ELUCIDATING THE EXPOSURE BIAS IN DIFFUSION MODELS

Ning, M., M. Li, J. Su, A.A. Salah, I. Onal Ertugrul.

ABSTRACT

Diffusion models have demonstrated impressive generative capabilities, but their *exposure bias* problem, described as the input mismatch between training and sampling, lacks in-depth exploration. In this paper, we systematically investigate the exposure bias problem in diffusion models by first analytically modelling the sampling distribution, based on which we then attribute the prediction error at each sampling step as the root cause of the exposure bias issue. Furthermore, we discuss potential solutions to this issue and propose an intuitive metric for it. Along with the elucidation of exposure bias, we propose a simple, yet effective, training-free method called Epsilon Scaling to alleviate the exposure bias. We show that Epsilon Scaling explicitly moves the sampling trajectory closer to the vector field learned in the training phase by scaling down the network output (Epsilon), mitigating the input mismatch between training and sampling. Experiments on various diffusion frameworks (ADM, DDPM/DDIM, EDM, LDM), unconditional and conditional settings, and deterministic vs. stochastic sampling verify the effectiveness of our method. Remarkably, our ADM-ES, as a SOTA stochastic sampler, obtains 2.17 FID on CIFAR-10 under 100-step unconditional generation.

1 INTRODUCTION

Due to the outstanding generation quality and diversity, diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019) have achieved unprecedented success in image generation (Dhariwal & Nichol, 2021; Nichol et al., 2022; Rombach et al., 2022; Saharia et al., 2022), audio synthesis (Kong et al., 2021; Chen et al., 2021) and video generation (Ho et al., 2022). Unlike generative adversarial networks (GANs) (Goodfellow et al., 2014), variational autoencoders (VAEs) (Kingma & Welling, 2014) and flow-based models (Dinh et al., 2014; 2017), diffusion models stably learn the data distribution through a noise/score prediction objective and progressively removes noise from random initial vectors in the iterative sampling stage.

A key feature of diffusion models is that good sample quality requires a long iterative sampling chain since the Gaussian assumption of reverse diffusion only holds for small step sizes (Xiao et al., 2022). However, Ning et al. (2023) claim that the iterative sampling chain also leads to the *exposure bias* problem (Ranzato et al., 2016; Schmidt, 2019). Concretely, given the noise prediction network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$, exposure bias refers to the input mismatch between training and inference, where the former is always exposed to the ground truth training sample $\boldsymbol{x}_t$ while the latter depends on the previously generated sample $\hat{\boldsymbol{x}}_t$. The difference between $\boldsymbol{x}_t$ and $\hat{\boldsymbol{x}}_t$ causes the discrepancy between $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t)$ and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t)$, which leads to the error accumulation and the sampling drift (Li et al., 2023a).

We point out that the exposure bias problem in diffusion models lacks in-depth exploration. For example, there is no proper metric to quantify the exposure bias and no explicit error analysis for it. To shed light on exposure bias, we conduct a systematical investigation in this paper by first modelling the sampling distribution with prediction error. Based on our analysis, we find that the practical sampling distribution has a variance larger than the ground truth distribution at every single step, demonstrating the analytic difference between $\boldsymbol{x}_t$ in training and $\hat{\boldsymbol{x}}_t$ in sampling. Along with the sampling distribution analysis, we propose a metric δ_t to evaluate exposure bias by comparing the variance difference between training and sampling. Finally, we discuss potential solutions to exposure bias, and propose a simple yet effective *training-free and plug-in* method called Epsilon Scaling to alleviate this issue.

We test our approach on four different diffusion frameworks using deterministic and stochastic sampling, and on conditional and unconditional generation tasks. Without affecting the recall and precision (Kynkäänniemi et al., 2019), our method yields dramatic Fréchet Inception Distance (FID) (Heusel et al., 2017) improvements. Also, we illustrate that Epsilon Scaling effectively reduces the exposure bias by moving the sampling trajectory towards the training trajectory. Overall, our contributions to diffusion models are:

- We systematically investigate the exposure bias problem and propose a metric for it.
- We suggest potential solutions to the exposure bias issue and propose a training-free, plug-in method (Epsilon Scaling) which significantly improves the sample quality.
- Our extensive experiments demonstrate the generality of Epsilon Scaling and its wide applicability to different diffusion architectures.

2 Related Work

Diffusion models were introduced by Sohl-Dickstein et al. (2015) and later improved by Song & Ermon (2019), Ho et al. (2020) and Nichol & Dhariwal (2021). Song et al. (2021b) unify score-based models and Denoising Diffusion Probabilistic Models (DDPMs) via stochastic differential equations. Furthermore, Karras et al. (2022) disentangle the design space of diffusion models and introduce the EDM model to further boost the performance in image generation. With the advances in diffusion theory, conditional generation (Ho & Salimans, 2022; Choi et al., 2021) also flourishes in various scenarios, including text-to-image generation (Nichol et al., 2022; Ramesh et al., 2022; Rombach et al., 2022; Saharia et al., 2022), controllable image synthesis (Zhang & Agrawala, 2023; Li et al., 2023b; Zheng et al., 2023), as well as generating other modalities, for instance, audio (Chen et al., 2021; Kong et al., 2021), object shape (Zhou et al., 2021) and time series (Rasul et al., 2021). In the meantime, accelerating the time-consuming reverse diffusion sampling has been extensively investigated in many works (Song et al., 2021a; Lu et al., 2022; Liu et al., 2022). For example, distillation (Salimans & Ho, 2022), Restart sampler (Xu et al., 2023) and fast ODE samplers (Zhao et al., 2023) have been proposed to speed up the sampling.

The phenomenon of exposure bias within the diffusion model was first identified by Ning et al. (2023). They introduced an extra noise at each step during the training to mitigate the discrepancy between training and inference, thereby reducing the impact of exposure bias. Additionally, another approach presented by Li et al. (2023a) sought to address exposure bias without necessitating retraining of the model. Their method involved a manipulation of the time step during the backward generation process. However, the exposure bias in diffusion models still lacks illuminating research in terms of the explicit sampling distribution, metric and root cause, which is the objective of this paper. Besides, we propose a solution called Epsilon Scaling in the sampling phase based on the observation of the prediction deviation between training and sampling. This method, while straightforward, proves effective in mitigating the exposure bias issue.

3 Exposure Bias in Diffusion Models

3.1 Background

We first briefly review DDPMs (Ho et al., 2020). Given a sample $\boldsymbol{x}_0$ from the data distribution $q(\boldsymbol{x}_0)$ and a well-behaved noise schedule $(\beta_1, ..., \beta_T)$, DDPM defines the forward process as a Markov chain $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and iteratively adds Gaussian noise by $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I})$ until obtaining the prior $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The Gaussian forward process allows us to sample $\boldsymbol{x}_t$ directly conditioned on the input $\boldsymbol{x}_0$:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}), \qquad \boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{1}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, $\alpha_t = 1 - \beta_t$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Then, the reverse distribution $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is approximated by a neural network, from which we can sample $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and iteratively run the reverse process $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \sigma_t \boldsymbol{I})$ to get a sample from $q(\boldsymbol{x}_0)$. The optimisation objective is $D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)))$ in which the ground truth forward process posterior $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ is tractable when conditioned on $\boldsymbol{x}_0$ using Bayes theorem:

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t \boldsymbol{I}) \tag{2}$$

$$\tilde{\boldsymbol{\mu}}(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t \tag{3}$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{4}$$

Regarding parameterising $\mu_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$, Ho et al. (2020) found that using a neural network to predict $\boldsymbol{\epsilon}$ (Eq. 6) worked better than predicting $\boldsymbol{x}_0$ (Eq. 5) in practice:

$$\mu_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t \tag{5}$$

$$= \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)), \tag{6}$$

where $\boldsymbol{x}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ denotes the denoising model which predicts $\boldsymbol{x}_0$ given $\boldsymbol{x}_t$. *For simplicity, we use $\boldsymbol{x}_{\boldsymbol{\theta}}^t$ as the short notation of $\boldsymbol{x}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$* in the rest of this paper.

3.2 Sampling Distribution with Prediction Error

The exposure bias problem of diffusion models is described as the input mismatch between $\boldsymbol{x}_t$ during training and $\hat{\boldsymbol{x}}_t$ during sampling (Ning et al., 2023). In this section, we explicitly derive the sampling distribution $q(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1}, \boldsymbol{x}_{\boldsymbol{\theta}}^{t+1})$ and compare it with the training distribution $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$, where the former shows the $\hat{\boldsymbol{x}}_t$ seen by the network in the sampling phase, while the latter is the $\boldsymbol{x}_t$ seen by the network during training at timestep t.

