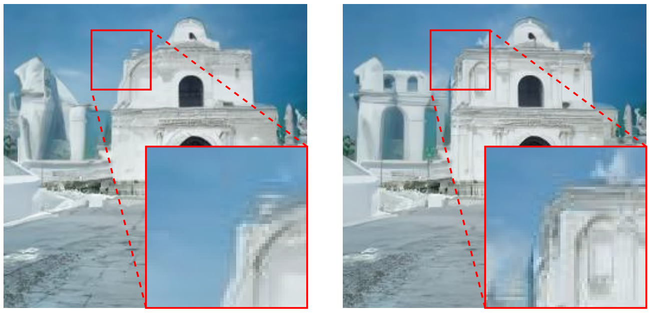
Boosting Diffusion Models with an Adaptive Momentum Sampler

Xiyu Wang, Anh-Dung Dinh, Daochang Liu, Chang Xu
School of Computer Science, Faculty of Engineering, The University of Sydney, Australia
xiyuwang.usyd@gmail.com, adin6536@uni.sydney.edu.au,
daochang.liu@sydney.edu.au, c.xu@sydney.edu.au

Abstract

Diffusion probabilistic models (DPMs) have been shown to generate high-quality images without the need for delicate adversarial training. The sampling process of DPMs is mathematically similar to Stochastic Gradient Descent (SGD), with both being iteratively updated with a function increment. Building on this, we present a novel reverse sampler for DPMs in this paper, drawing inspiration from the widely-used Adam optimizer. Our proposed sampler can be readily applied to a pre-trained diffusion model, utilizing momentum mechanisms and adaptive updating to enhance the generated image's quality. By effectively reusing update directions from early steps, our proposed sampler achieves a better balance between high-level semantics and low-level details. Additionally, this sampler is flexible and can be easily integrated into pre-trained DPMs regardless of the sampler used during training. Our experimental results on multiple benchmarks demonstrate that our proposed reverse sampler yields remarkable improvements over different baselines.



(a) DDPM

(b) Our (DDPM + AM)

Figure 1: We introduce a new sampler featuring adaptive momentum (AM). Compared to existing reverse sampling processes in DPMs (left), our method (right) is more efficient in recovering high-level components while effectively minimizing noise in the generated images. This enhancement leads to clearer and more accurate image generation, setting our approach apart from current methodologies.

1 Introduction

Deep-generative modeling has emerged as a popular area of research due to its significance in comprehending and managing data. In image generation, GANs have dominated the field [Odena *et al.*, 2017; Gong *et al.*, 2019; Brock *et al.*, 2018; Karras *et al.*, 2020; Dung and Binh, 2022] since its birth in Goodfellow *et al.*. Compared to other log-likelihood generative models [Kingma and Welling, 2013; Hinton and Salakhutdinov, 2006; Shao *et al.*, 2021; Guo *et al.*, 2020; Wang *et al.*, 2023], GANs show superiority over them in terms of both quality and diversity. However, the employment of adversarial training in GANs causes instability during its training and mode collapse, requiring specific optimization techniques and architectures.

As an alternative to adversarial training, iterative models such as Diffusion Probabilistic Models (DPMs) [Ho *et al.*, 2020; Song *et al.*, 2020b] have emerged as promising option. Denoising Diffusion Probabilistic Model for Diffusion (DDPM) [Ho *et al.*, 2020] and its variants [Nichol and Dhariwal, 2021; Song *et al.*, 2020a] are one of the streams based on the iterative process whose process includes two phases. The first phase is to diffuse the image into a predefined Gaussian noise, and the second phase will try to reverse the trajectory to recover the image. As a recent development, Denoising Diffusion Implicit Model (DDIM) [Song *et al.*, 2020a] devises an accelerated version by turning DDPM into an implicit process and facilitates the reduction of the number of time steps during image generation. Meanwhile, the score-based scheme [Song and Ermon, 2019; Song and Ermon, 2020; Song *et al.*, 2020b] is a theoretical version that works on the same mechanism. The main difference of the score-based model lies in its utilization of analytic tools, Stochastic Differential Equations (SDEs), to diffuse and denoise the image. Several higher-order solvers for diffusion models [Yu *et al.*, 2022; Karras *et al.*, 2022; Bruce *et al.*, 2024] enable faster sampling and the use of momentum in diffusion model training [Wu *et al.*, 2023]. The two iterative schemes of diffusion models and score-based models are unified by the crucial connection observed in DDIM between its optimization technique and Ordinary Differential Equations (ODEs). Therefore, we focus on the diffusion models since the two categories belong to the same broader generalization.

