SELF-IMPROVING DIFFUSION MODELS WITH SYNTHETIC DATA

Sina Alemohammad[*†], **Ahmed Imtiaz Humayun**[*], **Shruti Agarwal**[†], **John Collomosse**[†], **Richard Baraniuk**[*]
[*]Rice University, [†]Adobe Research

ABSTRACT

The artificial intelligence (AI) world is running out of real data for training increasingly large generative models, resulting in accelerating pressure to train on synthetic data. Unfortunately, training new generative models with synthetic data from current or past generation models creates an *autophagous* (self-consuming) *loop* that degrades the quality and/or diversity of the synthetic data in what has been termed *model autophagy disorder* (MAD) and *model collapse*. Current thinking around model autophagy recommends that synthetic data is to be avoided for model training lest the system deteriorate into MADness. In this paper, we take a different tack that treats synthetic data differently from real data. Self-IMproving diffusion models with Synthetic data (SIMS) is a new training concept for diffusion models that uses self-synthesized data to provide *negative guidance* during the generation process to steer a model's generative process away from the non-ideal synthetic data manifold and towards the real data distribution. We demonstrate that SIMS is capable of *self-improvement*; it establishes new records based on the Fréchet inception distance (FID) metric for CIFAR-10 and ImageNet-64 generation and achieves competitive results on FFHQ-64 and ImageNet-512. Moreover, SIMS is, to the best of our knowledge, the first *prophylactic* generative AI algorithm that can be iteratively trained on self-generated synthetic data without going MAD. As a bonus, SIMS can adjust a diffusion model's synthetic data distribution to match any desired in-domain target distribution to help mitigate biases and ensure fairness.

Figure 1: **Self-IMproving diffusion models with Synthetic data (SIMS) simultaneously improves diffusion modeling and synthesis performance while acting as a prophylactic against Model Autophagy Disorder (MAD)**. First row: Samples from a base diffusion model (EDM2-S (Kynkäänniemi et al., 2024)) trained on 1.28M real images from the ImageNet-512 dataset Karras et al. (2024a) (Fréchet inception distance, FID = 2.56). Second row: Samples from the base model after fine-tuning with 1.5M images synthesized from the base model, which degrades synthesis performance and pushes the model towards MADness (aka model collapse) (Alemohammad et al., 2023; 2024; Shumailov et al., 2024) (FID = 6.07). Third row: Samples from the base model after applying SIMS using the same self-generated synthetic data as in the second row (FID = 1.73).

1 INTRODUCTION

Thanks to the ongoing rapid advances in the field of generative artificial intelligence (AI), we are witnessing a proliferation of synthetic data of various modalities that have been rapidly integrated into popular online platforms. The voracious appetite of generative models for training data (Yahoo-Finance, 2024; The Economist, 2023a;b; Villalobos et al., 2022) has caused practitioners to train new models either partially or completely using synthetic data from previous generations of models. Synthetic training data is actually hard to avoid, because many of today's popular training datasets have been inadvertently polluted with synthetic data (Alemohammad et al., 2023; 2024).

Unfortunately, there are hidden costs to synthetic data training. Training new generative models with synthetic data from current or past generation models creates an *autophagous* (self-consuming) *loop* (Alemohammad et al., 2023; 2024) that can have a detrimental effect on performance. In the limit over many generations of training, the *quality and/or diversity of the synthetic data will decrease*, in what has been termed Model Autophagy Disorder (MAD) (Alemohammad et al., 2023; 2024) and Model Collapse (Shumailov et al., 2024). MAD generative models also have major *fairness* issues, as they produce *increasingly biased samples* that lead to inaccurate representations across the attributes present in real data (e.g., related to demographic factors such as gender and race) (Wyllie et al., 2024).

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.[1] An autophagous loop causes any approximation errors to be compounded, ultimately resulting in performance deterioration and bias amplification.

Safely advancing the performance of generative AI systems in the synthetic data era requires that we make progress on both of the following open questions:

Q1. How can we best exploit synthetic data in generative model training to improve real data modeling and synthesis?

Q2. How can we exploit synthetic data in generative model training in a way that does not lead to MADness in the future?

In this paper, we develop *Self-IMproving diffusion models with Synthetic data* (SIMS), a new learning framework for generative models that addresses both of the above issues simultaneously. Our key insight is that, to most effectively exploit synthetic data in training a generative model, we need to change how we employ synthetic data. Instead of naïvely training a model on synthetic data as though it were real, SIMS guides the model towards better performance but away from the patterns that arise from synthetic data training.

We focus here on SIMS for *diffusion models* in the context of image generation, because their robust guidance capabilities enable us to efficiently guide them away from their own generated synthetic data. In particular, we use a base model's own synthetic data to obtain a *synthetic score function* associated with the synthetic data manifold and use it to provide *negative guidance* during the generation process. By doing so, we steer the model's generative process away from the non-ideal synthetic data manifold and towards the real data distribution.

Figure 2 depicts how SIMS models and synthesizes more closely to the ground truth, real data distribution by *reversing the trajectory towards MADness*. The green circle signifies the region in the function space of score functions that is inaccessible to a learning algorithm due to factors such as a limited amount of real data or sampling noise. As a result, training a first-generation base diffusion model on exclusively real data results in a score function $\boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)$ (parameterized by a learnable neural network with parameters θ_r) in the vicinity of the ground truth. Now, consider naïvely training a second-generation (auxilliary) model by fine-tuning the base model with synthetic data from the first-generation model. This corresponds to the synthetic augmentation loop in (Alemohammad et al., 2023; 2024). The resulting score function $\boldsymbol{s}_{\theta_\mathrm{s}}(\boldsymbol{x}_t, t)$ will be further away from the ground truth and on the path towards MADness. Rather than tolerate this degraded second-generation model, we can use $\boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)$ and $\boldsymbol{s}_{\theta_\mathrm{s}}(\boldsymbol{x}_t, t)$ to (linearly) extrapolate back into the inaccessible region. Data generated using the resulting score function (denoted SIMS in the figure) promise to be closer to the

Corresponding author: sa86@rice.edu

[1] In this paper, by *real data* we mean direct samples from a target distribution. For example, in the context of natural images, real data would be digital photographs taken by a camera in a physical space.

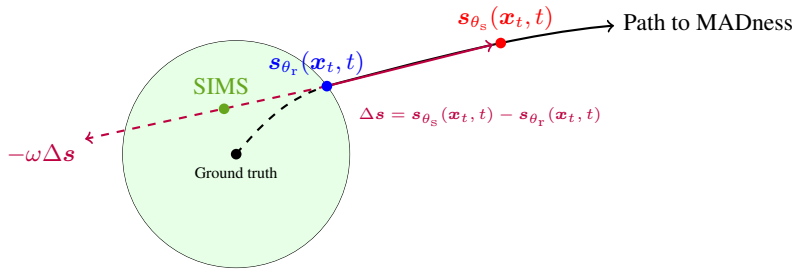


Figure 2: **SIMS simultaneously self-improves diffusion model modeling and synthesis performance while acting as a prophylactic against MAD.** SIMS improves the score function $s_{\theta_r}(x_t, t)$ for a base diffusion model trained on real data by training an auxiliary model on the same real data plus synthetic data from the base model. The score function $s_{\theta_s}(x_t, t)$ of the auxiliary model can be combined with that of the base model to extrapolate a new score function (denoted SIMS) that is closer to the real data distribution.

real data distribution than those from the base model. Moreover, by explicitly reversing the MADness trajectory, diffusion models learned using SIMS promise to be less prone to MADness when the above process is repeated.

To summarize, given a training dataset, SIMS performs the following four steps to obtain a self-improved diffusion model using self-generated synthetic data:

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 $s_{\theta_r}(x_t, t)$.
2: **Generate auxiliary synthetic data**: Create an internal synthetic dataset $\mathcal{S}$ by generating $n_s = |\mathcal{S}|$ 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 $s_{\theta_s}(x_t, t)$. Discard $\mathcal{S}$.
4: **Extrapolate the score function**: Use $s_{\theta_s}(x_t, t)$ to extrapolate backwards from $s_{\theta_r}(x_t, t)$ to the SIMS score function

$$s_\theta(x_t, t) = s_{\theta_r}(x_t, t) - \omega(s_{\theta_s}(x_t, t) - s_{\theta_r}(x_t, t)) = (1+\omega)s_{\theta_r}(x_t, t) - \omega s_{\theta_s}(x_t, t).$$

Synthesize: Generate synthetic data from the model using the SIMS score function $s_\theta(x_t, t)$.

We tally our contributions as follows:

C1. Self-improvement: We demonstrate that SIMS makes significant progress on Q1 above by using self-generated synthetic data to significantly enhance the generation quality of image diffusion models. SIMS establishes new records based on the Fréchet inception distance (FID) metric (Heusel et al., 2017) for CIFAR-10 and ImageNet-64 generation and achieves competitive results on FFHQ-64 and ImageNet-512 (see Table 1 and Figure 4). Moreover, given a diffusion model trained on a set of real data and a set of data synthesized from that model, we show that combining the real and synthetic data via SIMS results in higher performance than either the original model or a model trained on the aggregate of the real and synthetic data (see Figures 1 and 6).

C2. MAD-prophylactic: We demonstrate that SIMS makes significant progress on Q2 above. It is, to the best of our knowledge, *the first generative AI model that can be iteratively trained on self-generated, synthetic data without going MAD.* We iterate the process in Algorithm 1 for 100 generations to show that there exist guidance parameter ω settings such that no MAD degradation occurs (see Figure 5).

C3. Distribution controllability: We show that SIMS can adjust a diffusion model's synthetic data distribution to match any desired in-domain target distribution. This can help mitigate biases and ensure model fairness, all while improving the quality of the generated outputs (see Figure 7).