Comparing Eq. 3 with Eq. 5, Song et al. (2021a) emphasise that the sampling distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is in fact parameterised as $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_{\boldsymbol{\theta}}^t)$ where $\boldsymbol{x}_{\boldsymbol{\theta}}^t$ means the predicted $\boldsymbol{x}_0$ given $\boldsymbol{x}_t$. Therefore, the practical sampling paradigm is that we first predict $\boldsymbol{\epsilon}$ using $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$. Then we derive the estimation $\boldsymbol{x}_{\boldsymbol{\theta}}^t$ for $\boldsymbol{x}_0$ using Eq. 1. Finally, based on the ground truth posterior $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$, $\boldsymbol{x}_{t-1}$ is generated using $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_{\boldsymbol{\theta}}^t)$ by replacing $\boldsymbol{x}_0$ with $\boldsymbol{x}_{\boldsymbol{\theta}}^t$. Since $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_{\boldsymbol{\theta}}^t) = q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ holds only if $\boldsymbol{x}_{\boldsymbol{\theta}}^t = \boldsymbol{x}_0$, this requires the network to make no prediction error about $\boldsymbol{x}_0$ to ensure $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_{\boldsymbol{\theta}}^t)$ share the same variance with $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$. However, $\boldsymbol{x}_{\boldsymbol{\theta}}^t - \boldsymbol{x}_0$ is practically non-zero and we claim that the prediction error of $\boldsymbol{x}_0$ needs to be considered to derive the real sampling distribution. Following Analytic-DPM (Bao et al., 2022b) and Bao et al. (2022a), we model $\boldsymbol{x}_{\boldsymbol{\theta}}^t$ as $p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_t)$ and approximate it by a Gaussian distribution:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{\boldsymbol{\theta}}^t; \boldsymbol{x}_0, e_t^2\boldsymbol{I}), \qquad \boldsymbol{x}_{\boldsymbol{\theta}}^t = \boldsymbol{x}_0 + e_t\boldsymbol{\epsilon}_0 \quad (\boldsymbol{\epsilon}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})) \tag{7}$$

Taking the prediction error into account, we now compute $q(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1}, \boldsymbol{x}_{\boldsymbol{\theta}}^{t+1})$, which is the same distribution as $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_{\boldsymbol{\theta}}^t)$, by substituting with the index $t+1$ and using $\hat{\boldsymbol{x}}_t$ to highlight that it is generated in the sampling stage. Based on Eq. 2, we know $q(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1}, \boldsymbol{x}_{\boldsymbol{\theta}}^{t+1}) = \mathcal{N}(\hat{\boldsymbol{x}}_t; \mu_{\boldsymbol{\theta}}(\boldsymbol{x}_{t+1}, t+1), \tilde{\beta}_{t+1}\boldsymbol{I})$. Its mean and variance can be further derived according to Eq. 5 and Eq. 4, respectively. Thus, a sample from the distribution is $\hat{\boldsymbol{x}}_t = \mu_{\boldsymbol{\theta}}(\boldsymbol{x}_{t+1}, t+1) + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1$, namely:

$$\hat{\boldsymbol{x}}_t = \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}\boldsymbol{x}_{\boldsymbol{\theta}}^{t+1} + \frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\boldsymbol{x}_{t+1} + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1 \tag{8}$$

Plugging Eq. 7 and Eq. 1 (using index $t+1$) into Eq. 8, we obtain the final analytical form of $\hat{\boldsymbol{x}}_t$ (see Appendix A.1 for the full derivation):

$$\hat{\boldsymbol{x}}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t + (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2}\boldsymbol{\epsilon}_3 \tag{9}$$

where, $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_3 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. From Eq. 9, we obtain the mean and variance of $q(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1}, \boldsymbol{x}_{\boldsymbol{\theta}}^{t+1})$ and compare them with the parameters of $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$. *For simplicity, we denote $q(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1}, \boldsymbol{x}_{\boldsymbol{\theta}}^{t+1})$ as $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ from now on.* In Table 1, $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ shows the $\boldsymbol{x}_t$ seen by the network during training while $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ indicates the $\hat{\boldsymbol{x}}_t$ exposed to the network during sampling. Note that, the method of solving out $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ can be generalised to $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\boldsymbol{x}_{t+1})$ and $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_T)$ (see Appendix A.2). In the same spirit of modelling $\boldsymbol{x}_{\boldsymbol{\theta}}^t$ as a Gaussian, we also derived the sampling distribution $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ for DDIM (Song et al., 2021a) in Appendix A.3.

Table 1: The distribution $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ during training and $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ during DDPM sampling.

	Mean	Variance
$q(\boldsymbol{x}_t\|\boldsymbol{x}_0)$	$\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$	$(1-\bar{\alpha}_t)\boldsymbol{I}$
$q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t\|\boldsymbol{x}_{t+1})$	$\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$	$(1-\bar{\alpha}_t+(\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2)\boldsymbol{I}$

3.3 Exposure Bias Due to Prediction Error

It is clear from Table 1 that the variance of the sampling distribution $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ is always larger than the variance of the training distribution $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ by the magnitude $(\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2$. Note that, this variance gap between training and sampling is produced just in a single reverse diffusion step, given that the network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ can get access to the ground truth input $\boldsymbol{x}_{t+1}$. What makes the situation worse is that the error of single-step sampling accumulates in the multi-step sampling, resulting in an explosion of sampling variance error. For instance, $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\boldsymbol{x}_{t+1})$ (in Appendix A.2) shows the variance error in two consecutive sampling steps when compared with $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)$. On CIFAR-10 (Krizhevsky et al., 2009), we designed an experiment to statistically measure both the single-step variance error of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ and multi-step variance error of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_T)$ using 20-step sampling (see Appendix A.4). The results in Fig. 1 indicate that the closer to $t=1$ (the end of sampling), the larger the variance error of multi-step sampling. The explosion of sampling variance error results in the sampling drift (exposure bias) problem and we attribute the prediction error $\boldsymbol{x}_{\boldsymbol{\theta}}^t-\boldsymbol{x}_0$ as the root cause of the exposure bias in diffusion models.

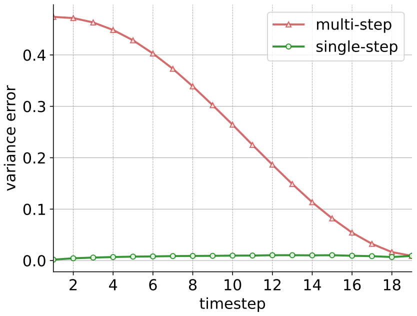


Figure 1: Variance error in single-step and multi-step samplings.

Intuitively, a possible solution to exposure bias is using a sampling noise variance $\beta^{'}$, which is smaller than $\tilde{\beta}_t$, to counteract the extra variance term $(\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2$ caused by the prediction error $\boldsymbol{x}_{\boldsymbol{\theta}}^t-\boldsymbol{x}_0$. Unfortunately, $\tilde{\beta}_t$ is the lower bound of the sampling noise schedule $\dot{\beta}_t\in[\tilde{\beta}_t,\beta_t]$, where the lower bound and upper bound are the sampling variances given by $q(\boldsymbol{x}_0)$ being a delta function and isotropic Gaussian function, respectively (Nichol & Dhariwal, 2021). Therefore, we can draw a conclusion that the exposure bias problem can not be alleviated by manipulating the sampling noise schedule $\dot{\beta}_t$.

Interestingly, Bao et al. (2022b) analytically provide the optimal sampling noise schedule $\beta_t^{\star}$ which is larger than the lower bound $\tilde{\beta}_t$. Based on what we discussed earlier, $\beta_t^{\star}$ would cause a more severe exposure bias issue than $\tilde{\beta}_t$. A strange phenomenon, but not explained by Bao et al. (2022b) is that $\beta_t^{\star}$ leads to a worse FID than using $\tilde{\beta}_t$ under 1000 sampling steps. We believe the exposure bias is in the position to account for this phenomenon: under the long sampling, the negative impact of exposure bias exceeds the positive gain of the optimal variance $\beta_t^{\star}$.

3.4 Metric for Exposure Bias

Although some literature has already discussed the exposure bias problem in diffusion models (Ning et al., 2023; Li et al., 2023a), there still lacks a well-defined and straightforward metric for this concept. We propose to use the variance error of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_T)$ to quantify the exposure bias at timestep t under multi-step sampling. Specifically, our metric δ_t for exposure bias is defined as $\delta_t=(\sqrt{\hat{\beta}_t}-\sqrt{\bar{\beta}_t})^2$, where $\bar{\beta}_t=1-\bar{\alpha}_t$ denotes the variance of $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ during training and $\hat{\beta}_t$ presents the variance of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_T)$ in the regular sampling process. The metric δ_t is inspired by the Fréchet distance (Dowson & Landau, 1982) between $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ and $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_T)$, which is $d^2=N(\sqrt{\hat{\beta}_t}-\sqrt{\bar{\beta}_t})^2$ where N is the dimensions of $\boldsymbol{x}_t$. In Appendix A.6, we empirically find that δ_t exhibits a strong correlation with FID given a trained model. Our method of measuring δ_t is described in Algorithm 3 (see Appendix A.5). The key step of Algorithm 3 is that we subtract the mean $\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0$ and the remaining term $\hat{\boldsymbol{x}}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0$ corresponds to the stochastic term of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\boldsymbol{x}_T)$.

