images); 64 $\times$ 64 resolution FFHQ-64 (70k images), 64 $\times$ 64 resolution ImageNet-64, and 512 $\times$ 512 resolution ImageNet-512 (1.2M images).
- **Base Model**: For CIFAR-10 and FFHQ-64, use the unconditional Variance Preserving (VP) variant of the EDM diffusion model as the base model for SIMS. For ImageNet-64 and ImageNet-512, use the conditional EDM2-S model.
- **Baseline Comparison**: Standard diffusion-based image generation baselines, including ADM, RIN, EDM2-{S, M, L, XL}, DDPM, EDM-VP, NCSN++. Generative adversarial networks, including StyleGAN-XL and StyleGAN-2-ADA. Discriminator guided models EDM-G++ and LSGM-G++. Autoguidance guided models EDM2-{S,XL}.

Qualitative Results

CIFAR-10 32×32 **(Unconditional)**

Model	FID ↓	NFE ↓	Mparams
DDPM (Ho et al., 2020)	3.17	1000	-
StyleGAN2-ADA (Karras et al., 2020)	2.92	1	-
LSGM (Vahdat et al., 2021)	2.10	138	-
NCSN++ (Song et al., 2021)	2.20	2000	-
GDD Distill. (Zheng and Yang, 2024)	1.66	1	-
GDD-I Distill. (Zheng and Yang, 2024)	1.54	1	-
EDM-VP (Karras et al., 2022)	1.97	35	280
EDM-G++ (Kim et al., 2023)	1.77	35	-
LSGM-G++ (Kim et al., 2023)	1.94	138	-
EDM-VP + SIMS (Ours)	1.41	70	560
EDM-VP + SIMS + ST (Ours)	**1.33**	70	560

ImageNet 64×64

Model	FID ↓	NFE ↓	Mparams
ADM (Dhariwal and Nichol, 2021)	2.07	250	-
StyleGAN-XL (Sauer et al., 2022)	1.51	1	-
RIN (Jabri et al., 2023)	1.23	1000	280
EDM2-S (Karras et al., 2024a)	1.58	63	280
EDM2-M	1.43	63	498
EDM2-L	1.33	63	777
EDM2-XL	1.33	63	1119
AutoGuidance-S (Karras et al., 2024b)	1.01	126	560
GDD-I Distill. (Zheng and Yang, 2024)	1.21	1	-
EDM2-S + SIMS (Ours)	**0.92**	126	560

FFHQ 64×64

Model	FID ↓	NFE ↓	Mparams
EDM-VE (Karras et al., 2022)	2.53	79	280
EDM-VP (Karras et al., 2022)	2.39	79	280
EDM-G++ (Kim et al., 2023)	1.98	71	-
GDD Distill. (Zheng and Yang, 2024)	1.08	1	-
GDD-I Distill. (Zheng and Yang, 2024)	**0.85**	1	-
EDM-VP + SIMS (Ours)	1.04	158	560
EDM-VP + SIMS + ST (Ours)	1.03	158	560

ImageNet 512×512

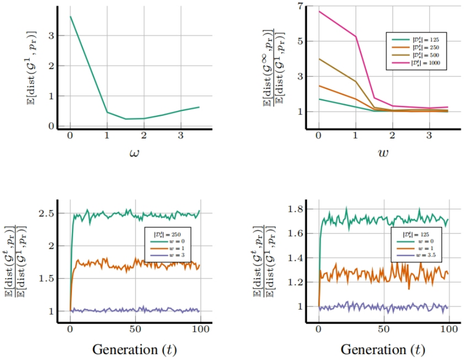
Model	FID ↓	NFE ↓	Mparams
ADM-G (Dhariwal and Nichol, 2021)	7.72	250	-
StyleGAN-XL (Sauer et al., 2022)	2.41	1	-
RIN (Jabri et al., 2023)	3.95	1000	320
EDM2-S (Karras et al., 2024a)	2.56	63	280
EDM2-M	2.25	63	498
EDM2-L	2.06	63	777
EDM2-XL	1.96	63	1119
EDM2-XXL	1.91	63	1523
AutoGuidance-S (Karras et al., 2024b)	1.34	126	560
AutoGuidance-XL (Karras et al., 2024b)	**1.25**	126	2236
EDM2-S + SIMS (Ours)	1.73	126	560

- Self-improvement with synthetic data is more effective than simply scaling up model parameters.
- Guiding away from the model's flawed distribution is a more powerful strategy than guiding towards a realism score.