Our findings clearly demonstrate that synthetic data can actually be both useful and safe for learning diffusion models and counters recent recommendations (Alemohammad et al., 2023; 2024; Shumailov et al., 2024) that synthetic data is to be avoided in learning. The difference in conclusions is due to SIMS' unique approach: while training directly on (real data aggregated with) synthetic data causes a model to drift away from the true data distribution, SIMS instead uses the synthetic data to explicitly avoid the synthetic data manifold and extrapolate closer to the true data distribution.

This paper is organized as follows. Section 2 overviews diffusion generative models, self-consuming loops, and past work attempting to slow or arrest MADness. Section 3 presents the details of the SIMS procedure that we outlined in Algorithm 1. Section 4 exhibits the results of numerous computational experiments that demonstrated convincingly that SIMS both improves model performance and either mitigates or completely prevents MADness. We close with a discussion, recommendations, and direcions for future research in Section 5.

2 Background

Diffusion models. Let p denote the distribution we seek to model. Diffusion models gradually diffuse the training data over time $t \in [0, T]$ and sample from p by inversely modeling the forward diffusion process (Ho et al., 2020; Song and Ermon, 2019). Typically, this diffusion process involves transforming instances drawn from p into noisy versions with scale schedule a_t and noise schedule σ_t at time t. Hence, the conditional distribution of the noisy sample $\boldsymbol{x}_t$ at time t can be formalized as

$$q_t(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t \mid \boldsymbol{\mu} = a_t\boldsymbol{x}_0, \boldsymbol{\Sigma} = \sigma_t\boldsymbol{I}), \tag{1}$$

where $\boldsymbol{x}_0$ is the data instance drawn from p. The diffusion process can be formalized using a stochastic differential equation (SDE) (Song and Ermon, 2019)

$$d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}, \tag{2}$$

where $\boldsymbol{w}$ is the standard Wiener process. Different choices for $f(\boldsymbol{x}, t)$ and $g(t)$ result in different scaling a_t and noise σ_t schedules in (1). We refer the reader to (Karras et al., 2024a) for more details on different SDE formulations for diffusion models.

The solution to the SDE in (2) is another SDE described by (Anderson, 1982)

$$d\boldsymbol{x} = \Big[f(\boldsymbol{x}, t) - g^2(t)\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\Big]dt + g(t)d\bar{\boldsymbol{w}}, \tag{3}$$

where $d\bar{\boldsymbol{w}}$ is the standard Wiener process when time flows in the reverse direction, and q_t is the unconditional distribution in (1) obtained by the forward SDE through (2). The solution of the SDE in (3) starting from the samples of $\boldsymbol{x}_T \sim q_T$ results in samples $\boldsymbol{x} \sim q_0(\boldsymbol{x}_0)$ that enable data generation from p.

Since the score function $\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)$ is unknown, the objective is to train a neural network with parameters θ to approximate the score function $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \approx \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)$ through

$$\min_\theta \frac{1}{|\mathcal{D}|} \sum_{\boldsymbol{x}_0 \in \mathcal{D}} \mathbb{E}_{t\in[0,T], \boldsymbol{x}_t \sim q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)} \Big[\lambda(t) \|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\|^2\Big], \tag{4}$$

where $\mathcal{D}$ is the training set containing samples from p, and $\lambda(t)$ is a temporal weighting function. The SDE in (3) can be solved by replacing $\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)$ with $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ and performing numerical integration. For conditional generation, one can also impose a condition on the score function during training to obtain the conditional score.

Self-consuming generative models. Let $\mathcal{A}(\cdot)$ represent 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 $t \in \mathbb{N}$, where each model approximates some reference (typically real data) probability distribution p_{r}.

Definition 1. Self-consuming (autophagous) loop (Alemohammad et al., 2023; 2024): An autophagous loop is a sequence of distributions $(\mathcal{G}^t)_{t\in\mathbb{N}}$ where each generative model $\mathcal{G}^t$ is trained on data that includes samples from previous generation models $(\mathcal{G}^\tau)_{\tau=1}^{t-1}$.

Definition 2. Model Authophagy Disorder (MAD) (Alemohammad et al., 2023; 2024): Let $\text{dist}(\cdot, \cdot)$ denote a distance metric on distributions. A *MAD generative process* is a sequence of distributions $(\mathcal{G}^t)_{t\in\mathbb{N}}$ such that $\mathbb{E}[\text{dist}(\mathcal{G}^t, p_{\text{r}})]$ increases with t.

One can form a variety of self-consuming loops based on how $\mathcal{D}^t$, the training data at generation t, is constructed from real data $\mathcal{D}_{\mathrm{r}}^t$ drawn from p_{r} and synthetic data $\mathcal{D}_{\mathrm{s}}^t$ generated by the model $\mathcal{G}^t$. Let the first generation model be trained solely on real data, i.e, $\mathcal{G}^1 = \mathcal{A}(\mathcal{D}_{\mathrm{r}})$. For subsequent generation models $\mathcal{G}^t = \mathcal{A}(\mathcal{D}^t), t \geq 2$, the three main loop types proposed in (Alemohammad et al., 2023; 2024) are based on how $\mathcal{D}^t$ is constructed:

- **Fully synthetic loop**: Each model $\mathcal{G}^t$ for $t \geq 2$ trains exclusively on synthetic data sampled from models from the previous generation model, i.e., $\mathcal{D}^t = \mathcal{D}_{\mathrm{s}}^{t-1}$.
- **Synthetic augmentation loop**: Each model $\mathcal{G}^t$ for $t \geq 2$ trains on the dataset $\mathcal{D}^t = \mathcal{D}_{\mathrm{r}} \cup \mathcal{D}_{\mathrm{s}}^{t-1}$ comprising a fixed set of real data $\mathcal{D}_{\mathrm{r}}$ from p_{r} plus synthetic data $\mathcal{D}_{\mathrm{s}}^{t-1}$ from the previous generation model.
- **Fresh data loop**: Each model $\mathcal{G}^t$ for $t \geq 2$ trains on the dataset $\mathcal{D}^t = \mathcal{D}_{\mathrm{r}}^t \cup \mathcal{D}_{\mathrm{s}}^{t-1}$ comprising a fresh (new) set of real data $\mathcal{D}_{\mathrm{r}}^t$ drawn from p_{r} plus synthetic data $\mathcal{D}_{\mathrm{s}}^{t-1}$ from the previous generation model.

This paper focuses on the first two loop types above, which in general deteriorate into MADness of some kind. In particular, for the fully synthetic loop, it has been shown theoretically and experimentally that $\mathbb{E}[\mathrm{dist}(\mathcal{G}^\infty, p_{\mathrm{r}})] \to \infty$ (Alemohammad et al., 2023; 2024). In this scenario, often referred to as "model collapse" (Shumailov et al., 2024) in the literature, the sequence of models drifts away from the real data distribution until it no longer resembles it.

Mitigating MADness. Several groups have developed methods to mitigate MADness, which we define as ensuring that $\mathbb{E}[\mathrm{dist}(\mathcal{G}^\infty, p_{\mathrm{r}})] \leq C$ for some bounded C. In words, the performance of a mitigated-MAD family of models does not diverge into full MADness ($C \to \infty$) but plateaus at a level that does not exceed the performance of the first-generation model, i.e., $\mathbb{E}[\mathrm{dist}(\mathcal{G}^\infty, p_{\mathrm{r}})] > \mathbb{E}[\mathrm{dist}(\mathcal{G}^1, p_{\mathrm{r}})]$.

(Bertrand et al., 2023; Feng et al., 2024a) show that MADness can be mitigated in the synthetic augmentation loop. The continuous inclusion of real data in the training set prevents the model from drifting too far from the initial model. (Dohmatob et al., 2024a; Gerstgrasser et al., 2024) show that it is possible to mitigate MADness without incorporating real data in every generation, as long as the synthetic dataset size increases linearly across generations by accumulating synthetic data from all previous generations.

Preventing MADness. To more completely address the problem of performance degradation in self-consuming loops, one should aim to not just mitigate but *prevent MADness*, where the sequence of model generations at least maintains and ideally improves on the performance of the first-generation base model, i.e., $\mathbb{E}[\mathrm{dist}(\mathcal{G}^\infty, p_{\mathrm{r}})] \leq \mathbb{E}[\mathrm{dist}(\mathcal{G}^1, p_{\mathrm{r}})]$.

The above results involve a closed loop, where the only external information about the target distribution p_{r} is a fixed initial real dataset. Incorporating new external information in self-consuming loops — such as a verifier to oversee synthetic data selection Feng et al. (2024b); Setlur et al. (2024), external guidance during the generation process Gillman et al. (2024), or fresh real data (Alemohammad et al., 2023; 2024) — has been shown to prevent MADness.

Research on self-consuming loops has not yet identified an approach where the inclusion of synthetic data in a closed loop with no external knowledge not only mitigates MADness across generations but completely prevents it. In the next section, we introduce SIMS, and in Section 4.2, we show that using SIMS as the training algorithm $\mathcal{A}(\cdot)$ in the synthetic augmentation loop can fully prevent MADness.

3 SIMS: Self Improvement with Synthetic Data

In this section, we develop the *Self-IMproving diffusion models with Synthetic data* (SIMS) framework (recall Algorithm 1) for improving the performance of a diffusion model using its own synthetic data; we term this *self-improvement*. Note that while we explain SIMS in the context of unconditional diffusion models, our method extends to conditional diffusion models as well.