3.5 Solution Discussion

We now discuss possible solutions to the exposure bias issue of diffusion models based on the analysis throughout Section 3. Recall that the prediction error of $\boldsymbol{x}_{\boldsymbol{\theta}}^t - \boldsymbol{x}_0$ is the root cause of exposure bias. Thus, the most straightforward way of reducing exposure bias is learning an accurate $\boldsymbol{\epsilon}$ or score function (Song & Ermon, 2019) prediction network. For example, by delicately designing the network and hyper-parameter tuning, EDM (Karras et al., 2022) improves the FID from 3.01 to 2.51 on CIFAR-10 dataset, presenting a significant improvement. Secondly, we believe that data augmentation can reduce the risk of learning inaccurate $\boldsymbol{\epsilon}$ or score function for $\hat{\boldsymbol{x}}_t$ by learning a denser vector field than vanilla diffusion models. For instance, Karras et al. (2022) has shown that the geometric augmentation (Karras et al., 2020) benefits the network training and sample quality. In the same spirit, DDPM-IP (Ning et al., 2023) augments each training sample $\boldsymbol{x}_t$ by a Gaussian term and achieves substantial improvements in FID.

It is worth pointing out that the above-mentioned methods require retraining the network and expensive parameter searching during the training. This naturally drives us to the question: *Can we alleviate the exposure bias in the sampling stage, without any retraining*?

4 Method

4.1 Epsilon Scaling

In Section 3.3, we have concluded that the exposure bias issue can not be solved by reducing the sampling noise variance, thus another direction to be explored in the sampling phase is the prediction of the network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$. Since we already know from Table 1 that $\boldsymbol{x}_t$ inputted to the network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ in training differs from $\hat{\boldsymbol{x}}_t$ fed into the network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ in sampling, we are interested in understanding the difference in the output of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ between training and inference.

For simplicity, we denote the output of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ as $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^t$ in training and as $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s$ in sampling. Although the ground truth of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s$ is not accessible during inference, we are still able to speculate the behaviour of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s$ from the L2-norm perspective. In Fig. 2, we plot the L2-norm of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^t$ and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s$ at each timestep. In detail, given a trained, frozen model and ground truth $\boldsymbol{x}_t$, $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^t$ is collected by $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^t = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$. In this way, we simulate the training stage and analyse its $\boldsymbol{\epsilon}$ prediction. In contrast, $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s$ is gathered in the real sampling process, namely $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t, t)$. It is clear from Fig. 2 that the L2-norm of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s$ is always larger than that of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^t$. Since $\hat{\boldsymbol{x}}_t$ lies around $\boldsymbol{x}_t$ with a larger variance (Section 3.2), we can infer the network learns an inaccurate vector field $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t, t)$ for each $\hat{\boldsymbol{x}}_t$ in the vicinity of $\boldsymbol{x}_t$ with the vector length longer than that of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$.

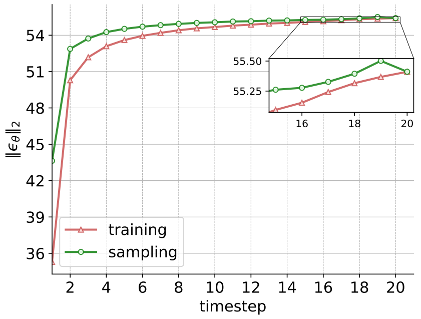


Figure 2: $\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)\|_2$ during training and sampling on CIFAR-10. We use 20-step sampling and report the L2-norm using 50k samples at each timestep.

One can infer that the prediction $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s$ could be improved if we can move the input $(\hat{\boldsymbol{x}}_t, t)$ from the inaccurate vector field (green curve in Fig. 2) towards the reliable vector field (red curve in Fig. 2). To this end, we propose to scale down $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s$ by a factor λ_t at sampling timestep t. Our solution is based on the observation: $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^t$ and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s$ share the same input $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ at timestep $t = T$, but from timestep $T-1$, $\hat{\boldsymbol{x}}_t$ (input of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s$) starts to diverge from $\boldsymbol{x}_t$ (input of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^t$) due to the $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ error made at previous timestep. This iterative process continues along the sampling chain and results in exposure bias. Therefore, we can push $\hat{\boldsymbol{x}}_t$ closer to $\boldsymbol{x}_t$ by scaling down the over-predicted magnitude of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s$. According to the regular sampling (Eq. 6), our sampling method only differs in the λ_t term and is expressed as $\mu_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\lambda_t})$. Note that, our Epsilon Scaling serving as a plug-in method adds no computational load to the original sampling of diffusion models.

4.2 The Design of Scaling Schedule

Intuitively, the scaling schedule λ_t should be directly decided by the L2-norm quotient $\left\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^s\right\|_2 / \left\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^t\right\|_2$, denoted as $\Delta N(t)$, at each timestep t. However, we emphasise that $\Delta N(t)$ reflects the accumulated effect of the biased prediction error made at each timestep $T, T-1, ..., t$. Suppose the L2-norm of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}^t$ to be scaled at timestep t is λ_t, then $\Delta N(t)$ satisfy $\Delta N(t) = \int_t^T \lambda_t dt$ given a linearly distributed vector field learned by $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ in the vicinity of $\boldsymbol{x}_t$: $\|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_2 = c(\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t)\|_2 - \|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t)\|_2)$, where c is a constant. In practice, $\Delta N(t) \approx \int_t^T \lambda_t dt$ holds for a non-overfitting network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$.

As shown by Nichol & Dhariwal (2021) and Benny & Wolf (2022), the $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ predictions near $t = 0$ are very bad, with the loss larger than other timesteps by several orders of magnitude. Thereby, we can ignore the area close to $t = 0$ to design λ_t, because scaling a problematic $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ does not lead to a better predicted $\boldsymbol{\epsilon}$. We plot the $\Delta N(t)$ curve in the cases of 20-step and 50-step sampling on CIFAR-10 in Fig. 3. It shows that $\Delta N(t)$ can be fitted by a quadratic function in the interval $t \sim (5, T)$. Thereby, the integrand λ_t is a linear function $\lambda_t = kt + b$ where k, b are constants. Another observation from Fig. 3 is that $\Delta N(t)$ under 50-step sampling has a smaller curvature than 20-step sampling. This tendency applies to a larger sampling step, for example, 100-step sampling presents a flatter curvature than 50-step sampling in our experiments. Overall, the design principle for λ_t is that the longer the sampling step, the smaller k we should use. In Section 5.1, we will see that k is practically a small number and would decay to 0 around 50 sampling steps.

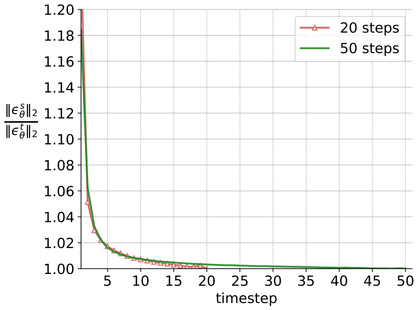


Figure 3: $\Delta N(t)$ at each timestep.

5 Results

In this section, we evaluate the performance of Epsilon Scaling using FID (Heusel et al., 2017). To demonstrate that Epsilon Scaling is a generic solution to exposure bias, we test the approach on various diffusion frameworks, samplers and conditional settings. Following the fast sampling paradigm (Karras et al., 2022) in the diffusion community, we apply the sampling steps T' less than the training diffusion step T and we focus on $T' \leqslant 100$ for practical usages. Our FID computation is consistent with (Dhariwal & Nichol, 2021) for equal comparison. All FIDs are reported using 50k generated samples and the full training set as the reference batch, except for the LSUN dataset where we follow (Dhariwal & Nichol, 2021) and use 50k training samples as the reference batch. Lastly, Epsilon Scaling does not affect the precision and recall, and we report these results in Appendix A.7.

5.1 Main Results on ADM

Since Epsilon Scaling is a training-free method, we utilise the pre-trained ADM model as the baseline and compare it against our ADM-ES (ADM with Epsilon Scaling) on datasets CIFAR-10 (Krizhevsky et al., 2009), LSUN tower (Yu et al., 2015) and FFHQ (Karras et al., 2019) for unconditional generation and on datasets ImageNet 64×64 and ImageNet 128×128 (Chrabaszcz et al., 2017) for class-conditional generation. We employ the respacing sampling technique (Nichol & Dhariwal, 2021) to enable fast stochastic sampling.

Table 2 shows that independently of the dataset and the number of sampling steps T', our ADM-ES outperforms ADM by a large margin in terms of FID, indicating the remarkable effectiveness of Epsilon Scaling. For instance, on FFHQ 128×128, ADM-ES exhibits less than half the FID of ADM, with 7.75, 16.65 and 34.52 FID improvement under 100, 50 and 20 sampling steps, respectively. Moreover, when compared with the previous best stochastic samplers, ADM-ES outperforms EDM (Karras et al., 2022), Improved SDE (Karras et al., 2022), Restart Sampler (Xu et al., 2023) and SA-Solver (Xue et al., 2023), exhibiting state-of-the-art stochastic sampler (SDE solver). For example, ADM-ES not only achieves a better FID (2.17) than EDM and Improved SDE, but also accelerates the sampling speed by 5-fold to 20-fold (see Table 3). Even under 50-step sampling, Epsilon Scaling surpasses SA-Solver and obtains competitive FID against other samplers.