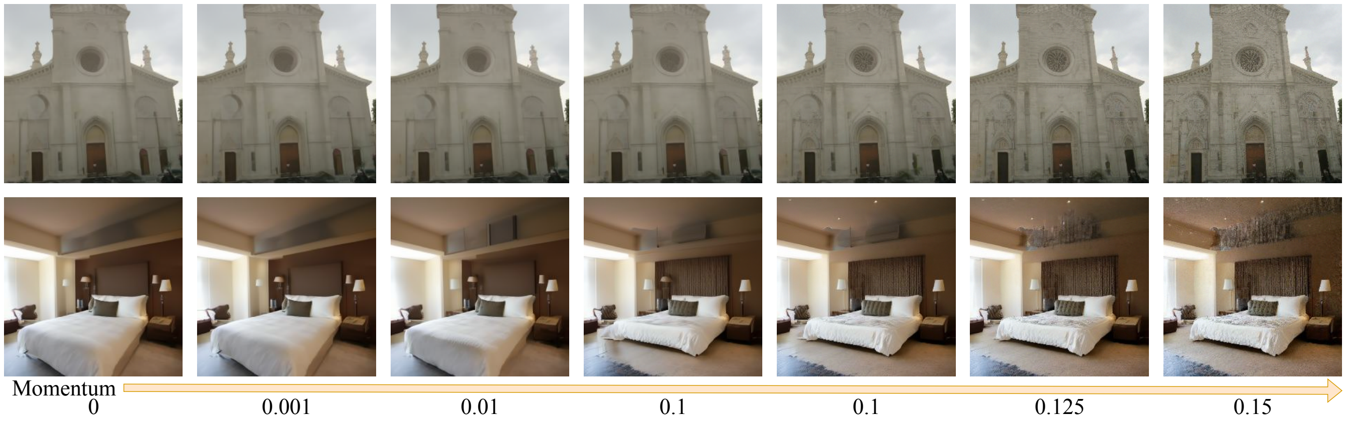


Figure 2: The images presented are generated from the final iteration, $\bar{x}_0$, after running 1000 steps of DDPM with our novel sampler with different scales of momentum. The momentum value (b) progressively increases from left to right (0 to 1.75), indicating an increasing reliance on earlier steps. With larger momentum, the final images prioritize intricate patterns and textures while also maintaining shapes and contours. By demonstrating the impact of varying momentum degrees, these final images provide an intuitive illustration of the adaptive momentum sampler's ability to improve low-level details while retaining high-level semantics.

The reverse process in DPMs can be likened to the annealed Langevin dynamics method, prevalent in diffusion sampling. The sampling process of DPMs shifts the sample in the gradient direction of the log probability density and incorporates a noise element. Similarly, Stochastic Gradient Descent (SGD) updates neural networks in the loss function gradient's direction. This similarity has been utilized to improve the diffusion models sampling in many works especially in diffusion guidance sampling [Dhariwal and Nichol, 2021; Dinh *et al.*, 2024; Dinh *et al.*, 2023; Liu *et al.*, 2023] Drawing inspiration from how the Adam [Kingma and Ba, 2014] optimizer improved SGD, our goal is to enhance the sample generation process of DPMs. To achieve this, we introduce a new training-free reverse sampler for diffusion models, which accumulates a velocity vector over steps to refine the update direction. Additionally, it maintains a moving average of second-order moments, allowing for an adaptive adjustment of the update pace, which effectively functions as a natural form of step-size annealing.

The first benefit of this new sampler is the foremost being a better trade-off between generating high-level semantics, such as shapes and outlines, and restoring low-level details, such as fine texture. To illustrate this point, we refer to Fig. 2, which depicts samples generated using different scales of momentum. As observed, employing a larger momentum that heavily reuses the update direction of early steps leads to augmented details while still preserving high-level information. This phenomenon aligns with the two stages of learning for likelihood-based models [Rombach *et al.*, 2022], where the first stage focuses on learning high-frequency details, while the second stage focuses on learning conceptual compositions. With a suitable momentum value, the adaptive momentum in our sampler coordinates these two learning stages, thus effectively balancing between high-level and low-level information.

The second benefit of the proposed sampler is its flexibility to be easily integrated with existing pre-trained diffusion models. This non-parameterized sampler requires no additional training, and its momentum is dynamically adjusted on the fly during sampling. This adaptability is particularly advantageous in mitigating the train-test gap, as the sampler can be easily plugged into diffusion models pre-trained with different samplers or customized settings.

Experimental results on five common benchmarks, CIFAR-10 [Krizhevsky *et al.*, 2009], CelebA [Liu *et al.*, 2018], ImageNet [Deng *et al.*, 2009], LSUN [Yu *et al.*, 2015] and CelebA-HQ [Karras *et al.*, 2017], show that the proposed adaptive momentum sampler improves both DDPM and DDIM in terms of generating quality. In summary, the contribution of this paper is to newly propose a balanced, flexible, and highly-performing sampler that can be easily applied to pre-trained diffusion models.

2 Preliminary

2.1 Denoising Diffusion Probabilistic Model

Generally, Gaussian diffusion models are utilized to approximate the data distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ by $p_\theta(\mathbf{x}_0)$. $p_\theta(\mathbf{x}_0)$ is modelled to be the form of latent variables models:

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}, \text{where}$$

$$p_\theta(\mathbf{x}_{0:T}) = p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta^{(t)}(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

where $x_1, x_2, ..., x_T$ are latent variables with the same dimensions with x_0.

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) \text{ , where} \tag{1}$$

$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right) .$$

Thus, the diffusion process from a data distribution to a Gaussian distribution for time step t can be expressed as:

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon \ , \tag{2}$$

where the $\alpha_t := \prod_{i=0}^{t}(1-\beta_i)$ and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The Ho *et al.*[Ho *et al.*, 2020] trains a U-net [Ronneberger *et al.*, 2015] model θ to fit the data distribution $\mathbf{x}_0$ by maximizing the following variational lower-bound:

$$\begin{aligned} &\underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{q(\mathbf{x}_0)}\left[-\log p_\theta\left(\mathbf{x}_0\right)\right] \leq \qquad (3) \\ &\underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T)}\left[\log q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right) - \log p_\theta\left(\mathbf{x}_{0:T}\right)\right] \ , \end{aligned}$$

where the $q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)$ is a certain inference distribution, which could be calculated by the Bayes theorem with $\mathbf{x}_0$ and $\mathbf{x}_T$, over the latent variable.

2.2 Denoising Diffusion Implicit Model

DDPM's dependence on the Markovian process has two main problems. First, due to the Markovian property, the generative process is forced to have a similar number of time steps to the diffusion process. Secondly, the stochastic process causes uncertainty in the synthetic images. This challenges the image interpolation in the latent space.