SIMS: Extrapolating to Self-Improvement. Let us unpack the SIMS steps outlined in Algorithm 1 in the introduction. Consider a *base diffusion model* characterized by the score function $\boldsymbol{s}_{\theta_{\mathrm{r}}}(\boldsymbol{x}_t, t)$

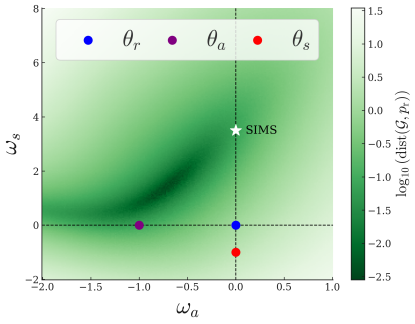


Figure 3: Distance of a diffusion model $\mathcal{G}$ from ground truth distributions p_r with score function $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$, where $\theta = (1 + \omega_a + \omega_s)\theta_r - \omega_a\theta_a - \omega_s\theta_s$.

that was trained on real data samples drawn from a target real data distribution p_r. At noise level t, the score function $\boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)$ outputs a vector $\boldsymbol{z}_r$ that points in the direction of increasing log probability density $\log p_\mathrm{r}$. By numerically solving the reverse SDE in (3) using the score function $\boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)$, we obtain samples that follow the synthetic data distribution p_s. The goal is to have $p_s \approx p_\mathrm{r}$. However, due to various factors — including, but not limited to, the limited size of the training dataset, inaccuracies in solving the reverse SDE, and implicit algorithmic biases — the synthetic data distribution p_s does not exactly match the target distribution p_r. This discrepancy results in a *model-induced distribution shift.*

To address this shift, we train a separate, *auxiliary diffusion model* using the same training hyperparameters used for the base model (i.e., for obtaining $\boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)$) using a temporary, internal synthetic dataset $\mathcal{S}$ containing samples drawn from p_s. This results in the score function $\boldsymbol{s}_{\theta_\mathrm{s}}(\boldsymbol{x}_t, t)$. Since $\boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)$ and $\boldsymbol{s}_{\theta_\mathrm{s}}(\boldsymbol{x}_t, t)$ are approximations of p_r and p_s, respectively, their difference serves as a useful surrogate for the model-induced distribution shift. By leveraging this difference, we can guide the model away from p_s during the generation process, thereby mollifying the impact of the shift. This guidance can be applied either to the model's parameters or directly in the functional space of the score functions.

In the function space, we guide the model away using $\boldsymbol{s}_{\theta_\mathrm{s}}(\boldsymbol{x}_t, t) - \boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)$ during the generation process, which the motivates the *SIMS score function* that we introduced in Algorithm 1 and reprise here:

$$\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t) - \omega(\boldsymbol{s}_{\theta_\mathrm{s}}(\boldsymbol{x}_t, t) - \boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)) = (1 + \omega)\boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t) - \omega\boldsymbol{s}_{\theta_\mathrm{s}}(\boldsymbol{x}_t, t), \tag{5}$$

where ω is the guidance strength. Alternatively, the SIMS score function $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ can also be derived by directly modifying the model's parameters, as follows:

$$\boldsymbol{s}_\theta(\boldsymbol{x}_t, t); \;\; \theta = \theta_r - \omega(\theta_s - \theta_r) = (1 + \omega)\theta_r - \omega\theta_s. \tag{6}$$

When parameters are close to each other, ensemble methods in the parameter space are approximately equivalent to those in the functional space. Specifically, if $\theta = \sum_{i=1}^n \omega_i\theta_i$, where $\sum_{i=1}^n \omega_i = 1$, (Biggs et al., 2024) showed that we can approximate the score function as:

$$\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \approx \sum_{i=1}^n \omega_i \boldsymbol{s}_{\theta_i}(\boldsymbol{x}_t, t) \tag{7}$$

provided that all parameters θ_i are within close proximity i.e, $\|\theta_i - \theta_j\|$ are small. The proof for this result is detailed in (Biggs et al., 2024), and we also include it in the Appendix **??** for the completeness of this paper.

Figure 3 illustrates how SIMS operates by exploring the parameter space to learn a two-dimensional Gaussian distribution $p_\mathrm{r} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu} = [0, 0]^\top$ and covariance $\boldsymbol{\Sigma} = [2, 1; 1, 2]$. A DDPM diffusion model (Ho et al., 2020; Álvaro Jiménez, 2023) is first trained on $|\mathcal{D}_\mathrm{r}| = 10^3$ samples to create a base model $\boldsymbol{s}_{\theta_\mathrm{r}}(\boldsymbol{x}_t, t)$ with parameters θ_r.

Using this base model, $|\mathcal{S}| = 10^5$ synthetic samples are generated and used for fine-tuning, producing an auxiliary score function $\boldsymbol{s}_{\theta_s}(\boldsymbol{x}_t, t)$. Separately, an additional $|\mathcal{D}_\mathrm{r}| = 10^5$ real samples are used to independently fine-tune the base model, resulting in an advanced score function $\boldsymbol{s}_{\theta_a}(\boldsymbol{x}_t, t)$, which better approximates p_r due to the larger real dataset.

A final score function is constructed as $\theta = (1 + \omega_a + \omega_s)\theta_r - \omega_a\theta_a - \omega_s\theta_s$. Figure 3 shows the Wasserstein distance to p_{r}, with dashed lines for $\omega_a = 0$ (self-improvement path) and $\omega_s = 0$ (base-to-advanced model path). Improvements in both directions reflect the benefits of synthetic data and larger real datasets, with the best model leveraging parameters from both advanced and auxiliary models.

Notably, both SIMS and the advanced model converge to the same local minimum, progressing toward improved models within this region. The guidance direction from θ_s along the SIMS path, originally derived from the base model, continues to hold for advanced models and intermediates. This indicates that the SIMS correction direction is robust across different model performance levels, offering a consistent and universal path for improvement.

For parameter space guidance to be effective (Equation 5), θ_s must remain close to θ_r within the same local minimum. This requires fine-tuning the base model with a large synthetic dataset ($n_s = |\mathcal{S}|$) and a small learning rate. Larger learning rates or smaller n_s risk moving the model out of the local minimum near θ_r, making guidance ineffective. In contrast, function space guidance is more robust to these hyperparameters, as parameter permutations can align models in different local minima without altering the score function Entezari et al. (2022). Ablation studies on learning rates and synthetic dataset sizes for SIMS are provided in Appendix **??**, and we adopt function space guidance in Algorithm 1 for its robustness. Our experiments suggest that matching the synthetic dataset size to the real dataset and using the same learning rate as the base model is effective for obtaining the auxiliary model for function space guidance.

The training budget $\mathcal{B}$ of the auxiliary model has a more profound effect on SIMS' performance. The goal is to obtain a score function $\boldsymbol{s}_{\theta_{\mathrm{s}}}(\boldsymbol{x}_t, t)$ that is neither too different from nor too similar to $\boldsymbol{s}_{\theta_{\mathrm{r}}}(\boldsymbol{x}_t, t)$. Therefore, we initialize $\boldsymbol{s}_{\theta_{\mathrm{s}}}(\boldsymbol{x}_t, t)$ with parameters θ_{r} and then fine-tune with a training budget $\mathcal{B}$ on the synthetic dataset $\mathcal{S}$. In this paper, we quantify the training budget by how many images are seen by the model. When $\mathcal{B} = 0$ at the start of training, we have $\boldsymbol{s}_{\theta_{\mathrm{s}}}(\boldsymbol{x}_t, t) = \boldsymbol{s}_{\theta_{\mathrm{r}}}(\boldsymbol{x}_t, t)$, and SIMS is equivalent to only using $\boldsymbol{s}_{\theta_{\mathrm{r}}}(\boldsymbol{x}_t, t)$ for data synthesis regardless of the value of ω. As we increase $\mathcal{B}$, we can expect the SIMS score function in (5) to approach the ground truth distribution and then depart as the auxiliary model becomes influenced less by the training data $\mathcal{D}$ and more by the synthetic data $\mathcal{S}$. Consequently, we stop the auxiliary model fine-tuning process at the optimizing $\mathcal{B}$.

Inference Computational Cost. While guidance in parameter space results in a single score function, SIMS using function space guidance requires twice the number of function evaluations as the base model at inference time because of the auxiliary model score function (recall (5)). However, the number of function evaluations can be reduced with minimal impact on performance by applying guidance from the auxiliary model within a limited interval, as proposed in (Kynkäänniemi et al., 2024), or by fine-tuning only a portion of the base model to obtain the auxiliary model. Appendix A provides ablation studies regarding reducing the number of function evaluations and the effect of the synthetic dataset size on fine-tuning.