Note that, ADM-ES refers to the uniform schedule $\lambda_t = b$ and ADM-ES* applies the linear schedule $\lambda_t = kt + b$ in Table 2. In our experiments, we find that the slope k is approaching 0 as the sampling step T' increases. Taking CIFAR-10 as an example, ADM-ES* gains 0.84 FID improvement over

Table 2: FID on ADM baseline. We compare ADM with our ADM-ES (uniform λ_t) and ADM-ES* (linear λ_t). ImageNet 64×64 results are reported without classifier guidance and ImageNet 128×128 is under classifier guidance with scale=0.5

T'	Model	Unconditional			Conditional	
		CIFAR-10 32×32	LSUN 64×64	FFHQ 128×128	ImageNet 64×64	ImageNet 128×128
100	ADM	3.37	3.59	14.52	2.71	3.55
	ADM-ES	**2.17**	**2.91**	**6.77**	**2.39**	**3.37**
50	ADM	4.43	7.28	26.15	3.75	5.15
	ADM-ES	**2.49**	**3.68**	**9.50**	**3.07**	**4.33**
20	ADM	10.36	23.92	59.35	10.96	12.48
	ADM-ES	5.15	8.22	26.14	7.52	9.95
	ADM-ES*	**4.31**	**7.60**	**24.83**	**7.37**	**9.86**

Table 3: We compare ADM-ES with recent stochastic diffusion (SDE) samplers regarding FID. We report their best FID achieved under T' sampling steps.

Model	T'	Unconditional
		CIFAR-10 32×32
EDM (VP) (Karras et al., 2022)	511	2.27
EDM (VE) (Karras et al., 2022)	2047	2.23
Improved SDE (Karras et al., 2022)	1023	2.35
Restart (VP) (Xu et al., 2023)	115	2.21
SA-Solver (Xue et al., 2023)	95	2.63
ADM-IP (Ning et al., 2023)	100	2.38
ADM-ES (ours)	**50**	2.49
ADM-ES (ours)	100	**2.17**

ADM-ES at 20-step sampling with $k = 0.0005$. By contrast, using 50-step sampling, the optimal $k = 0.00007$ of ADM-ES* yields only 0.02 FID improvement (not shown in Table 2) compared with ADM-ES. Given this fact, we suggest a uniform schedule λ_t for practical application and the easy searching of the parameter b. We present the full parameters k, b used in all experiments in Appendix A.8 and provide detailed guidance on the choice of k, b. Overall, λ_t is around 1.005 on ADM baseline and a smaller λ_t should be used when a larger T' is chosen.

5.2 Epsilon Scaling Alleviates Exposure Bias

Apart from the FID improvements, we now show the exposure bias alleviated by our method using the proposed metric δ_t and we also demonstrate the sampling trajectory corrected by Epsilon Scaling. Using Algorithm 3, we measure δ_t on the dataset CIFAR-10 under 20-step sampling for ADM and ADM-ES models. Fig. 4 shows that ADM-ES obtains a lower δ_t at the end of sampling $t = 1$ than the baseline ADM, exhibiting a smaller variance error and sampling drift (see Appendix A.9 for results on other datasets).

Based on Fig. 2, we apply the same method to measure the L2-norm of $\boldsymbol{\epsilon_\theta}(\cdot)$ in the sampling phase with Epsilon Scaling. Fig. 5 indicates that our method explicitly moves the original sampling trajectory closer to the vector field learned in the training phase given the condition that the $\|\boldsymbol{\epsilon_\theta}(\boldsymbol{x}_t)\|_2$ is locally monotonic around $\boldsymbol{x}_t$. This condition is satisfied in denoising networks (Goodfellow et al., 2016; Song & Ermon, 2019) because of the monotonic score vectors around the local maximal probability density. We emphasise that Epsilon Scaling corrects the magnitude error of $\boldsymbol{\epsilon_\theta}(\cdot)$, but not the direction error. Thus we can not completely eliminate the exposure bias to achieve $\delta_t = 0$ or push the sampling trajectory to the exact training vector field.

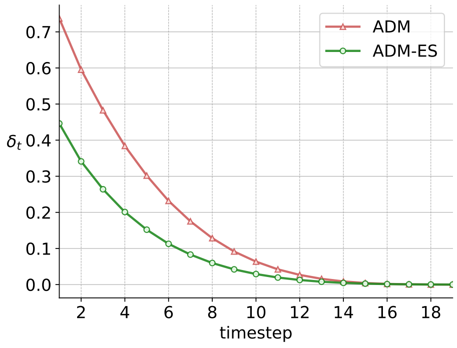


Figure 4: Exposure bias measured by δ_t on LSUN 64×64. Epsilon Scaling achieves a smaller δ_t at the end of sampling ($t = 1$)

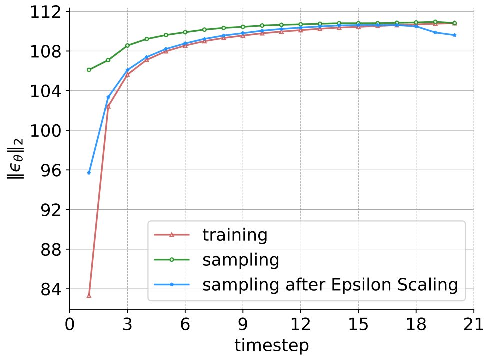


Figure 5: $\|\boldsymbol{\epsilon_\theta}(\cdot)\|_2$ on LSUN 64×64. After applying Epsilon Scaling, the sampling $\|\boldsymbol{\epsilon_\theta}\|_2$ (blue) gets closer to the training $\|\boldsymbol{\epsilon_\theta}\|_2$ (red).

5.3 Results on DDIM/DDPM

To show the generality of our proposed method, we conduct experiments on DDIM/DDPM framework across CIFAR-10 and CelebA 64×64 datasets (Liu et al., 2015). The results are detailed in Table 4, wherein the designations $\eta = 0$ and $\eta = 1$ correspond to DDIM and DDPM samplers, respectively. The findings in Table 3 illustrate that our method can further boost the performance of both DDIM and DDPM samplers on the CIFAR-10 and CelebA datasets. Specifically, our proposed Epsilon Scaling technique improves the performance of DDPM sampler on CelebA dataset by 47.7%, 63.1%, 60.7% with 20, 50, and 100 sampling steps, respectively. Similar performance improvement can also be observed on CIFAR-10 dataset. We also notice that our method brings less performance improvement for DDIM sampler. This could arise from the FID advantage of deterministic sampling under a short sampling chain and the noise term in DDPM sampler can actively correct for errors made in earlier sampling steps Karras et al. (2022).

Table 4: FID on DDIM baseline for unconditional generations.

T'	Model	CIFAR-10 32×32		CelebA 64×64	
		$\eta = 0$	$\eta = 1$	$\eta = 0$	$\eta = 1$
100	DDIM	4.06	6.73	5.67	11.33
	DDIM-ES (ours)	**3.38**	**4.01**	**5.05**	**4.45**
50	DDIM	4.82	10.29	6.88	15.09
	DDIM-ES	**4.17**	**4.57**	**6.20**	**5.57**
20	DDIM	8.21	20.15	10.43	22.61
	DDIM-ES	**6.54**	**7.78**	**10.38**	**11.83**

Table 5: FID on EDM baseline and CIFAR-10 dateset.

T'	Model	Unconditional		Conditional	
		Heun	Euler	Heun	Euler
35	EDM	1.97	3.81	1.82	3.74
	EDM-ES (ours)	**1.95**	**2.80**	**1.80**	**2.59**
21	EDM	2.33	6.29	2.17	5.91
	EDM-ES	**2.24**	**4.32**	**2.08**	**3.74**
13	EDM	7.16	12.28	6.69	10.66
	EDM-ES	**6.54**	**8.39**	**6.16**	**6.59**

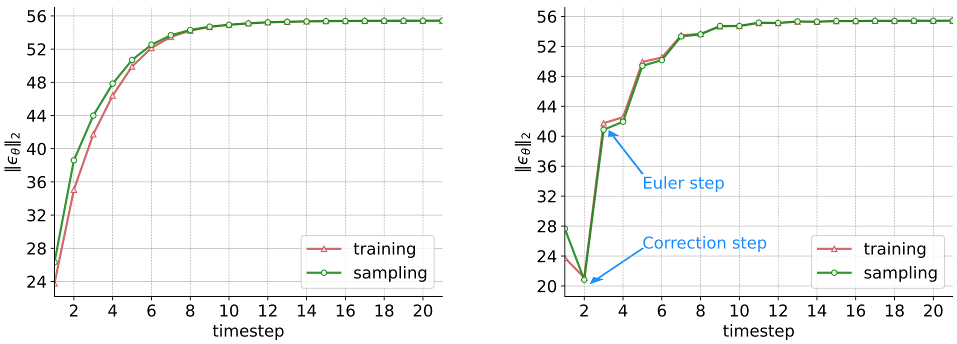
5.4 Results on EDM

We test the effectiveness of Epsilon Scaling on EDM (Karras et al., 2022) because it achieves state-of-the-art image generation under a few sampling steps and provides a unified framework for diffusion models. Since the main advantage of EDM is its Ordinary Differential Equation (ODE) solver, we evaluate our Epsilon Scaling using their Heun 2^{nd} order ODE solver (Ascher & Petzold, 1998) and Euler 1^{st} order ODE solver, respectively. Although the network output of EDM is not $\boldsymbol{\epsilon}$, we still can extract the signal $\boldsymbol{\epsilon}$ at each sampling step and then apply Epsilon Scaling.