DDIM generalizes the DDPM as a Non-Markovian process. Different from the Eq. 1, x_{t-1} is conditioned by both x_t and x_0:

$$q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0) = q_\sigma(\mathbf{x}_T|x_0)\prod_{t=2}^{T} q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \tag{4}$$

Where $q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is chosen so that $q_\sigma(x_T|x_0) = \mathcal{N}(\sqrt{\alpha_T}x_0, (1-\alpha_T)\mathbf{I})$:

$$\begin{aligned} q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) =& \mathcal{N}(\sqrt{\alpha_{t-1}}\mathbf{x_0} + \\ & \sqrt{1-\alpha_{t-1}-\sigma_t^2}\,\frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_T}}, \sigma_t^2\mathbf{I}) \end{aligned}$$

From Eq. 2, $\mathbf{x}_t$ can be obtained by sampling $\mathbf{x}_0 \sim q(x_0)$ and $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$. By training the model $\epsilon_\theta^{(t)}$ to predict ϵ_t at each time step, x_0 predicting function $f_\theta^{(t)}$ is defined as:

$$f_\theta^{(t)}(\mathbf{x}_t) = \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{\alpha_T}}$$

This results in the whole generative process as:

$$p_\theta^{(t)}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \begin{cases} \mathcal{N}(f_\theta^{(1)}(\mathbf{x}_1), \sigma_1^2\mathbf{I}) & \text{if } t = 1 \\ q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, f_\theta^{(t)}(\mathbf{x}_t)) & \text{otherwise,} \end{cases}$$

θ is optimized as Eq. 3 with $q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0)$ is defined in Eq. 4. Based on the defined generative process, given a sample $\mathbf{x}_t$, we could sample the $\mathbf{x}_{t-1}$ as:

$$\begin{aligned} \mathbf{x}_{t-1} =& \sqrt{\alpha_{t-1}}\underbrace{\left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\mathbf{x}_t)}{\alpha_t}\right)}_{\text{predicted } \mathbf{x}_0} + \\ & \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^{(t)}(\mathbf{x}_t)}_{\text{direction pointing to } \mathbf{x}_t} + \underbrace{\sigma_t\epsilon_t}_{\text{random noise}} \qquad (5) \end{aligned}$$

where $\sigma_t = \eta\sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{1-\alpha_t/\alpha_{t-1}}$ and $\eta = 0$ [Song *et al.*, 2020a] or $\eta = 1$ [Song *et al.*, 2020a] or $\eta = \sqrt{(1-\alpha_t)/(1-\alpha_{t-1})}$ [Ho *et al.*, 2020]. Yang Song *et al.*[Song and Ermon, 2019] have proved that when $\eta = \sqrt{(1-\alpha_t)/(1-\alpha_{t-1})}$ the Eq. 5 is essentially a different discretization to the same reverse-time SDEs and Jiaming Song [Song *et al.*, 2020a] shows that when $\eta = 0$ the Eq. 5 similarity to Euler integration for solving ODEs. We provide when the $\eta = 1$ is the one type of DDPM, the reverse process is approximate solution of the same reverse-time SDEs in Appendix B.

3 Methodology

In this paper, we propose a training-free Adaptive Momentum Sampler to generate high-quality images, which can build long-term communications between the previously explored path and the current denoising direction. The first part of this section presents how to integrate the basic momentum method into the origin inference process of DDPM/DDIM. After that, we illustrate how to effectively adjust the pacing rate for denoising with the moving average of the squared prior increments. In each part, we also show the design of how to trade-off between high- and low-level information.

3.1 Momentum Sampler

We employ the momentum method to enhance the reverse process in diffusion models. This method is commonly utilized to accelerate gradient flow optimization in neural network training. By incorporating momentum, we aim to significantly improve the efficiency and effectiveness of the reverse sampling process in diffusion models, leading to more optimized outcomes. The memorization of previously searched directions contributes to the optimizer's overcoming narrow valleys, small humps. We find that the momentum method can also benefit the reverse process of diffusion SDEs and ODEs.

Inspired by SGD with momentum in optimizing neural networks, we replace the Euler-Maruyama method with the momentum method in the reverse process of diffusion models to avoid the plight mentioned. We find that the prediction of $\mathbf{x}_0$ is relatively stable and tractable during the sampling phase. Therefore, the sampling process contains some trend information, enabling our momentum sampler to utilize such information and improve the sampling procedure.

We first repeat the Eq. 5 as an incremental form of $\mathbf{x}$. For brevity, we let the $\bar{\mathbf{x}}_t = \frac{\mathbf{x}_t}{\sqrt{\alpha_t}}$ and the $\mu = \left(\sqrt{\frac{1-\alpha_{t-1}-\sigma_t^2}{\alpha_{t-1}}} - \sqrt{\frac{1-\alpha_t}{\alpha_t}}\right)$. And then, we can formulate the increment of $\bar{\mathbf{x}}_t$ and rewrite the DDPM (or DDIM) iteration Eq. 5, as:

$$\mathbf{d}\bar{\mathbf{x}}_t = \mu \cdot \epsilon_\theta^{(t)}(\mathbf{x}_t) + \frac{\sigma_t}{\sqrt{\alpha_{t-1}}} \cdot \epsilon_t \ , \tag{6}$$

$$\bar{\mathbf{x}}_{t-1} = \bar{\mathbf{x}}_t + \mathbf{d}\bar{\mathbf{x}}_t \ . \tag{7}$$

As demonstrated in the equation referenced as Eq. 7, the updating process in dispersed sampling can be viewed as an increment involving the current x_t and the previous x_{t-1}. Similarly, the SGD optimizer updates the network's parameters

(ω) in the direction of the loss function (Q), following the formula $\omega := \omega - \bigtriangledown Q(\omega)$. This reveals a strong similarity between the formulas used in SGD and the reverse process of diffusion models. Therefore, our objective is to adjust the reverse process in a way akin to how the SGD + momentum optimizer refines SGD, which involves enhancing the update mechanism in the reverse process to optimize the generation of samples in diffusion models. Instead of only relying on the single last step result to generate samples, we incorporate the effects of prior directions as follows:

$$\begin{aligned} \mathbf{m}_{t-1} &= a \cdot \mathbf{m}_t + b \cdot \mathbf{d}\bar{\mathbf{x}}_t \ , \\ \bar{\mathbf{x}}_{t-1} &= \bar{\mathbf{x}}_t + \mathbf{m}_{t-1} \ , \end{aligned}$$

where the a is the damping coefficient, and the a is a history-dependent velocity coefficient. Since the superposition property of the Gaussian distribution, we use the spherical difference, $a^2 + b^2 = 1$, to balance the momentum and current value. The $\mathbf{m}_t$ is the current momentum of t-th iteration, and the $\mathbf{m}_T = \mathbf{0}$. The strength of the momentum method is controlled by the a and b coefficients.