Related Work. Augmenting the score function of a diffusion model with guidance from external models has been an active research direction in diffusion-based generative modeling. Dhariwal and Nichol (2021) introduced the notion of classifier guidance, which involves training a separate conditional classifier using denoised images from a base model and using the gradients of the classifier to steer the denoising trajectory at every step. Wallace et al. (2023) introduced a method to perform plug-and-play guidance with pre-trained classifiers. Ho and Salimans (2022) introduced classifier-free guidance, where a diffusion model is trained to learn both a conditional and unconditional score function. During denoising, the unconditional score function is used as negative guidance, which leads to impressive gains in generation fidelity. Kim et al. (2023) proposed discriminator guidance, where gradients of a discriminator are used to perform guidance. The discriminator is trained to classify between real images from the training dataset and synthetic samples generated by the target diffusion model. Discriminator guidance can be considered a proto *self-improvement* method since it employs synthetic data from the base model to increase the realism of the generated samples. Ahn et al. (2024) proposed a method to self-improve conditional or unconditional diffusion models without any training — by performing negative guidance using a clone of the base model with the attention weights replaced by an identity matrix, effectively resulting in a worse version of the base model. A similar strategy was applied in the concurrent work of Karras et al. (2024b), who suggest using a "bad" version of the base model for negative guidance. The authors suggest training a reduced-parameter

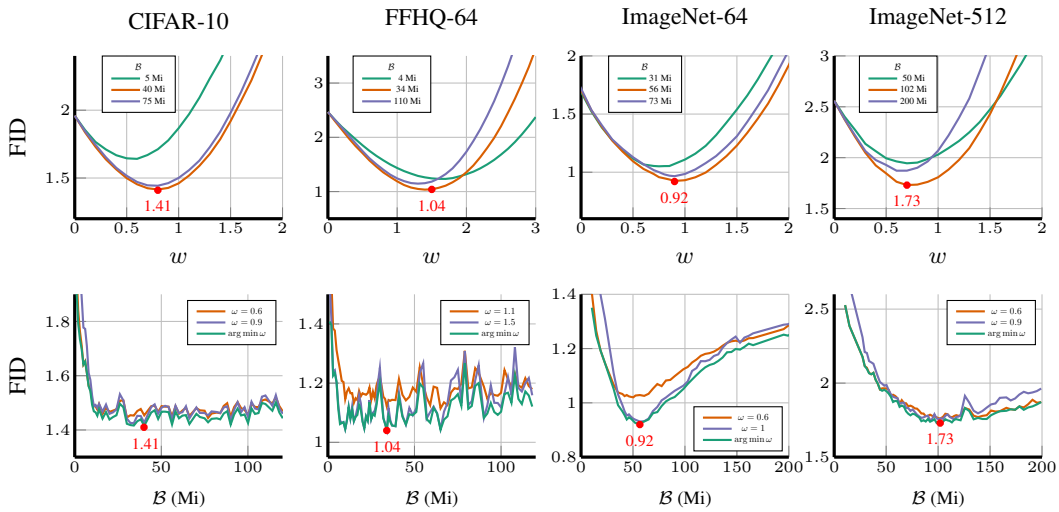


Figure 4: **SIMS consistently self-improves diffusion models.** Top row: FID between the SIMS model from Algorithm 1 and the real data distribution as a function of the guidance parameter ω at three different checkpoints of the training budget $\mathcal{B}$ as measured by the number of million-images-seen (Mi) during fine tuning of the auxiliary model. Bottom row: FID of the SIMS model as a function of training budget for three different values of the guidance parameter ω.

model for fewer training epochs to obtain a bad version of the base model. To draw an analogy, SIMS can be interpreted as a method to create a bad version of the base model by training on its own synthetic data.

4 Experimental Results

In this section, we present the results of an array of computational experiments with SIMS. In Section 4.1 we demonstrate that SIMS makes significant progress on open question Q1 from the Introduction by self-improving the modeling performance of large-scale diffusion models using self-synthesized data. In Section 4.2 we demonstrate that SIMS makes significant progress on open question Q2 from the Introduction by acting as a prophylactic against MADness. In Section 4.3 we show how SIMS can adjust a diffusion model's synthetic data distribution to match any desired in-domain target distribution to mitigate biases and ensure model fairness.

4.1 Self-Improving Diffusion Models

Experimental Setup. We use four diverse real image datasets $\mathcal{D}_\text{r}$ for performance evaluation: 32×32 resolution CIFAR-10 (50k images) (Krizhevsky and Hinton, 2009), 64×64 resolution FFHQ-64 (70k images) (Karras et al., 2019), 64×64 resolution ImageNet-64 (1.2M images), and 512×512 resolution ImageNet-512 (1.2M images) (Deng et al., 2009).

For CIFAR-10 and FFHQ-64, we use the unconditional Variance Preserving (VP) variant of the EDM diffusion model from (Karras et al., 2022) as the base model for SIMS. For ImageNet-64 and ImageNet-512, we use the conditional EDM2-S model from (Karras et al., 2024a). While we use RGB-space diffusion models for CIFAR-10, FFHQ-64, and ImageNet-64, the ImageNet-512 model operates as a latent diffusion model with a latent space dimensionality of $64 \times 64 \times 4$. For all experiments with ImageNet-512, we keep the encoder-decoder VAE fixed and use StabilityVAE (Rombach et al., 2022) as in (Karras et al., 2024a). For all models, we use Heun's second-order solver (Süli and Mayers, 2003) for the de-noising process as proposed in (Karras et al., 2022).

For each base model, we use the publicly available code and pre-trained model weights from (Karras et al., 2024a; 2022). To train each auxiliary model (recall Algorithm 1), we first generate $n_\text{s} = |\mathcal{S}|$ synthetic data samples from the base model and then fine-tune the base model using $\mathcal{S}$ and the same training configuration as the base model. We then discard $\mathcal{S}$. We generate internal synthetic

Table 1: **SIMS attains state-of-the-art image generation performance.** Image generation performance comparison between SIMS and image generation baselines on the CIFAR-10, FFHQ-64, ImageNet-64, and ImageNet-512 datasets. SIMS consistently improves upon the base models EDM-VP and EDM-S. Indeed, SIMS establishes the new state-of-the-art FID for CIFAR-10 and ImageNet-64 (bold). We also compare the number of function evaluations (NFE) required for inference and the number of parameters (Million parameters, Mparams) for each model.

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

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 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

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

datasets of a scale similar to the real training data used for the pre-trained base models: $n_s = 100$k synthetic samples for CIFAR-10 and FFHQ-64 and $n_s = 1.5$M synthetic samples for both ImageNet resolutions.

To estimate the distance $\mathrm{dist}(\mathcal{G}, p_r)$ between the synthetic data distribution $\mathcal{G}$ and the real data distribution p_r, we use the Fréchet Inception Distance (FID) (Heusel et al., 2017). For all generative models and datasets, we generate 50k samples for evaluation, unless stated otherwise. Unless specified, we quote the paper-reported metrics for the baseline methods in our comparisons.

Quantitative Results. To demonstrate that SIMS achieves self-improvement, we need to show that the SIMS diffusion model produced by Algorithm 1 outperforms the base model. In Figure 4, we plot the FID between the SIMS model and the real data distribution as a function of the guidance strength parameter ω and the training budget $\mathcal{B}$ as measured by the number of million-images-seen (Mi) during fine tuning of the auxiliary model. In the top row, $\omega = 0$ corresponds to no guidance, which establishes the FID attained by the base model. The key takeaway from Figure 4 is that, across all four datasets, even a small negative guidance ω and a small amount of fine-tuning (small Mi) results in a SIMS model that outperforms the base model. Moreover, for properly tuned guidance and training budget, the self-improvement can be substantial: for CIFAR-10, FFHQ-64, ImageNet-64, and ImageNet-512, SIMS yields a relative FID self-improvement of 32.5%, 56.9%, 41.8%, and 32.4%, respectively.

Qualitative Results. Figure 1 presents example images generated by the pre-trained EDM2-S base model (top row), the base model after fine-tuning for 102Mi with 1.5M images synthesized from the base model (middle-row), and SIMS using the same 1.5M synthetic samples from the base model with a guidance strength of $\omega = 0.7$. For all three models, we start with the same initial latent vectors. We see that SIMS qualitatively improves the generated samples in each case. Appendices B–E present additional qualitative comparisons for the CIFAR10, FFHQ-64, ImageNet-64, and ImageNet-512 datasets.

Baseline Comparison. Table 1 compares the results obtained by SIMS with several standard diffusion based image generation baselines, including ADM (Dhariwal and Nichol, 2021) optionally used with classifier guidance (ADM-G), RIN (Jabri et al., 2023), EDM2-{S,M,L,XL} (Karras et al., 2024a), DDPM (Ho et al., 2020), EDM-VP (Karras et al., 2022), NCSN++ with improved sampling (Song et al., 2021), latent score based model (Vahdat et al., 2021). We also compare with generative

adversarial networks (GANs) such as StyleGAN-XL (Sauer et al., 2022) and StyleGAN-2-ADA (Karras et al., 2020). Additionally, we compare with methods that similar to SIMS, improve the performance of a base model, such as the distilled single step diffusion models GDD and GDD-I (Zheng and Yang, 2024), discriminator guided models EDM-G++ and LSGM-G++ (Kim et al., 2023), and the EDM2-{S,XL} models guided by Autoguidance (Karras et al., 2024b). Note that, for all the aforementioned methods, we present their paper-reported metrics in the table. For ImageNet-64 SIMS with EDM2-S and for CIFAR-10 SIMS with EDM-VP outperforms all of the baseline methods and reaches the new state-of-the-art FIDs of 0.92 and 1.33, respectively, representing a relative improvement of 8.9% and 13.6% over the closest baseline methods, Autoguidance-S and GDD-I.

Here are two highlights from Table 1. First, EDM2-S equipped with SIMS surpasses the performance of EDM2-XL by a significant margin for both ImageNet-64 and ImageNet-512, demonstrating that scaling the number of parameters cannot match the performance obtained by training an auxiliary model with synthetic data. Second, SIMS outperforms discriminator guidance (EDM-G++ and LSGM-G++) by a significant margin for both CIFAR-10 and FFHQ-64, demonstrating that reducing the probability under the synthetic data distribution at each denoising step outperforms increasing the realism score via a discriminator. For ImageNet-512, while EDM2-S with SIMS outperforms EDM2-S, SIMS is outperformed by Autoguidance.

4.2 MAD Prevention using SIMS

A fundamental assumption in training a generative model is that the training dataset $\mathcal{D}$ consists exclusively of data that aligns with the ground-truth target distribution. When synthetic data generated by previous models is naïvely included in $\mathcal{D}$ in a self-consuming loop, the the supposed "ground-truth" distribution becomes increasingly distorted and ultimately goes MAD. In this section we study the abilities of SIMS to mitigate and even prevent MADness.