The experiments are implemented on CIFAR-10 dataset and we report the FID results in Table 5 using VP framework. The sampling step T' in Table 5 is equivalent to the Neural Function Evaluations (NFE) used in EDM paper. Similar to the results on ADM and DDIM, Epsilon Scaling gains consistent FID improvement on EDM baseline regardless of the conditional settings and the ODE solver types. For instance, EDM-ES improves the FID from 3.81 to 2.80 and from 3.74 to 2.59 in the unconditional and conditional groups using the 35-step Euler sampler.

An interesting phenomenon in Table 5 is that the FID gain of Epsilon Scaling in the Euler sampler group is larger than that in the Heun sampler group. We claim that there are two factors accounting for this phenomenon. On the one hand, higher-order ODE solvers (for example, Heun solvers) introduce less truncation error than Euler 1^{st} order solvers. On the other hand, the correction steps in the Heun solver reduce the exposure bias by pulling the drifted sampling trajectory back to the accurate vector field. We illustrate these two factors through Fig. 6 which is plotted using the same method of Fig. 2. It is apparent from Fig. 6 that the Heun sampler exhibits a smaller gap between the training trajectory and sampling trajectory when compared with the Euler sampler. This corresponds to the truncation error factor in these two ODE solvers. Furthermore, in the Heun 2^{nd} ODE sampler, the prediction error (cause of exposure bias) made in each Euler step is corrected in the subsequent Correction step (Fig. 6(b)), resulting in a reduced exposure bias. This exposure bias perspective explains the superiority of the Heun solver in diffusion models beyond the truncation error viewpoint.

5.5 Results on LDM



(a) EDM: Euler 1^{st} order sampler

(b) EDM: Heun 2^{nd} order sampler

Figure 6: $\|\boldsymbol{\epsilon_\theta}(\cdot)\|_2$ during training and sampling on CIFAR-10. We use 21-step sampling and report the L2-norm using 50k samples at each timestep. The sampling is from right to left in the figures.

To further verify the generality of Epsilon Scaling, we adopt Latent Diffusion Model (LDM) as the base model which introduces an Autoeoconder and performs the diffusion process in the latent space (Rombach et al., 2022). We test the performance of Epsilon Scaling (LDM-ES) on FFHQ 256×256 and CelebA-HQ 256×256 datasets using T' steps DDPM sampler. It is clear from Table 6 that Epsilon Scaling gains substantial FID improvements on the two high-resolution datasets, where LDM-ES achieves 15.68 FID under $T' = 20$ on CelebA-HQ, almost half that of LDM. Epsilon Scaling also yields better FID under 50 and 100 sampling steps on CelebA-HQ with 7.36 FID at $T' = 100$. Similar FID improvements are obtained on FFHQ dataset over different T'.

Table 6: FID on LDM baseline using DDPM unconditional sampling.

T'	Model	FFHQ 256×256	CelebA-HQ 256×256
100	LDM LDM-ES (ours)	10.90 **9.83**	9.31 **7.36**
50	LDM LDM-ES	14.34 **11.57**	13.95 **9.16**
20	LDM LDM-ES	33.13 **20.91**	29.62 **15.68**

5.6 Qualitative Comparison

In order to visually show the effect of Epsilon Scaling on image synthesis, we set the same random seed for the base model and our Epsilon Scaling model in the sampling phase to ensure a similar trajectory for both models. Fig. 7 displays the generated samples using 100 steps on FFHQ 128×128 dataset. It is clear that ADM-ES effectively refines the sample issues of ADM, including overexposure, underexposure, coarse background and detail defects from left to right in Fig. 7 (see Appendix A.10 for more qualitative comparisons). Besides, the qualitative comparison also empirically confirms that Epsilon Scaling guides the sampling trajectory of the base model to an adjacent but better trajectory because both models reach the same or similar modes given the common starting point $\boldsymbol{x}_T$ and the same random seed at each sampling step.

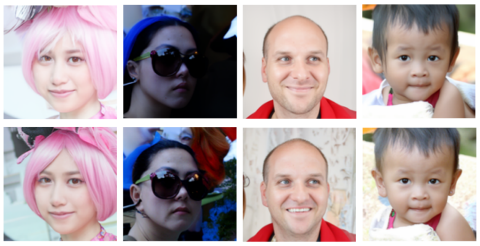


Figure 7: Qualitative comparison between ADM (first row) and ADM-ES (second row).

6 Conclusions

In this paper, we elucidate the exposure bias issue in diffusion models by analytically showing the difference between the training distribution and sampling distribution. Moreover, we discuss solutions to exposure bias and propose a training-free method to refine the deficient sampling trajectory by explicitly scaling the prediction vector. Through extensive experiments, we demonstrate that Epsilon Scaling is a generic solution to exposure bias and its simplicity enables a wide range of applications. Finally, the significant FID improvement in our method indicates the benefits of generation quality by solving the exposure bias problem. Training an accurate $\boldsymbol{\epsilon}$ or score network is a promising direction for future research.

REFERENCES

Uri M Ascher and Linda R Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam, 1998.

Fan Bao, Chongxuan Li, Jiacheng Sun, Jun Zhu, and Bo Zhang. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In *ICML*, pp. 1555–1584. PMLR, 2022a.

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *ICLR*, 2022b.

Yaniv Benny and Lior Wolf. Dynamic dual-output diffusion models. In *CVPR*, pp. 11482–11491, 2022.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. *ICLR*, 2021.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *ICCV*, pp. 14367–14376, 2021.

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of ImageNet as an alternative to the CIFAR datasets. *arXiv:1707.08819*, 2017.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *NeurIPS*, 34:8780–8794, 2021.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *ICLR*, 2017.

DC Dowson and BV666017 Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 27, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

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. *NeurIPS*, 30, 2017.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33: 6840–6851, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pp. 4401–4410, 2019.

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

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 35:26565–26577, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *ICLR*, 2021.

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

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *NeurIPS*, 32, 2019.

Mingxiao Li, Tingyu Qu, Wei Sun, and Marie-Francine Moens. Alleviating exposure bias in diffusion models through sampling with shifted time steps. *arXiv preprint arXiv:2305.15583*, 2023a.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, pp. 22511–22521, 2023b.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *ICLR*, 2022. URL `https://openreview.net/forum?id=PlKWVd2yBkY`.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, December 2015.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *NeurIPS*, 35:5775–5787, 2022.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pp. 8162–8171. PMLR, 2021.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, pp. 16784–16804. PMLR, 2022.

Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. *ICML*, 2023.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv:2204.06125*, 2022.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *ICLR*, 2016.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *ICML*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 35:36479–36494, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

Florian Schmidt. Generalization in generation: A closer look at exposure bias. In *Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP*, 2019.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021b.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *ICLR*, 2022.

Yilun Xu, Mingyang Deng, Xiang Cheng, Yonglong Tian, Ziming Liu, and Tommi Jaakkola. Restart sampling for improving generative processes. *arXiv preprint arXiv:2306.14878*, 2023.

Shuchen Xue, Mingyang Yi, Weijian Luo, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhi-Ming Ma. Sa-solver: Stochastic adams solver for fast sampling of diffusion models. *arXiv preprint arXiv:2309.05019*, 2023.

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.

Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *arXiv preprint arXiv:2302.04867*, 2023.

Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *CVPR*, pp. 22490–22499, 2023.

Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, pp. 5826–5835, 2021.

A Appendix

A.1 Derivation of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ for DDPM

We show the full derivation of Eq. 3 below. From Eq. 10 to Eq. 11, we plug in $\boldsymbol{x}_{\boldsymbol{\theta}}^{t+1} = \boldsymbol{x}_0 + e_{t+1}\boldsymbol{\epsilon}_0$ (Eq. 7) and $\boldsymbol{x}_{t+1} = \sqrt{\bar{\alpha}_{t+1}}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon}$ (Eq. 1), thus a sample from $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ is:

$$\begin{align*}
\hat{\boldsymbol{x}}_t &= \mu_{\boldsymbol{\theta}}(\boldsymbol{x}_{t+1}, t+1) + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1 \\
&= \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}\boldsymbol{x}_{\boldsymbol{\theta}}^{t+1} + \frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\boldsymbol{x}_{t+1} + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1 \tag{10} \\
&= \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}(\boldsymbol{x}_0 + e_{t+1}\boldsymbol{\epsilon}_0) + \frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}(\sqrt{\bar{\alpha}_{t+1}}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon}) + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1 \tag{11} \\
&= \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\sqrt{\bar{\alpha}_{t+1}}\boldsymbol{x}_0 + \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1}\boldsymbol{\epsilon}_0 + \frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\sqrt{1-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon} + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1 \\
&= \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1} + \sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)\sqrt{\bar{\alpha}_{t+1}}}{1-\bar{\alpha}_{t+1}}\boldsymbol{x}_0 + \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1}\boldsymbol{\epsilon}_0 + \frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\sqrt{1-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon} + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1 \\
&= \frac{\sqrt{\bar{\alpha}_t}(1-\alpha_{t+1}) + \sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)\sqrt{\bar{\alpha}_{t+1}}}{1-\bar{\alpha}_{t+1}}\boldsymbol{x}_0 + \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1}\boldsymbol{\epsilon}_0 + \frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\sqrt{1-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon} + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1 \\
&= \frac{\sqrt{\bar{\alpha}_t}(1-\alpha_{t+1}) + \alpha_{t+1}(1-\bar{\alpha}_t)\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_{t+1}}\boldsymbol{x}_0 + \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1}\boldsymbol{\epsilon}_0 + \frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\sqrt{1-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon} + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1 \\
&= \frac{\sqrt{\bar{\alpha}_t}(1-\alpha_{t+1}+\alpha_{t+1}-\bar{\alpha}_{t+1})}{1-\bar{\alpha}_{t+1}}\boldsymbol{x}_0 + \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1}\boldsymbol{\epsilon}_0 + \frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\sqrt{1-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon} + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1 \\
&= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1}\boldsymbol{\epsilon}_0 + \frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\sqrt{1-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon} + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1 \tag{12}
\end{align*}$$