3.2 Adaptive Momentum Sampler

In this section, we study how to adapt the learning rate based on the running average of recent magnitudes of the increment, $\mathbf{d}\bar{\mathbf{x}}_t$. Besides the momentum-based sampling process only depending on the advantage of momentum by using a moving average, the idea of Root Mean Square Propagation [Mukkamala and Hein, 2017] (RMSProp) can also be applied to the probabilistic diffusion models generation process. To be more specific, we imitate RMSProp to automatically tailor the step size with a decaying average on each pixel. In this way, the process solver will focus on the most recently observed partial gradients and discard history from the extreme past.

We propose an Adaptive Momentum sampling process that adapts the learning rates of each pixel per time step The adaptive adjusting of denoising step size is based not only on the average first moment but also on the average of the second moments of the gradients. Based on the momentum sampling process, we use a $\mathbf{v_t}$ to store an exponentially decaying average of prior squared increment:

$$\begin{aligned} \mathbf{v}_{t-1} &= (1-c) \cdot \mathbf{v_t} + c \cdot \|\mathbf{d}\bar{\mathbf{x}}_t\|_{\mathrm{F}}^2 & (8) \\ \mathbf{m}_{t-1} &= a \cdot \mathbf{m}_t + b \cdot \mathbf{d}\bar{\mathbf{x}}_t & (9) \\ \bar{\mathbf{x}}_{t-1} &= \bar{\mathbf{x}}_t + \frac{\mathbf{m}_{t-1}}{\sqrt{\mathbf{v}_{t-1}} + \zeta} \ , \end{aligned}$$

where the c is the decay rate to control the moving averages of squared increment, and the $\mathbf{v_t}$ is the current averages of prior squared increment, and the $\mathbf{v_T} = \mathbf{1}$, and the ζ is a smoothing term (a small number) to avoid any division by 0. We use this moving average of second-order moments to make the reverse process concentrate on the current high-frequency information to avoid the image being over-smooth. Compared with the low-frequency information, the high-frequency information, such as the pattern and fine texture of images, is more sensitive to the second-order increment $\mathbf{d}\bar{\mathbf{x}}$. In this assumption, the pacing rate of the high-frequency information will be adaptively magnified, while the pacing rate of the low-frequency information is relatively stable. The adaptive momentum sampler can future balance between high-frequency and low-frequency information.

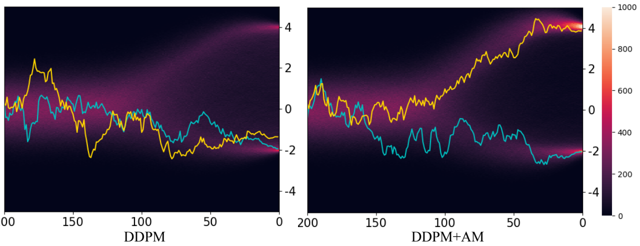


Figure 3: This toy example on one-dimensional data to produce two points $x = -2$ and $x = 4$ in 200 steps. The background is the heat map with 10,000 samples. Additionally, the cyan and gold lines represent the value of the generation sampling process with the original DDPM sampler and our momentum sampler.

Algorithm 1 illustrates the adaptive momentum sampler process. Specifically, we use the momentum increment, $\mathbf{m}_{t-1}$, to replace the current increment of the t-th iteration of noised image. The a and b build a trade-off between the former and current time-dependent score function, demonstrating a strong numerical solver of the reverse process. As shown in Fig. 2, this adjustment can force the reverse sampling process to focus only on the early steps to generate more high-level semantics, such as shapes and outlines, and smooth the low-level information, such as the detailed pattern and texture. The c controls the exponential decay proportion for the second-moment estimates. In order to avoid excessive changes in the magnitude of the momentum increment for each pixel, we amass the square of the $\mathbf{d}\bar{\mathbf{x}}_t$'s Frobenius norm and set the $\mathbf{v}_T = 1$ during the implement. The basic momentum sampler could be considered as a particular case of the adaptive momentum solver when $c = 1$ all the time. If $a = 0$, $b = 1$, and $c = 1$ for all the t-th iterations, the sampling process would degenerate to the original DDPM or DDIM generation process.

3.3 Synthetic Data and Analysis

In this section, we will compare the generation process of the vanilla sampler with our proposed adaptive momentum sampler, both in practice and in theory.

Visualization. To perform a quantitative analysis, we trained a diffusion model to generate 1-dimension toy data with two points ($x = -2$ and $x = 4$) over 200 timesteps. In Fig. 3, we provide a visual representation of the differences between the original DDPM sampler and our proposed adaptive momentum sampler during the sampling process. The heatmaps in the background display the distribution of values for 10,000 samples generated by the original DDPM sampler and our adaptive momentum sampler, respectively. The lighter the color in the background, the greater the number of samples. Specifically, the heat map generated by the adaptive momentum sampler is lighter, especially around the points $x = -2$ and $x = 4$, indicating that it can more precisely produce the value of point. Furthermore, the generation