4.2.1 Two dimensional Gaussian Data in a Synthetic Augmentation Loop

We now use a simple low-dimensional experiment to demonstrate the effectiveness of SIMS in *preventing* the negative impacts of synthetic data training that can lead to MADness. Recall from Section 2 that demonstrating that SIMS prevents MAD for a sequence of models $(\mathcal{G}^t)_{t\in\mathbb{N}}$ in a self-consuming loop requires showing that $\mathbb{E}[\mathrm{dist}(\mathcal{G}^\infty, p_\mathrm{r})] \leq \mathbb{E}[\mathrm{dist}(\mathcal{G}^1, p_\mathrm{r})]$.

Experimental Setup. We start with the task of learning a simple two-dimensional Gaussian distribution $p_\mathrm{r} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean $\boldsymbol{\mu} = [0, 0]^\top$ and covariance $\boldsymbol{\Sigma} = [2, 1; 1, 2]$ using a DDPM diffusion model Ho et al. (2020); Álvaro Jiménez (2023). We sample a real dataset $\mathcal{D}_\mathrm{r}$ of size $|\mathcal{D}_\mathrm{r}| = 1000$ from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and train the base model $\mathcal{G}^1 = \mathcal{A}(\mathcal{D}_\mathrm{r})$. We then form a synthetic augmentation loop, where for generation t of the loop, $\mathcal{G}^t = \mathcal{A}(\mathcal{D}_\mathrm{r} \cup \mathcal{D}_\mathrm{s}^{t-1})$, where $\mathcal{D}_\mathrm{s}^{t-1}$ is synthetic data generated from the previous generation model $\mathcal{G}^{t-1}$. We quantify the performance of the models in terms of the Wasserstein distance $\mathrm{dist}(\cdot, \cdot)$ between the synthetic and real data distributions $\mathbb{E}[\mathrm{dist}(\mathcal{G}^t, p_\mathrm{r})]$.

We compare two different training approaches:

- **Standard training**, where we train the generation-t model on the dataset $\mathcal{D}^t = \mathcal{D}_\mathrm{r} \cup \mathcal{D}_\mathrm{s}^{t-1}$ in which the real data is *polluted* with synthetic data from the previous generation.
- **SIMS**, where we train the generation-t base model on the polluted dataset $\mathcal{D}^t$.

For both approaches, we trained the base model for 100 epochs on $\mathcal{D}_\mathrm{r}$. For SIMS, we obtained the auxiliary model at generation t by fine-tuning the base model for 50 epochs using $n_\mathrm{s} = |\mathcal{S}| = 2000$ data points synthesized from the base model. We calculated expectations over 1000 independent runs, with each run starting with a new real dataset $\mathcal{D}_\mathrm{r}$ drawn from p_r and continuing the synthetic augmentation loop for 100 generations. When there is no guidance ($\omega = 0$), standard training and SIMS coincide and produce identical models.

Results. First, we confirm SIMS's *self-improvement*. Figure 5 top left plots the expected Wasserstein distance $\mathbb{E}[\mathrm{dist}(\mathcal{G}^1, p_\mathrm{r})]$ for the first generation model $\mathcal{G}^1 = \mathcal{A}(\mathcal{D}_\mathrm{r})$ for various values of ω in SIMS. We see clearly that SIMS has exploited its self-synthesized data to self-improve over the base model. trained on purely real data (there is no synthetic data pollution in generation 1).

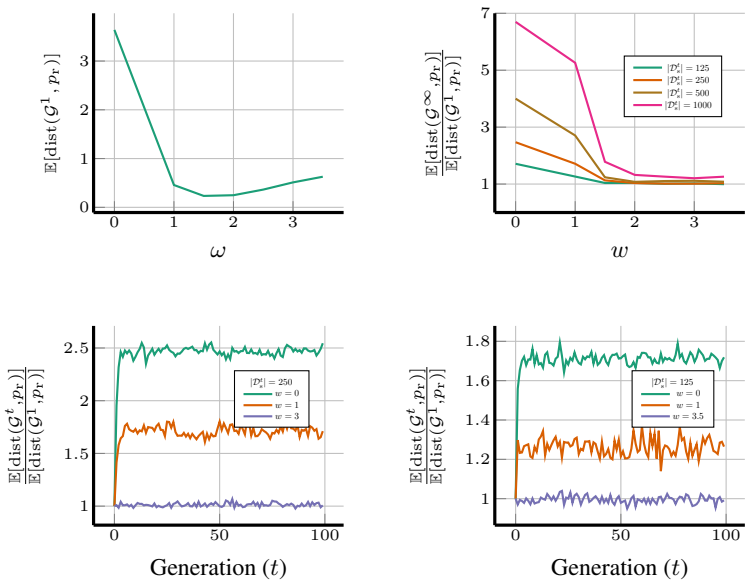


Figure 5: **SIMS simultaneously self-improves and prevents MADness in the synthetic augmentation self-consuming loop.** We compare standard synthetic augmentation training (Alemohammad et al., 2023; 2024) to SIMS training in a synthetic augmentation loop across 100 generations for two-dimensional Gaussian data. Standard training corresponds to guidance $\omega = 0$ in all cases. At top left, we confirm SIMS's *self-improvement* by noting that, for a wide range of ω, the expected Wasserstein distance $\mathbb{E}[\text{dist}(\mathcal{G}^1, p_r)]$ between the first generation model $\mathcal{G}^1 = \mathcal{A}(\mathcal{D}_r)$ and the real data distribution drops. At the bottom, we confirm that SIMS can act a *prophylactic for MADness*. We plot $\frac{\mathbb{E}[\text{dist}(\mathcal{G}^t, p_r)]}{\mathbb{E}[\text{dist}(\mathcal{G}^1, p_r)]}$, the ratio of the expected Wasserstein Distance at generation t to that at generation 1 for $|\mathcal{D}_s^t| = 250$ and 125. The green/orange/purple curves correspond to weak MADness mitigation/strong MADness mitigation/MADness prevention. At top right, we plot the normalized expected Wasserstein distance at convergence as a function of ω for four different synthetic data sizes $|\mathcal{D}_s^t|$. A guidance parameter of $\omega \approx 3$ results in either strong MADness mitigation or complete MADness prevention.

Next, we confirm that SIMS can act a *prophylactic against MADness*. In Figure 5 bottom, we plot $\frac{\mathbb{E}[\text{dist}(\mathcal{G}^t, p_r)]}{\mathbb{E}[\text{dist}(\mathcal{G}^1, p_r)]}$, the ratio of the expected Wasserstein Distance at generation t to that at generation 1, over 100 synthetic augmentation loop generations for two synthetic dataset sizes: $|\mathcal{D}_s| = 250$ and 125. With standard training ($\omega = 0$, green curves), we observe that the Wasserstein distance ratio quickly increases to a value much larger than 1, confirming MADness. In words, the performance of models that aggregate the real and synthetic data together and use standard training deteriorates with each generation t in the synthetic augmentation loop until it converges to a stable point, consistent with the findings regarding MADness mitigation in Bertrand et al. (2023); Gillman et al. (2024); Dohmatob et al. (2024b). However, as ω increases (orange curves), the SIMS Wasserstein distance ratio remains closer to 1, meaning that the negative impacts of synthetic training have been reduced. Moreover, for an optimized ω (purple curves), the SIMS Wasserstein distance ratio does not deviate from 1, meaning that MADness has been completely *prevented*.

To gain insight into the convergence limit for different ω, we calculated $\mathbb{E}[\text{dist}(\mathcal{G}^\infty, p_r)]$ by averaging $\{\mathbb{E}[\text{dist}(\mathcal{G}^t, p_r)]\}_{t=20}^{100}$ and plot its ratio to $\mathbb{E}[\text{dist}(\mathcal{G}^1, p_r)]$ in Figure 5 top right. The minimum values of $\frac{\mathbb{E}[\text{dist}(\mathcal{G}^\infty, p_r)]}{\mathbb{E}[\text{dist}(\mathcal{G}^1, p_r)]}$ over different ω for $|\mathcal{D}_s^t| = 125, 250, 500, 1000$ were $0.996, 1.013, 1.078, 1.204$, respectively. The corresponding ratios for standard data training were $1.71, 2.46, 3.99, 6.69$.

These results suggest that SIMS features a *prophylactic threshold* on the amount of synthetic data pollution, below which MADness prevention is possible but above which only MADness mitigation is possible. In this particular experiment, that threshold is approximately $|\mathcal{D}_s| = 250$. There are interesting parallels between this property and the fresh data threshold of the fresh data self-consuming loop in (Alemohammad et al., 2023; 2024). Exploring and characterizing this threshold are interesting avenues for further research.

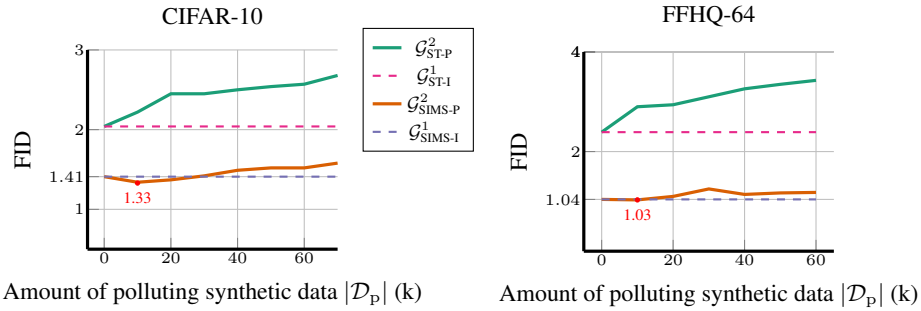


Figure 6: **SIMS acts as a prophylactic against MADness for realistic training datasets polluted with synthetic data.** For the CIFAR-10 (50k real images, left) and FFHQ-64 (70k real images, right) datasets, we plot the FID of the four training scenarios from Section 4.2.2 as a function of the amount of polluting synthetic data $|\mathcal{D}_{\text{p}}|$. While the modeling performance of standard training is strongly affected by increasing amounts of synthetic data pollution (compare $\mathcal{G}^2_{\text{ST-P}}$ to $\mathcal{G}^2_{\text{ST-I}}$), the performance of SIMS training is relatively immune (compare $\mathcal{G}^2_{\text{SIMS-P}}$ to $\mathcal{G}^2_{\text{SIMS-I}}$).