From Eq. 12, we know that the mean of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ is $\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$. We now focus on the variance by looking at $\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1}\boldsymbol{\epsilon}_0 + \frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\sqrt{1-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon} + \sqrt{\tilde{\beta}_{t+1}}\boldsymbol{\epsilon}_1$:

$$\begin{align*}
Var(\hat{\boldsymbol{x}}_t) &= (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2 + (\frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\sqrt{1-\bar{\alpha}_{t+1}})^2 + \tilde{\beta}_{t+1} \tag{13} \\
&= (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2 + (\frac{\sqrt{\alpha_{t+1}}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_{t+1}}\sqrt{1-\bar{\alpha}_{t+1}})^2 + \frac{(1-\bar{\alpha}_t)(1-\alpha_{t+1})}{1-\bar{\alpha}_{t+1}} \\
&= (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2 + \frac{\alpha_{t+1}(1-\bar{\alpha}_t)^2}{1-\bar{\alpha}_{t+1}} + \frac{(1-\bar{\alpha}_t)(1-\alpha_{t+1})}{1-\bar{\alpha}_{t+1}} \\
&= (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2 + \frac{\alpha_{t+1}(1-\bar{\alpha}_t)^2 + (1-\bar{\alpha}_t)(1-\alpha_{t+1})}{1-\bar{\alpha}_{t+1}} \\
&= (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2 + \frac{(1-\bar{\alpha}_t)[\alpha_{t+1}(1-\bar{\alpha}_t) + (1-\alpha_{t+1})]}{1-\bar{\alpha}_{t+1}} \\
&= (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2 + \frac{(1-\bar{\alpha}_t)[\alpha_{t+1} - \bar{\alpha}_{t+1} + 1 - \alpha_{t+1}]}{1-\bar{\alpha}_{t+1}} \\
&= (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2 + \frac{(1-\bar{\alpha}_t)[1-\bar{\alpha}_{t+1}]}{1-\bar{\alpha}_{t+1}} \\
&= (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2 + 1 - \bar{\alpha}_t \tag{14}
\end{align*}$$

A.2 Derivation of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\boldsymbol{x}_{t+1})$ and More for DDPM

Since $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\boldsymbol{x}_{t+1})$ contains two consecutive sampling steps: $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t}|\boldsymbol{x}_{t+1})$ and $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\hat{\boldsymbol{x}}_{t})$, we can solve out $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\boldsymbol{x}_{t+1})$ by iterative plugging-in. According to $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t}|\boldsymbol{x}_{t+1}) = \mathcal{N}(\hat{\boldsymbol{x}}_t; \mu_{\boldsymbol{\theta}}(\boldsymbol{x}_{t+1}, t+1), \tilde{\beta}_{t+1}\boldsymbol{I})$ and Eq. 10, we know that $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\hat{\boldsymbol{x}}_{t}) = \mathcal{N}(\hat{\boldsymbol{x}}_{t-1}; \mu_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t, t), \tilde{\beta}_t\boldsymbol{I})$ and a sample from $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\hat{\boldsymbol{x}}_{t})$ is:

$$\hat{\boldsymbol{x}}_{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_{\boldsymbol{\theta}}^t + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\hat{\boldsymbol{x}}_t + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}_1. \tag{15}$$

From Table 1, we know that $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t}|\boldsymbol{x}_{t+1}) = \mathcal{N}(\hat{\boldsymbol{x}}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t + (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2)\boldsymbol{I})$, so plug in $\hat{\boldsymbol{x}}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t + (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2}\boldsymbol{\epsilon}_3$ into Eq. 15, we know a sample from $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\boldsymbol{x}_{t+1})$ is:

$$\hat{\boldsymbol{x}}_{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_{\boldsymbol{\theta}}^t + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t + (\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2}\boldsymbol{\epsilon}_3) + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}_1. \tag{16}$$

By denoting $(\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}e_{t+1})^2$ as $f(t)$ and plugging in $\boldsymbol{x}_{\boldsymbol{\theta}}^t = \boldsymbol{x}_0 + e_t\boldsymbol{\epsilon}_0$ (Eq. 7), we have:

$$\begin{aligned}
\hat{\boldsymbol{x}}_{t-1} &= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}(\boldsymbol{x}_0 + e_t\boldsymbol{\epsilon}_0) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t + f(t)}\boldsymbol{\epsilon}_3) + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}_1 && (17)\\
&\approx \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}(\boldsymbol{x}_0 + e_t\boldsymbol{\epsilon}_0) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_3 + \frac{1}{2\sqrt{1-\bar{\alpha}_t}}f(t)\boldsymbol{\epsilon}_3) + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}_1 && (18)\\
&\approx \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}e_t\boldsymbol{\epsilon}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0\\
&\quad + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}(\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_3 + \frac{1}{2\sqrt{1-\bar{\alpha}_t}}f(t)\boldsymbol{\epsilon}_3) + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}_1 && (19)\\
&\approx \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0 + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}e_t\boldsymbol{\epsilon}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}(\sqrt{1-\bar{\alpha}_t} + \frac{1}{2\sqrt{1-\bar{\alpha}_t}}f(t))\boldsymbol{\epsilon}_3 + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}_1 && (20)
\end{aligned}$$

Taylor's theorem is used from Eq. 17 to Eq. 18. The process from Eq. 19 to Eq. 20 is similar to the simplification from Eq. 11 to Eq. 12. From Eq. 20, we know that the mean of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\boldsymbol{x}_{t+1})$ is $\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0$. We now focus on the variance:

$$\begin{aligned}
Var(\hat{\boldsymbol{x}}_{t-1}) &= (\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}e_t)^2 + (\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\sqrt{1-\bar{\alpha}_t})^2 + \tilde{\beta}_t\\
&\quad + (\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\frac{1}{2\sqrt{1-\bar{\alpha}_t}}f(t))^2 && (21)\\
&= 1-\bar{\alpha}_{t-1} + (\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}e_t)^2 + \frac{\alpha_t(1-\bar{\alpha}_{t-1})^2}{4(1-\bar{\alpha}_t)^3}f(t)^2 && (22)
\end{aligned}$$

The above derivation is similar to the progress from Eq. 13 to Eq. 14. Now we write the mean and variance of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\boldsymbol{x}_{t+1})$ in Table 7. In the same spirit of iterative plugging-in, we could derive $(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_T)$ which has the mean $\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$ and variance larger than $(1-\bar{\alpha}_t)\boldsymbol{I}$.

A.3 Derivation of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ for DDIM

We first review the derivation of the reverse diffusion $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ for DDIM. To keep the symbols consistent in this paper, we continue to use the notations of DDPM in the derivation of DDIM.

Table 7: The distribution $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)$ during training and $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}|\boldsymbol{x}_{t+1})$ during DDPM sampling.

	Mean	Variance
$q(\boldsymbol{x}_{t-1}\|\boldsymbol{x}_0)$	$\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0$	$(1-\bar{\alpha}_{t-1})\boldsymbol{I}$
$q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{t-1}\|\boldsymbol{x}_{t+1})$	$\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0$	$(1-\bar{\alpha}_{t-1}+(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}e_t)^2+\frac{\alpha_t(1-\bar{\alpha}_{t-1})^2}{4(1-\bar{\alpha}_t)^3}f(t)^2)\boldsymbol{I}$

Recall that DDIM and DDPM have the same loss function because they share the same marginal distribution $q(\boldsymbol{x}_t|\boldsymbol{x}_0)=\mathcal{N}(\boldsymbol{x}_t;\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0,(1-\bar{\alpha}_t)\boldsymbol{I})$. But the posterior $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)$ of DDIM is obtained under Non-Markovian diffusion process and is given by Song et al. (2021a):

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)=\mathcal{N}(\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}},\sigma_t^2\boldsymbol{I}). \tag{23}$$

Similar to DDPM, the reverse distribution of DDIM is parameterized as $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)=q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_{\boldsymbol{\theta}}^t)$, where $\boldsymbol{x}_{\boldsymbol{\theta}}^t$ means the predicted $\boldsymbol{x}_0$ given $\boldsymbol{x}_t$. Based on Eq. 23, the reverse diffusion $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_{\boldsymbol{\theta}}^t)$ is:

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_{\boldsymbol{\theta}}^t)=\mathcal{N}(\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_{\boldsymbol{\theta}}^t+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_{\boldsymbol{\theta}}^t}{\sqrt{1-\bar{\alpha}_t}},\sigma_t^2\boldsymbol{I}). \tag{24}$$