Algorithm 1 Sampling with the Adaptive Momentum

```
Initialization: $\mathbf{x_T} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \mathbf{m} = \mathbf{0}, \mathbf{v_T} = \mathbf{1}$
for $t = T, \cdots, 1$ do
    $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\epsilon_t = \mathbf{0}$
    Update $\mathbf{d\bar{x}}_t$ based on Equation 6
    $\mathbf{v}_{t-1} = (1 - c) \cdot \mathbf{v_t} + c \cdot \|\mathbf{d\bar{x}}_t\|_{\mathrm{F}}^2$
    $\mathbf{m}_{t-1} = a \cdot \mathbf{m}_t + b \cdot \mathbf{d\bar{x}}_t$
    $\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}} \left( \frac{\mathbf{x}_t}{\sqrt{\alpha_t}} + \frac{\mathbf{m}_{t-1}}{\sqrt{\mathbf{v}_{t-1}} + \zeta} \right)$
end for
return $\mathbf{x}_0$
```

Sampler	CIFAR10		ImageNet		CelebA
	IS	FID	IS	FID	FID
DDIM ($\eta = 0$)	8.16 ± 0.15	3.98	16.40 ± 0.25	19.09	3.40
DDIM + AM	8.16 ± 0.15	$\boxed{3.81}$	16.35 ± 0.28	$\boxed{18.85}$	$\boxed{3.24}$
Analytic-DDIM	8.32 ± 0.09	4.66	16.29 ± 0.30	17.63	3.26
Analytic-DDIM + AM	8.30 ± 0.10	$\boxed{\mathbf{3.50}}$	16.35 ± 0.28	$\boxed{\mathbf{17.51}}$	$\boxed{\mathbf{3.14}}$
DDPM ($\eta = 1$)	8.22 ± 0.09	4.72	17.14 ± 0.20	16.55	5.78
DDPM + AM	8.41 ± 0.09	$\boxed{\mathbf{3.53}}$	17.14 ± 0.21	$\boxed{16.32}$	$\boxed{2.50}$
Analytic-DDPM	8.51 ± 0.09	4.01	17.16 ± 0.21	16.42	5.21
Analytic-DDPM + AM	8.52 ± 0.09	$\boxed{\mathbf{3.53}}$	17.15 ± 0.22	$\boxed{\mathbf{16.19}}$	$\boxed{\mathbf{2.29}}$
DDPM ($\eta = \hat{\eta}$)	8.39 ± 0.15	3.16	17.15 ± 0.18	16.38	3.26
DDPM + AM	8.46 ± 0.08	$\boxed{\mathbf{3.03}}$	17.15 ± 0.19	$\boxed{\mathbf{16.05}}$	$\boxed{\mathbf{3.17}}$

Table 1: The low-resolution image generation results on CIFAR10 (32 × 32), ImageNet (64 × 64) and CelebA (64 × 64) measured in IS ↑ and FID ↓. Note that a same pre-trained diffusion model is utilized on each dataset with different sampling strategies. The improvement of the adaptive momentum sampling ('+AM') is remarkable on most of the datasets compared to the baselines in terms of FID.

trace of the adaptive momentum sampler is more stable, exhibiting less trembling, while the generation trace of the original DDPM sampler (gold line) features significant ineffective vibrations. Notably, for the gold line, the final value generated by the vanilla DDPM sampler is inaccurate. In contrast, the adaptive momentum sampler can more easily escape from narrow valleys, ensuring continuity in the generation process and producing a more accurate final value at last step.

Second-Order Approximation. We conducted a preliminary analysis to compare our method with the original method. As per Song et al. [Song *et al.*, 2020b], the sampling process of DDPM/DDIM uses a first-order approximation of numerical SDEs/ODEs solvers. Building on this conclusion, we demonstrate in Appendix B that our method represents a second-order approximation of SDEs/ODEs. Instead of utilizing the Euler-Maruyama method for numerical solving SDEs/ODEs, we predict the next result by utilizing the midpoint method with previously generated results and the current state. This approach allows us to demonstrate that our method converges to the actual sample of the pre-trained model of DDPM.

In general, the adaptive momentum sampler typically employs historical data to facilitate a smoother generation process, as demonstrated by the gold line in Fig 3 for DDPM+AM. By contrast, the gold line in Fig 3 for DDPM illustrates early time steps characterized by wild swings, which can lead to misguided outcomes and large variance values for the final point.

4 Experiments

In this section, we quantitatively demonstrate the effectiveness of our proposed sampling method. First, we provide the experimental setup. Next, the method is quantitatively evaluated against the baseline samplers. The improvement compared to the baseline will be highlighted on different datasets. Lastly, an ablation study will be offered to inspect the impact of each hyper-parameter. The code is publicly available at *github.com/ShinyGua/DPMs-with-Adam*

4.1 Experimental Setup

Datasets. Following most of the setup in DDPM/DDIM and Latent Diffusion Models (LDM) [Rombach *et al.*, 2022], we utilize CIFAR10 [Krizhevsky *et al.*, 2009] (32 × 32), CelebA [Liu *et al.*, 2018] (64 × 64), ImageNet [Deng *et al.*, 2009] (64 × 64), LSUN [Yu *et al.*, 2015] (256 × 256) and CelebA-HQ [Karras *et al.*, 2017] (256 × 256) in experiments.

Configurations. We mainly employ the original DDIM, DDPM, Analytic-DPM [Bao *et al.*, 2022] and LDM in our comparisons as these schemes are solid and compact baselines. Nonetheless, our proposed schemes can be applied to other improved variants as well. For DDPM/DDIM and Analytic-DPM, the same pre-trained diffusion models are used for the generation. The pre-trained models for CIFAR10, LSUN and CelebA-HQ are collected from the DDPM [Ho *et al.*, 2020], and the pre-trained models for CelebA and ImageNet are collected from DDIM [Song *et al.*, 2020a], and IDDPM [Song and Ermon, 2020] respectively. Besides, we also use the pre-trained models from LDM [Rombach *et al.*, 2022] for high-resolution image generation. The sampling steps are set to 4000 on ImageNet and 1000 on other datasets. For the Eq. 5, we select three η values, the $\eta = 0$ will be equivalent to DDIM, $\eta = 1$ will be a DDPM case in [Song *et al.*, 2020a], and $\eta = \hat{\eta}$ is another DDPM case in its original paper [Ho *et al.*, 2020]. For settings where the sampling step does not equal the original pre-trained model, we follow the same strategy proposed in the DDIM, which replaces the α_t with the corresponding scaled α_τ. All the hyperparameters for the sampling process are presented in Appendix. Our experiments run on one node with 8 NVIDIA A100 GPUs. We use 'AM' to denote sampling with our proposed adaptive momentum.