To summarize, *to the best of our knowledge, SIMS is the first synthetic-data learning algorithm that can prevent MAD in a self-consuming loop without injecting external knowledge.*

4.2.2 Realistic Data in a Synthetic Augmentation Loop

We continue our exploration of self-improvement and MADness prevention using realistic image data from the CIFAR-10 and FFHQ-64 datasets, large-scale diffusion models, and more pragmatic contexts regarding how the synthetic data enters the synthetic augmentation loop.

We compare four different training scenarios. The real dataset $\mathcal{D}_{\text{r}}$ (either CIFAR-10 or FFHQ-64) is the same in each scenario.

- **First generation, standard training with purely real data**, $\mathcal{G}^1_{\text{ST-I}}$: This scenario corresponds to training a primordial model using standard training and exclusively real data $\mathcal{D}_r$. As an archetype of today's lax data curation practices, data synthesized from $\mathcal{G}^1_{\text{ST-I}}$, which we denote by $\mathcal{D}_{\text{p}}$, pollutes the "real" training data of the last two second-generation models below.
- **Second generation, ideal SIMS training with purely real data**, $\mathcal{G}^1_{\text{SIMS-I}}$: This wishful, idealized scenario corresponds to how synthetic data training should be performed: by applying SIMS to self-improve the base model $\mathcal{G}^1_{\text{ST-I}}$ that was trained on purely real data.
- **Second generation, standard training with polluted real data,** $\mathcal{G}^2_{\text{ST-P}}$: This practical scenario corresponds to training a model using standard training with the *polluted* training data comprising the purely real data $\mathcal{D}_{\text{r}}$ combined with synthetic data $\mathcal{D}_{\text{p}}$ generated by $\mathcal{G}^1_{\text{ST-I}}$. We know from (Alemohammad et al., 2023; 2024) that this approach leads to MADness.
- **Second generation, SIMS training with polluted real data,** $\mathcal{G}^2_{\text{SIMS-P}}$: This practical scenario corresponds to training a model using SIMS training with the same polluted training data comprising the purely real data $\mathcal{D}_{\text{r}}$ combined with synthetic data $\mathcal{D}_{\text{p}}$ generated by $\mathcal{G}^1_{\text{ST-I}}$.

Experimental setup. For $\mathcal{G}^1_{\text{ST-I}}$, we used the EDM-VP models pre-trained on CIFAR-10 and FFHQ-64 from (Karras et al., 2022). For CIFAR-10, we trained both $\mathcal{G}^2_{\text{ST-P}}$ and the base model in $\mathcal{G}^2_{\text{SIMS-P}}$ from scratch for 200Mi. For FFHQ-64, to reduce computational costs, we fine-tuned $\mathcal{G}^1_{\text{ST-P}}$ and the base model in $\mathcal{G}^2_{\text{SIMS-P}}$ for 100Mi rather than training from scratch. For the training sets $\mathcal{S}$ of the auxiliary models in SIMS, we generated $|\mathcal{S}| = 100$k data from the corresponding base models. For each $|\mathcal{D}_{\text{p}}|$, we report the best FID for $\mathcal{G}^2_{\text{SIMS-P}}$ over various values of guidance ω and training budget $\mathcal{B}$ of the auxiliary model. The procedure for $\mathcal{G}^1_{\text{SIMS-I}}$ is identical to the self-improved models for CIFAR-10 and FFHQ-64 in Section 4.1, so we re-use those results here.

Results. Figure 6 plots the FIDs attained by the diffusion models learned by the four training scenarios above for the CIFAR-10 and FFHQ-64 datasets as we vary the amount of synthetic data $|\mathcal{D}_{\text{p}}|$ that is polluting the real training dataset. The same trends occur for both datasets. First, we

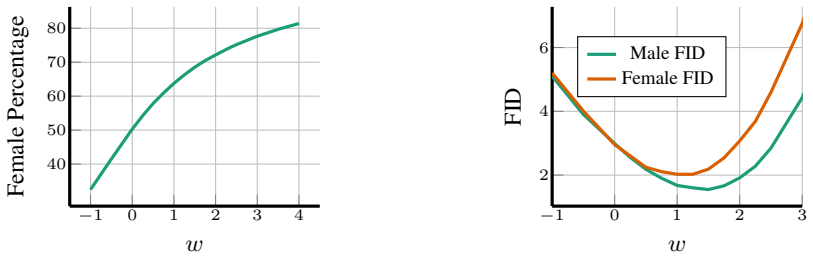


Figure 7: **SIMS can simultaneously shift the synthetic distribution to an arbitrary in-domain target distribution while self-improving the quality of generation.** (left) Percentage of female synthetic images for different values of the guidance ω. (right) FID of synthetic male and female images with respect to the male and female images in the FFHQ-64 dataset for different guidance levels ω.

see a substantial *self-improvement* in modeling performance from $\mathcal{G}^1_{\text{ST-I}}$ to $\mathcal{G}^1_{\text{SIMS-I}}$. Indeed, the drop in FID for CIFAR-10 from 1.41 (Section 4.1) to 1.33, *sets a new state-of-the-art FID benchmark for CIFAR-10 generation.* Second, we see that increasing amounts of polluting synthetic data $|\mathcal{D}_{\text{p}}|$ cause the performance of $\mathcal{G}^1_{\text{ST-P}}$ to diverge from $\mathcal{G}^1_{\text{ST-I}}$. Third, in contrast to standard training, the performance of SIMS training is relatively insensitive to the presence of polluting synthetic data in the base model, which indicates a *prophylactic* function against MADness. More precisely, the plots indicate that, for $|\mathcal{D}_{\text{p}}| < 30$k with CIFAR-10 (60% of $|\mathcal{D}_{\text{r}}|$) and $|\mathcal{D}_{\text{p}}| < 15$k for FFHQ-64 (20% of $|\mathcal{D}_{\text{r}}|$), SIMS not only prevents MADness in the second generation models but also achieves a self-improved FID by somehow exploiting the polluting synthetic data from the previous generation in its training set. The reason for this behavior remains an interesting open research question.

Our findings have potential implications for the future of diffusion generative models. Previous research has surfaced a "first mover" advantage for generative models, whereby large models trained early on real internet data will have a performance edge over later models trained on a mix of real and synthetic data from earlier generation models (Alemohammad et al., 2023; 2024; Shumailov et al., 2024). This advantage for standard training is evident in Figure 6, where the FID scores of the models degrade as the proportion of synthetic data increases. In contrast, and somewhat surprisingly, with SIMS training, model performance can actually improve when a small amount of synthetic data pollutes the training data.

4.3 Distribution Shifts with SIMS

Often, the datasets used for training AI models follow a distribution p that differs from some desired target distribution $\widehat{p}$. Consequently, the synthetic data distribution generated by a model will also reflect this discrepancy. This technical issue underlies why generative models tend to synthesize biased samples related to demographic factors such as gender and race, which leads to inaccurate representations across these attributes and potentially decreased fairness (Friedrich et al., 2023).

In this section, we demonstrate that SIMS can align the distribution of its generated images with an arbitrary in-domain target distribution $\widehat{p}$ that is distinct from the model's training data distribution p. Simultaneously, we aim to enhance the quality of individual samples. By doing so, SIMS has the potential to not only self-improve but also mitigate extant biases in a base model by shifting the model distribution towards a different distribution that promotes fairness.

We highlight SIMS' abilities for simultaneous self-improvement and distribution shifting with an example of altering group representation frequency using the FFHQ-64 dataset. This dataset comprises 70k images of faces varying in gender, age, and race, with an almost equal split of male and female subjects (51% female and 49% male). The pre-trained EDM-VP model trained on FFHQ-64 in (Karras et al., 2022) generate synthetic samples that are 50.3% perceived female and 49.7% perceived male (Karkkainen and Joo, 2021). This type of generation is arguably fair to both genders, but to demonstrate SIMS' ability to adapt to an arbitrary target distribution, our goal is to construct a model that overrepresents females compared to males, changing the percentage to 70% female and 30% male.

EDM-VP baseline, $\omega = 0$, 50.3% female

SIMS, $\omega = 1.5$, 68.5% female

Figure 8: **Distribution shifting with SIMS.** (left) Sample images synthesized from the pre-trained baseline diffusion model EDM-VP from (Karras et al., 2022) trained on the FFHQ-64 dataset are approximately 50% female. (right) Sample images synthesized using SIMS targeting a distribution shift to approximately 70% female. We used the same seed and randomness for both models to highlight the distribution shift.

The synthetic samples we constructed in Section 3 were generated without any intervention in order to match the distribution of the base model's synthetic data. Now, we generate samples and use the pre-trained classifier from (Karkkainen and Joo, 2021) to label the perceived genders of the generated faces. Using this information, we construct a synthetic dataset of 140k images containing 70% male and 30% female images. Since the score function of the auxiliary model $\boldsymbol{s}_{\theta_s}(\boldsymbol{x}_t, t)$ is used as a negative guidance, the distribution generated by the auxiliary model should be the complement of the target distribution $\widehat{p}$. Executing SIMS, we obtain the auxiliary model by fine-tuning the pre-trained diffusion model on FFHQ-64 for 50Mi and then combining the score functions of the base and auxiliary diffusion models with guidance strength ω.