Again, we point out that $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_{\boldsymbol{\theta}}^t)=q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)$ holds only if $\boldsymbol{x}_{\boldsymbol{\theta}}^t=\boldsymbol{x}_0$, this requires the network to make no prediction error about $\boldsymbol{x}_0$. Theoretically, we need to consider the uncertainty of the prediction $\boldsymbol{x}_{\boldsymbol{\theta}}^t$ and model it as a probabilistic distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_t)$. Following Analytical-DPM (Bao et al., 2022b), we approximate it by a Gaussian distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_t)=\mathcal{N}(\boldsymbol{x}_{\boldsymbol{\theta}}^t;\boldsymbol{x}_0,e_t^2\boldsymbol{I})$, namely $\boldsymbol{x}_{\boldsymbol{\theta}}^t=\boldsymbol{x}_0+e_t\boldsymbol{\epsilon}_0$. Thus, the practical reverse diffusion $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_{\boldsymbol{\theta}}^t)$ is

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_{\boldsymbol{\theta}}^t)=\mathcal{N}(\sqrt{\bar{\alpha}_{t-1}}(\boldsymbol{x}_0+e_t\boldsymbol{\epsilon}_0)+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}(\boldsymbol{x}_0+e_t\boldsymbol{\epsilon}_0)}{\sqrt{1-\bar{\alpha}_t}},\sigma_t^2\boldsymbol{I}). \tag{25}$$

Note that $\sigma_t=0$ for DDIM sampler, so a sample $\boldsymbol{x}_{t-1}$ from $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_{\boldsymbol{\theta}}^t)$ is:

$$\begin{aligned}
\boldsymbol{x}_{t-1}&=\sqrt{\bar{\alpha}_{t-1}}(\boldsymbol{x}_0+e_t\boldsymbol{\epsilon}_0)+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}(\boldsymbol{x}_0+e_t\boldsymbol{\epsilon}_0)}{\sqrt{1-\bar{\alpha}_t}}+\sigma_t\boldsymbol{\epsilon}_4\\
&=\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0+\sqrt{\bar{\alpha}_{t-1}}e_t\boldsymbol{\epsilon}_0+\sqrt{1-\bar{\alpha}_{t-1}}\cdot\frac{\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}}-\sqrt{1-\bar{\alpha}_{t-1}}\cdot\frac{\sqrt{\bar{\alpha}_t}e_t\boldsymbol{\epsilon}_0}{\sqrt{1-\bar{\alpha}_t}} \qquad (26)\\
&=\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0+\sqrt{\bar{\alpha}_{t-1}}e_t\boldsymbol{\epsilon}_0+\sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_5-\sqrt{1-\bar{\alpha}_{t-1}}\cdot\frac{\sqrt{\bar{\alpha}_t}e_t\boldsymbol{\epsilon}_0}{\sqrt{1-\bar{\alpha}_t}} \qquad (27)\\
&=\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0+\sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_5+(\sqrt{\bar{\alpha}_{t-1}}e_t-\sqrt{1-\bar{\alpha}_{t-1}}\cdot\frac{\sqrt{\bar{\alpha}_t}e_t}{\sqrt{1-\bar{\alpha}_t}})\boldsymbol{\epsilon}_0 \qquad (28)
\end{aligned}$$

From Eq. 26 to Eq. 27, we plug in $\boldsymbol{x}_t=\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_5$ where $\boldsymbol{\epsilon}_5\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$. We now compute the sampling distribution $q(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1},\boldsymbol{x}_{\boldsymbol{\theta}}^{t+1})$ which is the same distribution as $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_{\boldsymbol{\theta}}^t)$ by replacing the index t with $t+1$ and using $\hat{\boldsymbol{x}}_t$ to highlight it is a generated sample. According to Eq. 28, a sample $\hat{\boldsymbol{x}}_t$ from $q(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1},\boldsymbol{x}_{\boldsymbol{\theta}}^{t+1})$ is:

$$\hat{\boldsymbol{x}}_t=\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_5+(\sqrt{\bar{\alpha}_t}e_{t+1}-\sqrt{1-\bar{\alpha}_t}\cdot\frac{\sqrt{\bar{\alpha}_{t+1}}e_{t+1}}{\sqrt{1-\bar{\alpha}_{t+1}}})\boldsymbol{\epsilon}_0 \tag{29}$$

From Eq. 29, we know the mean of $q(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1}, \boldsymbol{x}_{\boldsymbol{\theta}}^{t+1})$ is $\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$. We now calculate the variance by looking at $\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_5 + (\sqrt{\bar{\alpha}_t}e_{t+1} - \sqrt{1-\bar{\alpha}_t} \cdot \frac{\sqrt{\bar{\alpha}_{t+1}}e_{t+1}}{\sqrt{1-\bar{\alpha}_{t+1}}})\boldsymbol{\epsilon}_0$:

$$\begin{aligned}
Var(\hat{\boldsymbol{x}}_t) &= (\sqrt{1-\bar{\alpha}_t})^2 + (\sqrt{\bar{\alpha}_t}e_{t+1} - \sqrt{1-\bar{\alpha}_t} \cdot \frac{\sqrt{\bar{\alpha}_{t+1}}e_{t+1}}{\sqrt{1-\bar{\alpha}_{t+1}}})^2 \\
&= 1-\bar{\alpha}_t + (\sqrt{\bar{\alpha}_t} - \sqrt{1-\bar{\alpha}_t} \cdot \frac{\sqrt{\bar{\alpha}_{t+1}}}{\sqrt{1-\bar{\alpha}_{t+1}}})^2 e_{t+1}^2 \\
&= 1-\bar{\alpha}_t + (\sqrt{\bar{\alpha}_t} - \frac{\sqrt{1-\bar{\alpha}_t}\sqrt{\bar{\alpha}_t}\sqrt{\alpha_{t+1}}}{\sqrt{1-\bar{\alpha}_{t+1}}})^2 e_{t+1}^2 \\
&= 1-\bar{\alpha}_t + (\sqrt{\bar{\alpha}_t}(1 - \frac{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_{t+1}}}{\sqrt{1-\bar{\alpha}_{t+1}}}))^2 e_{t+1}^2 \\
&= 1-\bar{\alpha}_t + \bar{\alpha}_t(1 - \frac{\sqrt{\alpha_{t+1}-\bar{\alpha}_{t+1}}}{\sqrt{1-\bar{\alpha}_{t+1}}})^2 e_{t+1}^2 \\
&= 1-\bar{\alpha}_t + (1 - \sqrt{\frac{\alpha_{t+1}-\bar{\alpha}_{t+1}}{1-\bar{\alpha}_{t+1}}})^2 \bar{\alpha}_t e_{t+1}^2
\end{aligned} \tag{30}$$

As a result, we can write the mean and variance of the sampling distribution $q(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1}, \boldsymbol{x}_{\boldsymbol{\theta}}^{t+1})$, i.e. $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$, and compare it with the training distribution $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ in Table 8.

Table 8: The mean and variance of $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ during training and $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ during DDIM sampling.

	Mean	Variance
$q(\boldsymbol{x}_t\|\boldsymbol{x}_0)$	$\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$	$(1-\bar{\alpha}_t)\boldsymbol{I}$
$q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t\|\boldsymbol{x}_{t+1})$	$\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$	$(1-\bar{\alpha}_t + (1 - \sqrt{\frac{\alpha_{t+1}-\bar{\alpha}_{t+1}}{1-\bar{\alpha}_{t+1}}})^2 \bar{\alpha}_t e_{t+1}^2)\boldsymbol{I}$

Since $\alpha_{t+1} < 1$, $\sqrt{\frac{\alpha_{t+1}-\bar{\alpha}_{t+1}}{1-\bar{\alpha}_{t+1}}} < 1$ and $(1 - \sqrt{\frac{\alpha_{t+1}-\bar{\alpha}_{t+1}}{1-\bar{\alpha}_{t+1}}}) > 0$ hold for any t in Eq. 30. Similar to DDPM sampler, the variance of $q(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1}, \boldsymbol{x}_{\boldsymbol{\theta}}^{t+1})$ is always larger than that of $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ by the magnitude $(1 - \sqrt{\frac{\alpha_{t+1}-\bar{\alpha}_{t+1}}{1-\bar{\alpha}_{t+1}}})^2 \bar{\alpha}_t e_{t+1}^2$, indicating the exposure bias issue in DDIM sampler.

A.4 Practical Variance Error of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ and $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_T)$

We measure the single-step variance error of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_{t+1})$ and multi-step variance error of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_T)$ using Algorithm 1 and Algorithm 2, respectively. Note that, the multi-step variance error measurement is similar to the exposure bias δ_t evaluation and we denote the single-step variance error as Δ_t and represent the multi-step variance error as Δ_t'. The experiments are implemented on CIFAR-10 (Krizhevsky et al., 2009) dataset and ADM model (Dhariwal & Nichol, 2021). The key difference between Δ_t and Δ_t' measurement is that the former can get access to the ground truth input $\boldsymbol{x}_t$ at each sampling step t, while the latter is only exposed to the predicted $\hat{\boldsymbol{x}}_t$ in the iterative sampling process.