Measurements. Similar to many other generative models [Goodfellow *et al.*, 2014; Bao *et al.*, 2022; Nichol and Dhariwal, 2021; Ho *et al.*, 2020], we mainly adopt two evaluation metrics which are Frechet Inception Score (FID) and Inception Score (IS) [Lucic *et al.*, 2018; Borji, 2019]. IS is highly correlated with human-annotators [Salimans *et al.*, 2016]. Nevertheless, this measure is often referred to as a method to measure inter-class diversity and is less sensitive to the diversity of the images inside one label [Lucic *et al.*, 2018; Borji, 2019]. In contrast, FID can detect intra-class mode collapsing as well [Lucic *et al.*, 2018]. Thus, FID is considered a

Sampler	CelebA-HQ	Church	Bedroom
DDIM ($\eta = 0$)	10.53	10.84	7.39
DDIM + AM	$\boxed{9.73}$	$\boxed{8.17}$	$\boxed{5.91}$
LDM-DDIM	9.29	3.98	3.87
LDM-DDIM + AM	$\boxed{\mathbf{9.13}}$	$\boxed{\mathbf{3.77}}$	$\boxed{\mathbf{3.54}}$
DDPM ($\eta = 1$)	12.34	7.81	6.24
DDPM + AM	$\boxed{10.97}$	$\boxed{7.74}$	$\boxed{5.54}$
LDM-DDPM	10.79	3.99	3.28
LDM-DDPM + AM	$\boxed{\mathbf{10.47}}$	$\boxed{\mathbf{3.92}}$	$\boxed{\mathbf{3.21}}$

Table 2: The high-resolution image generation results on CelebA-HQ (256 × 256) , Church (256 × 256) and Bedroom (256 × 256) measured in FID ↓. The improvement of the adaptive momentum sampling ('+AM') is remarkable on all of the datasets compared to the baselines in terms of FID.

better measure for image generation tasks. A lower FID value indicates better performance, while a larger IS is better. For each experiment, we draw 50K samples for evaluation.

4.2 Overall Performance

Our proposed sampling scheme is first compared with DDPM and DDIM in Table 1 and 2. The same pre-trained diffusion DDPM/DDIM is used on each dataset to generate the data with vanilla samplers and our adaptive momentum samplers (the '+AM' rows). Analytic-DPM [Bao *et al.*, 2022], and LDM [Rombach *et al.*, 2022] sampler is also used as a baseline to justify our 'AM' on different diffusion schemes.

The results in Table 1 show the improvement of the adaptive momentum sampling over baselines on most low-resolution datasets (lower than 64x64) regarding FID. For the CelebA dataset, the proposed method performs significantly better than DDPM with $\eta = 1$. It not only outperforms the baselines by around 50% but also achieves the state-of-the-art for this dataset which reaches around 2.29 FID. For CIFAR10, although the FID has been very low for the baselines, we still have some rooms to improve from 3.16 to 3.03.

Table 2 shows that our adaptive momentum scheme leads to significant improvements in most baselines when applied to high-resolution datasets such as CelebA-HQ, LSUN Church, and LSUN Bedroom. We also demonstrate the compatibility of our approach with a popular high-resolution model, Latent Diffusion Models, which indicates that our method can be easily incorporated into other DPM-based methods. These results demonstrate the flexibility of our adaptive momentum scheme in improving the performance of various generative models.

Fig. 4 presents a qualitative comparison of synthetic samples generated using different methods. In comparison to the original DDPM sampler, our proposed adaptive momentum sampler produces images with more realistic details and pronounced object outlines, confirming its ability to balance high-level and low-level information more effectively. For instance, our method enhances image details while preserving original patterns, such as the recovered person structure in row 2 column 4 and the enhanced facial details (forehead hair) in the last picture of the 2nd row.

	Sampling Steps			
Sampler	25	50	100	1000
DDIM ($\eta = 0$)	6.24	4.77	**4.25**	3.98
Analytic-DDIM	9.96	6.02	4.88	4.66
DDIM + AM	**6.15**	**4.76**	4.38	**3.81**
DDPM ($\eta = 1$)	14.44	8.23	5.82	4.72
Analytic-DDPM	8.50	5.50	4.45	4.31
DDPM + AM	**7.19**	**4.10**	**3.61**	**3.53**

Table 3: This table displays the FID ↓ scores for generation results obtained using varying numbers of sampling steps.

	Sampling Steps			
Sampler	25	50	100	1000
DDPM ($\eta = 1$)	14.44	8.23	5.82	4.72
DDPM + AM ($b = 0.05$)	14.43	8.13	5.71	4.68
DDPM + AM ($b = 0.1$)	10.13	6.52	5.88	4.09
DDPM + AM ($b = 0.15$)	$\boxed{7.25}$	$\boxed{4.41}$	$\boxed{3.65}$	$\boxed{3.54}$
DDPM + AM ($b = 0.2$)	20.26	5.07	4.04	3.98
DDPM + AM ($c = 0.001$)	7.23	4.39	3.64	3.54
DDPM + AM ($c = 0.005$)	7.22	4.35	3.62	$\boxed{\mathbf{3.53}}$
DDPM + AM ($c = 0.01$)	$\boxed{\mathbf{7.19}}$	4.33	$\boxed{\mathbf{3.61}}$	$\boxed{\mathbf{3.53}}$
DDPM + AM ($c = 0.1$)	7.61	$\boxed{\mathbf{4.10}}$	3.67	3.85

Table 4: This table presents the results of a hyperparameter experiment measuring FID ↓ for our method. The best results among different values of b and c are marked in boxes, and the best results of each full column are shown in bold.