MAD prevention using SIMS

- Learn a simple two-dimensional Gaussian distribution $p_{\mathrm{r}} = \mathcal{N}(\mu, \Sigma)$ with mean $\mu = [0, 0]^\top$ and covariance $\Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ using a DDPM diffusion model.
- Sample a real dataset $\mathcal{D}_{\mathrm{r}}$ of size $|\mathcal{D}_{\mathrm{r}}| = 1000$ from $\mathcal{N}(\mu, \Sigma)$ and train the base model $\mathcal{G}^1 = \mathcal{A}(\mathcal{D}_{\mathrm{r}})$.
- Then form a synthetic augmentation loop, $\mathcal{G}^t = \mathcal{A}(\mathcal{D}_{\mathrm{r}} \cup \mathcal{D}_{\mathrm{s}}^{t-1})$.
- Calculate the Wasserstein distance $\mathrm{dist}(\cdot, \cdot)$ between the synthetic and real data distributions $\mathbb{E}[\mathrm{dist}(\mathcal{G}^t, p_{\mathrm{r}})]$.
- Two different training approaches: **Standard training** and **SIMS**.

Results

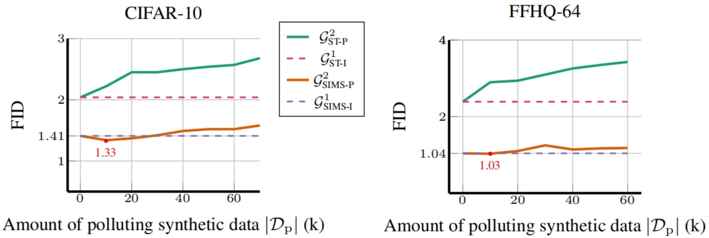


- **Standard training** on polluted data consistently degrades model performance.
- **SIMS** successfully counters this degradation, and with optimal guidance, can **completely prevent MADness**.
- The ability to prevent MAD depends on a **threshold**; too much synthetic data pollution limits SIMS to mitigation only.

Realistic Data in A Synthetic Augmentation Loop

- Use the EDM-VP model trained on CIFRA-10 and FFHQ-64.
- Standard training with purely real data, $\mathcal{G}^1_{\text{ST-I}}$.
- Ideal SIMS training with purely real data, $\mathcal{G}^1_{\text{SIMS-I}}$.
- Standard training with polluted real data, $\mathcal{G}^2_{\text{ST-P}}$.
- SIMS training with polluted real data, $\mathcal{G}^2_{\text{SIMS-P}}$.

Results

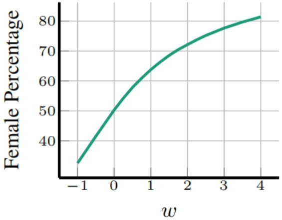


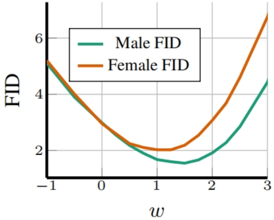
- **Standard training is fragile:** Its performance degrades severely with data pollution.
- **SIMS is robust:** Its performance is remarkably insensitive to the same data pollution.
- **SIMS exploits pollution:** It can leverage polluted data to **surpass the original, clean-data model**.

Distribution Shifts with SIMS

- **Goal**: Can SIMS not only improve sample quality but also **steer the output distribution** towards a desirable target, different from the training data?
- **Test Case:** Shift the gender representation in FFHQ-64 generations from the base model's **50% female / 50% male** to a target of **70% female / 30% male**.
- **"Negative Guidance" Method**: create a synthetic dataset S that is intentionally biased to be **30% female and 70% male**.
- **Execute SIMS:** By guiding away from the male-dominant auxiliary model, the final output is pushed towards the female-dominant target.

Results





- **Distribution is Controllable:** The percentage of female faces smoothly and predictably increases with guidance strength ω, reaching the 70% target.
- **Quality is Simultaneously Improved:** The FID scores for both male and female images improve, reaching optimal quality at specific ω values.
- **A Minor Trade-off Exists:** The optimal ω for distribution accuracy may not perfectly align with the optimal ω for image quality.

Outline

Conclusion

- **Conclusion:** This paper introduces a new algorithm that uses synthetic data for *negative guidance* rather than data aggregation, steering models away from their flaws. It achieves new SOTA results on major benchmarks and offers a powerful tool to mitigate bias.
- **Open Research Questions**
  1. Why does SIMS not only tolerate but *capitalize on* data pollution to improve performance?
  2. Can this negative guidance principle be adapted to other architectures like GANs or VAEs?
  3. Could two *different* high-performing models guide each other, creating a general defense against diverse synthetic data?
  4. Can SIMS be extended to align with human preferences by using user feedback to shape the "negative" distribution?

Thank you!