Results. Figure 7 (left) presents the evidence on distribution shifting. It plots the percentage of females with respect to the guidance ω. For $\omega = -1$, we sample only from the auxiliary model, which has been trained on a synthetic dataset of 70% males and 30% females, generating 32% female images. For $\omega = 0$, we sample from the base model and obtain 50% female images. As ω increases, the percentage of females increases, reaching approximately 68% at $\omega = 1.5$.

To asses the quality of image generation, we provide two FID measures: one between synthetic male images and real male images in FFHQ-64, and one between synthetic female images and real female images in FFHQ-64, using 35k synthetic images for each gender. To identify the gender of the synthesized, we again use the pre-trained classifier from (Karkkainen and Joo, 2021).

Figure 7 (right) presents the evidence on *simultaneous self-improvement*. It plots the FID scores for the synthesized male and female images. The FID follows a bowl-shaped pattern similar to the plots in Section 4.1. The minimum FID for male images occurs at $\omega = 1.5$, which coincides with the parameter value that achieves approximately 70% female generation. However, the minimum FID for female images is reached at a slightly lower value of $\omega = 1.25$. This indicates that optimizing both the target distribution shift and the quality of image generation may not necessarily align at the same ω value.

Figure 8 plots sample synthetic images for the baseline model (left) and the final model that are both distribution-shifted and self-improved (right).

5 Discussion

In this paper, we have developed *Self-IMproving diffusion models with Synthetic data* (SIMS), a new training algorithm for generative AI models designed to enhance the performance of diffusion models by using their own synthetic data. The key idea is to avoid aggregating the real and synthetic data together into one training dataset — which can lead to a divergence between the model's distribution and real-world data (MADness (Alemohammad et al., 2023; 2024; Shumailov et al., 2024)) that diminishes model quality and reinforces biases — and instead use the synthetic data to provide

negative guidance during the generation process to steer a model's generative process away from the non-ideal synthetic data manifold and towards the real data distribution. SIMS provides affirmative responses to questions Q1 and Q2 posed in the Introduction. In particular, (Q1) SIMS establishes new records for realistic synthetic data distribution on two important image datasets (CIFAR-10 and ImageNet-64), while (Q2) to the best of our knowledge, SIMS is the first generative AI model that can be iteratively trained on self-generated, synthetic data without going MAD. As an added bonus, SIMS can adjust a diffusion model's synthetic data distribution to match any desired in-domain target distribution, helping mitigate biases and ensure model fairness.

Our experiments have revealed that there is a *prophylactic threshold* on the ratio of the amount of synthetic data to the amount of real data that SIMS can tolerate before it is incapable of fully preventing MADness. According to our experiments in Section 4.2, this threshold is approximately 60% for CIFAR-10, 20% for FFHQ-64, and 25% for two-dimensional Gaussian data. Regardless, above this threshold, SIMS continues to mitigate MADness even if it cannot fully prevent it.

As synthetic data continues to proliferate online, naïve, unsupervised data collection will someday result in the amount of synthetic data exceeding the prophylactic threshold in standard training datasets and rendering methods like SIMS less effective at preventing MADness. Hence, careful dataset curation using recent advances in watermarking and synthetic data detection (Bui et al., 2023a;b; Wen et al., 2023) will be crucial for keeping the amount of synthetic data low enough in tomorrow's training datasets.

The SIMS concept and our experimental results point towards a number of interesting open research questions. We sketch out four of them here.

First, we conjecture that SIMS's performance might be similar if the auxiliary model differs from the base model but matches the base model's performance across the data domain (e.g., employ two different state-of-the-art diffusion models in Algorithm 1). Confirming this could result in new negative-guidance-based training algorithms that are *broad spectrum prophylactics* against synthetic data from a range of different generative models.

Second, it seems important to understand why SIMS does not just tolerate but capitalizes on synthetic data that is polluting the real training dataset employed by its base model. This suggests that negative-guidance-based training algorithms have unexplored generalizability properties.

Third, models beyond diffusion models can likely be equipped with self-improvement and MADness prophylactic capabilities. For instance, we can fine-tune a generative adversarial network (GAN) or variational autoencoder (VAE) base model with its own synthetic data and then design a latent-space sampler like Polarity (Humayun et al., 2022) to reduce the density of generated samples under the synthetic distribution.

Fourth, extending our results on distribution shift, we can envision extending SIMS to engage with users through synthetic data and collect feedback to adjust the model's distribution to align with user preferences. Interestingly, since the auxiliary model in SIMS needs to be trained on a synthetic dataset with complementary characteristics to the target distribution, the synthetic data should be curated based on what users do *not* prefer.

6 Acknowledgements

SA contributed to this work while he was an intern at Adobe research during summer 2024. SA, AH and RB were supported by NSF grants CCF-1911094 and IIS-1730574; ONR grants N00014-18-1-2571, N00014-20-1-2534, N00014-23-1-2714, N00014-24-1-2225, and MURI N00014-20-1-2787; AFOSR grant FA9550-22-1-0060; DOE grant DE-SC0020345; DOE grant 140D0423C0076; and a Vannevar Bush Faculty Fellowship, ONR grant N00014-18-1-2047. SA was partially supported by a Ken Kennedy Institute 2023–24 BP Graduate Fellowship.

References

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*, 2024.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024a.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G Baraniuk. Self-consuming generative models go MAD. *arXiv preprint arXiv:2307.01850*, July 2023.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go MAD. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ShjMHfmPs0.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

YahooFinance. AI's 'mad cow disease' problem tramples into earnings season. *Yahoo Finance*, April 2024. URL https://bit.ly/3Z6U25B.

The Economist. The bigger-is-better approach to AI is running out of road. *The Economist*, June 2023a. URL https://bit.ly/3AIPng8.

The Economist. Large, creative AI models will transform lives and labour markets. *The Economist*, April 2023b. URL https://bit.ly/4dxG80N.

Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv preprint arXiv:2211.04325*, 2022.

Sierra Wyllie, Ilia Shumailov, and Nicolas Papernot. Fairness feedback loops: training on synthetic data amplifies bias. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2113–2147, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Thirty-first Conference on Neural Information Processing Systems*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Thirty-third Conference on Neural Information Processing Systems*, 2019.

Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 1982.

Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. On the stability of iterative retraining of generative models on their own data. *arXiv preprint arxiv:2310.00429*, 2023.

Yunzhen Feng, Elvis Dohmatob, Pu Yang, Francois Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*, 2024a. URL https://openreview.net/forum?id=dE8BznbvZV.

Elvis Dohmatob, Yunzhen Feng, and Julia Kempe. Model collapse demystified: The case of regression. *arXiv preprint arXiv:2402.07712*, 2024a.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, et al. Is model collapse inevitable? Breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*, 2024.

Yunzhen Feng, Elvis Dohmatob, Pu Yang, Francois Charton, and Julia Kempe. Beyond model collapse: Scaling up with synthesized data requires reinforcement. *arXiv preprint arXiv:2406.07515*, 2024b.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. RL on incorrect synthetic data scales the efficiency of LLM math reasoning by eight-fold. *arXiv preprint arXiv:2406.14532*, 2024.

Nate Gillman, Michael Freeman, Daksh Aggarwal, Chia-Hong HSU, Calvin Luo, Yonglong Tian, and Chen Sun. Self-correcting self-consuming loops for generative model training. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=i0nVanexij.

Benjamin Biggs, Arjun Seshadri, Yang Zou, Achin Jain, Aditya Golatkar, Yusheng Xie, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Diffusion soup: Model merging for text-to-image diffusion models. *arXiv preprint arXiv:2406.08431*, 2024.

Álvaro Jiménez. Toy-diffusion, 2023. URL https://github.com/albarji/toy-diffusion?tab=readme-ov-file.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=dNigytemkL.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Thirty-fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt.

Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7280–7290, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Dongjun Kim, Yeongmin Kim, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Refining generative process with discriminator guidance in score-based diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 16567–16598. PMLR, 2023. URL https://proceedings.mlr.press/v202/kim23i.html.

Donghoon Ahn, Hyoungwon Cho, Jaewon Min, Wooseok Jang, Jungwoo Kim, SeonHwa Kim, Hyun Hee Park, Kyong Hwan Jin, and Seungryong Kim. Self-rectifying diffusion sampling with perturbed-attention guidance. *arXiv preprint arXiv:2403.17377*, 2024.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *arXiv preprint arXiv:2406.02507*, 2024b.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Thirty-sixth Conference on Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Endre Süli and David F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, 2003.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Thirty-fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=P9TYG0j-wtG.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Bowen Zheng and Tianming Yang. Diffusion models are innate one-step generators. *arXiv preprint arXiv:2405.20750*, 2024.

Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. In *ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH '22, New York, NY, USA, 2022.

Allan Jabri, David J. Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=KVvku47shW.

Felix Friedrich, Manuel Brack, Lukas Struppek, Dominik Hintersdorf, Patrick Schramowski, Sasha Luccioni, and Kristian Kersting. Fair diffusion: Instructing text-to-image generation models on fairness. *arXiv preprint arXiv:2302.10893*, 2023.

Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1548–1558, 2021.

Tu Bui, Shruti Agarwal, and John Collomosse. Trustmark: Universal watermarking for arbitrary resolution images. *arXiv preprint arXiv:2311.18297*, 2023a.

Tu Bui, Shruti Agarwal, Ning Yu, and John Collomosse. Rosteals: Robust steganography using autoencoder latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 933–942, 2023b.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Z57JrmubNl.

Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. Polarity sampling: Quality and diversity control of pre-trained generative networks via singular values. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10641–10650, June 2022.