4.3 Ablation Study

This subsection investigates different aspects of the settings, including different time steps and hyperparameters' effects.

Different Sampling Steps. Table 3 illustrates the performance of our method when fewer sample steps are used for acceleration on CIFAR10. The proposed method could perform better or be comparable when using different sampling steps. On most of the settings, we can provide up to 50% performance enhancements over the original DDPM sampler. It is observed that the improvement is significant on the larger number of sample steps while less evident on the smaller ones. This makes sense, possibly because using a small number of sampling steps is not always sufficient for accurate momentum estimation and noise suppression.

Hyper-Parameters. Considering about the effects of b and c in the Eq. 8 and 9, we setup with different values as in Table 4 on CIFAR10. The first five lines illustrate how the FID of the generated images changes as the b increases when $c = 0$. In general, the image quality will first increase and then decrease. When the $b = 0.15$, the FID achieves the lowest value for image generation in different sampling steps, which is the best trade-off point. The last four lines show how the b influence the sampling results when $b = 0.15$. Similarly, the FID first decreases and then increases when the b decrease. The last four rows reflect the impact of c with $b = 0.15$, where the most of best performance is obtained at $c = 0.01$. Empirically, different denoising steps can share the

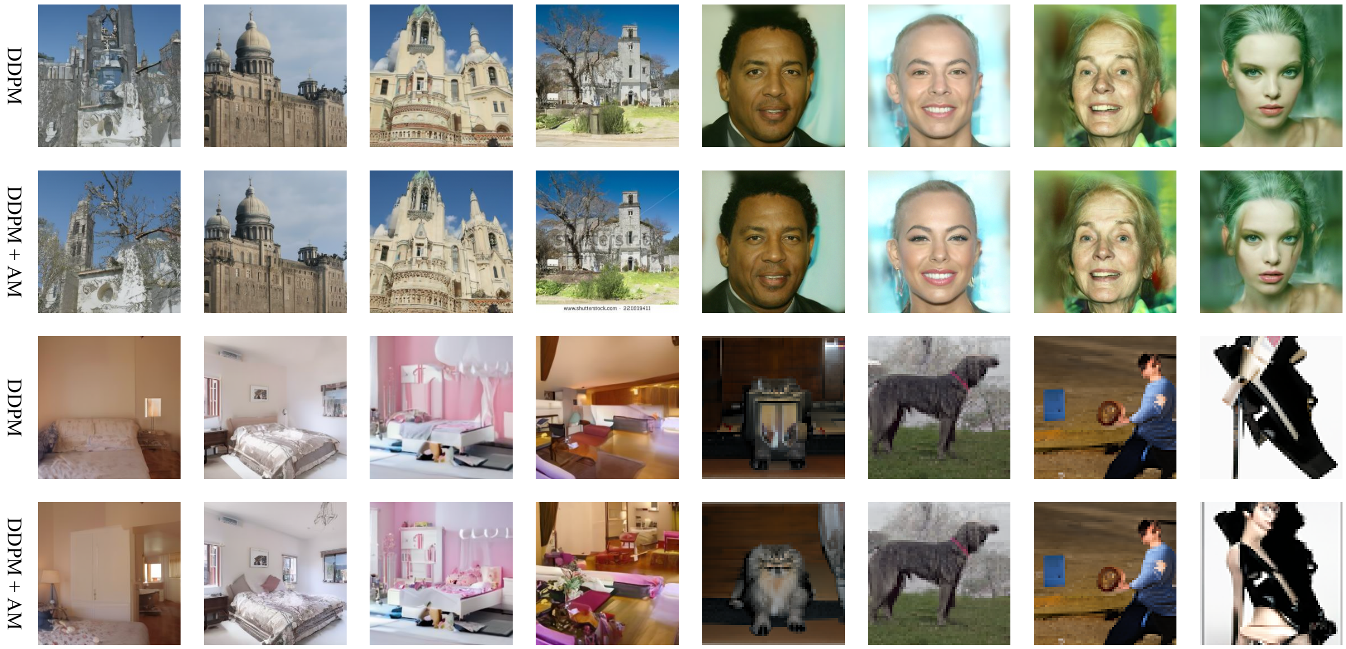


Figure 4: Generated samples for LSUN Church, LSUN Bedroom, CelebA-HQ and ImageNet

same hyper-parameters as shown in Table 3 and Appendix. Thus, we only need to find $b_{\max}$ like $10^{-3}, 10^{-2}$ or 10^{-1} in the small denoising steps, e.g., 100 or 200. Most of hyperparameter values are shared across different datasets.

Adaptivity of the Momentum. Table 4 also presents the results of an ablation study on our proposed method, the Adaptive Momentum Sampler. The first five rows display the outcomes when the sampling process only utilizes the moving average of the moment increment. Moreover, the comparison between the first and fourth rows reveals that our proposed method outperforms the original reverse process sampler, underscoring the efficacy of the momentum idea. Furthermore, comparing the fourth and eighth rows, we observe that introducing adaptivity through the moving average of second-order moments further enhances the performance of our adaptive momentum sampler, which demonstrates its excellent generalization ability. Thus, both non-adaptive and adaptive momentum samplers prove advantageous for DPMs.

4.4 Rate-Distortion Trade-Off

The rate-distortion curves of a trained model using both adaptive momentum inference and baseline inference schemes are plotted in Fig. 5. It was criticized in the literature that diffusion models tend to spend much more rates, *i.i.*, model capacity, on restoring imperceptible distortions than high-level semantics [Ho *et al.*, 2020; Rombach *et al.*, 2022], which is also observed in Fig. 5 for the baseline scheme. In contrast, the newly adaptive proposed momentum sampling is able to straighten the curve and thus strike a better balance between generating high-level semantics and low-level details. This could provide a possible explanation for the improved generation performance observed in our experiments. Note that it is reasonable for the red curve to have larger rates, which are computed by the cumulative sum of the variational bound terms [Ho *et al.*, 2020], since the momentum inference is being applied to a model pre-trained without momentum. We expect that incorporating an adaptive momentum strategy into the training process could further enhance the rate-distortion trade-off as well as the generation quality in future works.

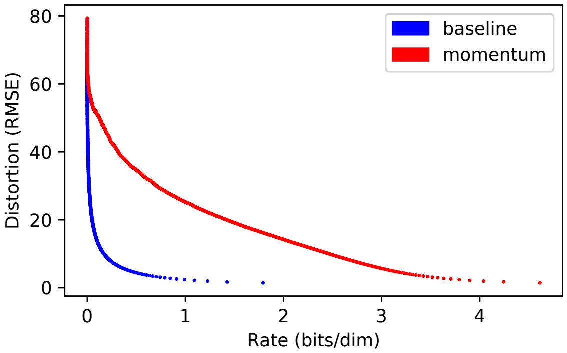


Figure 5: The rate-distortion trade-off on CIFAR-10. The proposed momentum scheme could achieve a better balance between high-level semantics and low-level details.

5 Conclusion

In this work, we propose a training-free sampler to improve the generated images of Diffusion Generative Models, especially for discrete discrete time steps settings. Through extensive experiments, we found out that the adaptive momentum sampler use the history the denoising trajectory to improve the quality of the generated images. In future works, we will expand the scheme to continuous settings as well as with solid theoretical versions.

Acknowledgements

This work was supported in part by the Australian Research Council under Projects DP210101859 and FT230100549. The authors acknowledge the use of the National Computational Infrastructure (NCI) which is supported by the Australian Government and accessed through the NCI AI Flagship Scheme and Sydney Informatics Hub HPC Allocation Scheme.

References

[Bao *et al.*, 2022] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022.

[Borji, 2019] Ali Borji. Pros and Cons of GAN evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.

[Brock *et al.*, 2018] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[Bruce *et al.*, 2024] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. *arXiv preprint arXiv:2402.15391*, 2024.

[Deng *et al.*, 2009] 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. Ieee, 2009.

[Dhariwal and Nichol, 2021] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[Dinh *et al.*, 2023] Anh-Dung Dinh, Daochang Liu, and Chang Xu. Pixelasparam: A gradient view on diffusion sampling with guidance. In *International Conference on Machine Learning*, pages 8120–8137. PMLR, 2023.

[Dinh *et al.*, 2024] Anh-Dung Dinh, Daochang Liu, and Chang Xu. Rethinking conditional diffusion sampling with progressive guidance. *Advances in Neural Information Processing Systems*, 36, 2024.

[Dung and Binh, 2022] Dinh Anh Dung and Huynh Thi Thanh Binh. Gdegan: Graphical discriminative embedding gan for tabular data. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–11. IEEE, 2022.

[Gong *et al.*, 2019] Mingming Gong, Yanwu Xu, Chunyuan Li, Kun Zhang, and Kayhan Batmanghelich. Twin auxilary classifiers gan. *Advances in neural information processing systems*, 32, 2019.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[Guo *et al.*, 2020] Tianyu Guo, Chang Xu, Jiajun Huang, Yunhe Wang, Boxin Shi, Chao Xu, and Dacheng Tao. On positive-unlabeled classification in gan. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 8385–8393, 2020.

[Hinton and Salakhutdinov, 2006] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[Karras *et al.*, 2017] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[Karras *et al.*, 2020] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[Karras *et al.*, 2022] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Liu *et al.*, 2018] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018.

[Liu *et al.*, 2023] Daochang Liu, Qiyue Li, Anh-Dung Dinh, Tingting Jiang, Mubarak Shah, and Chang Xu. Diffusion action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10139–10149, 2023.

[Lucic *et al.*, 2018] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs created equal? a large-scale study. *Advances in neural information processing systems*, 31, 2018.

[Mukkamala and Hein, 2017] Mahesh Chandra Mukkamala and Matthias Hein. Variants of RMSProp and Adagrad with logarithmic regret bounds. In *International conference on machine learning*, pages 2545–2553. PMLR, 2017.

[Nichol and Dhariwal, 2021] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[Odena *et al.*, 2017] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

[Rombach *et al.*, 2022] 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.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[Salimans *et al.*, 2016] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in neural information processing systems*, 29, 2016.

[Shao *et al.*, 2021] Huajie Shao, Zhisheng Xiao, Shuochao Yao, Dachun Sun, Aston Zhang, Shengzhong Liu, Tianshi Wang, Jinyang Li, and Tarek Abdelzaher. ControlVAE: Tuning, analytical properties, and performance analysis. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[Song and Ermon, 2019] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[Song and Ermon, 2020] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[Song *et al.*, 2020a] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[Song *et al.*, 2020b] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[Wang *et al.*, 2023] Yunke Wang, Xiyu Wang, Anh-Dung Dinh, Bo Du, and Charles Xu. Learning to schedule in diffusion probabilistic models. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2478–2488, 2023.

[Wu *et al.*, 2023] Zike Wu, Pan Zhou, Kenji Kawaguchi, and Hanwang Zhang. Momentum-accelerated diffusion process for faster training and sampling. 2023.

[Yu *et al.*, 2015] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[Yu *et al.*, 2022] Yuncong Yu, Dylan Kruyff, Jiao Jiao, Tim Becker, and Michael Behrisch. Pseudo: Interactive pattern search in multivariate time series with locality-sensitive hashing and relevance feedback. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):33–42, 2022.