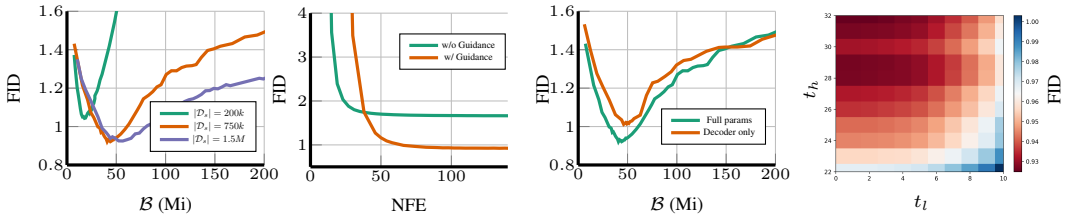


Figure 9: **Left**: training the auxilary model score function $\boldsymbol{s}_{\theta_s}(\boldsymbol{x}, t)$ using synthetic datasets of varying size for ImageNet-64. Increasing synthetic dataset size helps obtain better FID during self-improvement with diminishing returns. **Middle-left**: FID for different number of function evaluations (NFE). **Middle-right** Reducing the number of learnable parameters during auxiliary model fine-tuning. **Right** Changing the guidance interval for SIMS. Early and late denoising steps can be ignored with a minimal drop in FID.

A Ablation Studies for SIMS

In this section, we present ablations on the synthetic dataset size used for training the auxiliary model, FID for different number of function evaluations, and strategies for reducing number of function evalutions during inference.

Synthetic dataset size. For ImageNet-64, we change the dataset size used for training the auxiliary model score function $\boldsymbol{s}_{\theta_s}(\boldsymbol{x}, t)$, and present the FID over training budget. In Figure 9 (left), we see that increasing the dataset size allows obtaining better FID. However note that if $|\mathcal{D}_s| \to \infty$, $\boldsymbol{s}_{\theta_s}(\boldsymbol{x}, t) \to \boldsymbol{s}_{\theta_r}(\boldsymbol{x}, t)$, i.e., the score functions become identical and negative guidance yields no gain. Therefore increasing the synthetic dataset further to very large numbers may result in an decrease in FID.

Number of function evaluations. Number of function evaluations (NFE) refer to the number of times a score function is evaluated during denoising. For ImageNet-64 we compare NFE for the EDM2-S base model with and without SIMS. In Figure 9 (middle left), we see that naturally, with SIMS we need more function evaluations to achieve the lowest FID. At NFE$= 40$, FID for both with and without guidance cases are almost equal to 1.70. For the SIMS we use a guidance strength of $\omega = 0.9$ and the best FID auxiliary model trained upto 56 Mi seen during training.

Reducing number of function evaluations. For a fixed denoising step, SIMS uses twice the number of function evaluations (NFE) compared to the baseline method without any guidance. This results in doubling the inference time computation. We propose two strategies to reduce the NFE overhead.

The EDM model architecture consists of an encoder and a decoder, each responsible for half of the computations for one function evaluation. As illustrated in Figure 9 (middle right), during the fine-tuning of the base model, we froze the weights of the encoder and trained only the decoder part. At inference time, the encoder is shared between the base model and the auxiliary model, differing only in the decoder. Consequently, the effective number of function evaluations decreases from 2x to 1.5x. We observe that training only the decoder to obtain the auxiliary model slightly increases the minimum FID from 0.92 to 1.01 during fine-tuning while reducing the NFE from 2 to 1.5.

The second strategy involves applying guidance from the auxiliary model for a limited interval. To assess the impact of this guidance at different denoising steps, we compute the FID for SIMS with guidance applied to a limited interval (t_l, t_h), rather than the default setting of $(0, 32)$. As shown in Figure 9 (right), guidance is more crucial during the final denoising steps compared to the earlier ones. The results indicate that we can exclude the first 10 steps in the denoising process with only a minimal drop in FID, from 0.93 to 0.96. Utilizing the auxiliary model for guidance over a smaller number of intervals can effectively reduce inference time and costs.

B CIFAR-10 SYNTHESIZED IMAGES

SIMS: $w = 0.8$, Training budget: 40 Mi

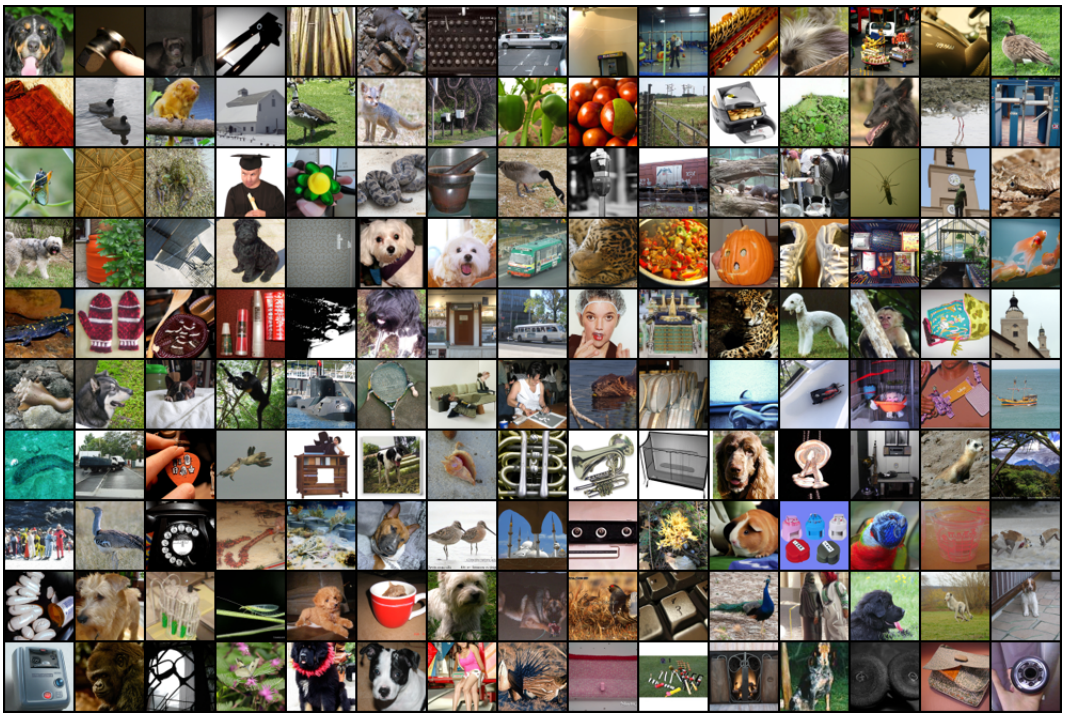
Base Model

C FFHQ-64 Synthesized Images

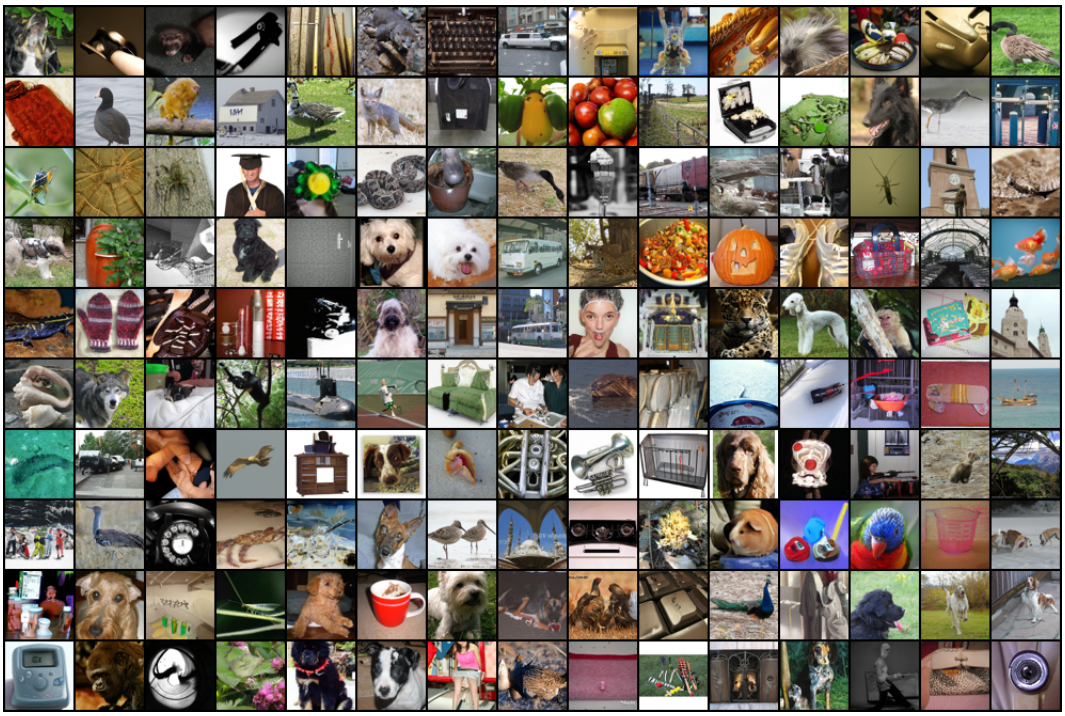
SIMS: $w = 1.5$, Training budget: 34 Mi

Base Model

D IMAGENET-64 SYNTHESIZED IMAGES



SIMS: $w = 0.9$, Training budget: 56 Mi



Base Model

E IMAGENET-512 SYNTHESIZED IMAGES

SIMS: $w = 0.7$, Training budget: 102 Mi

Base Model

F STANDARD TRAINING

Algorithm 2 Standard Training Procedure

Input: Training dataset $\mathcal{D}$

1: **Train diffusion model**: Use dataset $\mathcal{D}$ to train the diffusion model using standard training, resulting in the score function $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$.

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

The procedure of standard training is shown in Algorithm 2. Compared to SIMS (Algorithm 1), standard training is essentially the same as using only the base diffusion model's score function to generate synthetic data, which is equivalent to setting $\omega = 0$ in SIMS. It's important to note that if you already have a model trained using the standard approach, you can still apply steps 2-4 of SIMS to develop a self-improved model.