Algorithm 1 Variance error under single-step sampling

```
Initialize $\Delta_t = 0$, $n_t = list()$ $(\forall t \in \{1, ..., T-1\})$
for $t := T, ..., 1$ do
    repeat
        $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
        $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
        $\hat{\boldsymbol{x}}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)) + \sqrt{\tilde{\beta}_t}\boldsymbol{z} \quad (\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}))$
        $n_{t-1}.append(\hat{\boldsymbol{x}}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0)$
    until 50k iterations
end for
for $t := T, ..., 1$ do
    $\hat{\beta}_t = numpy.var(n_t)$
    $\Delta_t = \hat{\beta}_t - \bar{\beta}_t$
end for
```

Algorithm 2 Variance error under multi-step sampling

```
Initialize $\delta_t = 0$, $n_t = list()$ $(\forall t \in \{1, ..., T-1\})$
repeat
    $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
    $\boldsymbol{x}_T = \sqrt{\bar{\alpha}_T}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_T}\boldsymbol{\epsilon}$
    for $t := T, ..., 1$ do
        if $t == T$ then $\hat{\boldsymbol{x}}_t = \boldsymbol{x}_T$
        $\hat{\boldsymbol{x}}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\hat{\boldsymbol{x}}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t, t)) + \sqrt{\tilde{\beta}_t}\boldsymbol{z} \quad (\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}))$
        $n_{t-1}.append(\hat{\boldsymbol{x}}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0)$
    end for
until 50k iterations
for $t := T, ..., 1$ do
    $\hat{\beta}_t = numpy.var(n_t)$
    $\Delta'_t = \hat{\beta}_t - \bar{\beta}_t$
end for
```

A.5 Metric for Exposure Bias

The key step of Algorithm 3 is that we subtract the mean $\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0$ and the remaining term $\hat{\boldsymbol{x}}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0$ corresponds to the stochastic term of $q(\hat{\boldsymbol{x}}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}^t_{\boldsymbol{\theta}})$. In our experiments, we use $N = 50,000$ samples to compute the variance $\hat{\beta}_t$.

Algorithm 3 Measurement of Exposure Bias δ_t

```
Initialize $\delta_t = 0$, $n_t = list()$ $(\forall t \in \{1, ..., T-1\})$
repeat
    $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
    compute $\boldsymbol{x}_T$ using Eq. 1
    for $t := T, ..., 1$ do
        if $t == T$ then $\hat{\boldsymbol{x}}_t = \boldsymbol{x}_T$
        $\hat{\boldsymbol{x}}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\hat{\boldsymbol{x}}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t, t)) + \sqrt{\tilde{\beta}_t}\boldsymbol{z} \quad (\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}))$
        $n_{t-1}.append(\hat{\boldsymbol{x}}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0)$
    end for
until $N$ iterations
for $t := T, ..., 1$ do
    $\hat{\beta}_t = numpy.var(n_t)$
    $\delta_t = (\sqrt{\hat{\beta}_t} - \sqrt{\bar{\beta}_t})^2$
end for
```

A.6 Correlation between Exposure Bias Metric and FID

We define the exposure bias at timestep t as $\delta_t = (\sqrt{\hat{\beta}_t} - \sqrt{\bar{\beta}_t})^2$, where $\bar{\beta}_t = 1 - \bar{\alpha}_t$ denotes the variance of $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ during training and $\hat{\beta}_t$ presents the variance of $q_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_t|\boldsymbol{x}_T)$ in the regular sampling process. Although δ_t measures the discrepancy between network inputs and FID to evaluate the difference between training data and network outputs, we empirically find a strong correlation between δ_t and FID, which could arise from the benefit of defining δ_t from the Fréchet distance Dowson & Landau

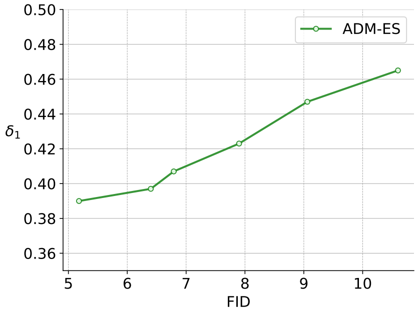


Figure 8: Correlation between FID - δ_1.

(1982) perspective. In Fig. 8, we present the FID-δ_1 relationships on CIFAR-10 and use 20-step sampling, wherein δ_1 represents the exposure bias in the last sampling step $t = 1$. Additionally, δ_t has the advantage of indicating the network input quality at any intermediate timestep t. Taking Fig. 4 as an example, we can see that the input quality decreases dramatically near the end of sampling ($t = 1$) as δ_t increases significantly.

A.7 Recall and Precision Results

Our method Epsilon Scaling does not affect the recall and precision of the base model. We present the complete recall and precision (Kynkäänniemi et al., 2019) results in Table 9 using the code provided by ADM (Dhariwal & Nichol, 2021). ADM-ES achieve higher recalls and slightly lower previsions across the five datasets. But the overall differences are minor.

Table 9: Recall and precision of ADM and ADM-ES using 100-step sampling.

Model	CIFAR-10 32×32		LSUN tower 64×64		FFHQ 128×128		ImageNet 64×64		ImageNet 128×128	
	recall	precision	recall	precision	recall	precision	recall	precision	recall	precision
ADM	0.591	**0.691**	0.605	**0.645**	0.497	**0.696**	0.621	**0.738**	0.586	0.771
ADM-ES	**0.613**	0.684	**0.606**	0.641	**0.545**	0.683	**0.632**	0.726	**0.592**	0.771

A.8 Epsilon Scaling Parameters: k, b

We present the parameters k, b of Epsilon Scaling we used in all of our experiments in Table 10, Table 11 and Table 12 for reproducibility. Apart from that, we provide guidance on how to search for the optimal parameters even though they are dependent on the dataset and how well the base model is trained. Our suggestions are:

- First search for the optimal uniform schedule λ_t (i.e. $\lambda_t = b$) with a large stride, for example, $b = 1.001, 1.003, 1.005...$.
- After locating the coarse range of the optimal b, apply a smaller stride to finetune the initial b.
- In general, the optimal b will decrease as the number of sampling steps T' increases.
- If one is interested in finding the optimal linear schedule $\lambda_t = kt + b$, we suggest taking the optimal uniform schedule $\lambda_t = b$ as the baseline and maintaining the mean of $\lambda_t = kt + b$ equal to the baseline. Then, a small k (for example, 0.0001) is a good starting point.
- Instead of generating 50k samples for FID computation, we find that 10k samples are enough for parameter searching.

Table 10: Epsilon Scaling schedule $\lambda_t = kt + b$ we used on ADM baseline. We keep the FID results in the table for comparisons and remark k, b underneath FIDs

T'	Model	Unconditional			Conditional	
		CIFAR-10 32×32	LSUN tower 64×64	FFHQ 128×128	ImageNet 64×64	ImageNet 128×128
100	ADM	3.37	3.59	14.52	2.71	3.55
	ADM-ES	2.17 **(b=1.017)**	2.91 **(b=1.006)**	6.77 **(b=1.005)**	2.39 **(b=1.006)**	3.37 **(b=1.004)**
50	ADM	4.43	7.28	26.15	3,75	5.15
	ADM-ES	2.49 **(b=1.017)**	3.68 **(b=1.007)**	9.50 **(b=1.007)**	3.07 **(b=1.006)**	4.33 **(b=1.004)**
20	ADM	10.36	23.92	59.35	10.96	12.48
	ADM-ES	5.15 **(b=1.017)**	8.22 **(b=1.011)**	26.14 **(b=1.008)**	7.52 **(b=1.006)**	9.95 **(b=1.005)**
	ADM-ES*	4.31 **(k=0.0025, b=1.0)**	7.60 **(k=0.0008, b=1.0034)**	24.83 **(k=0.0004, b=1.0042)**	7.37 **(k=0.0002, b=1.0041)**	9.86 **(k=0.00022, b=1.00291)**

Table 11: Epsilon Scaling schedule $\lambda_t = kt + b, (k = 0)$ we used on DDIM/DDPM and LDM baseline. We keep the FID results in the table for comparisons and remark b underneath FIDs

T'	Model	CIFAR-10 32×32		CelebA 64×64	
		$\eta = 0$	$\eta = 1$	$\eta = 0$	$\eta = 1$
100	DDIM	4.06	6.73	5.67	11.33
	DDIM-ES	3.38 **(b=1.0014)**	4.01 **(b=1.03)**	5.05 **(b=1.003)**	4.45 **(b=1.04)**
50	DDIM	4.82	10.29	6.88	15.09
	DDIM-ES	4.17 **(b=1.0030)**	4.57 **(b=1.04)**	6.20 **(b=1.004)**	5.57 **(b=1.05)**
20	DDIM	8.21	20.15	10.43	22.61
	DDIM-ES	6.54 **(b=1.0052)**	7.78 **(b=1.05)**	10.38 **(b=1.001)**	11.83 **(b=1.06)**

T'	Model	FFHQ 256×256	CelebA-HQ 256×256
100	LDM	10.90	9.31
	LDM-ES	9.83 **(b=1.00015)**	7.36 **(b=1.0009)**
50	LDM	14.34	13.95
	LDM-ES	11.57 **(b=1.0016)**	9.16 **(b=1.003)**
20	LDM	33.13	29.62
	LDM-ES	20.91 **(b=1.007)**	15.68 **(b=1.010)**

A.9 Epsilon Scaling Alleviates Exposure Bias

In Section 5.2, we have explicitly shown that Epsilon Scaling reduces the exposure bias of diffusion models via refining the sampling trajectory and achieves a lower δ_t on CIFAR-10 dataset.

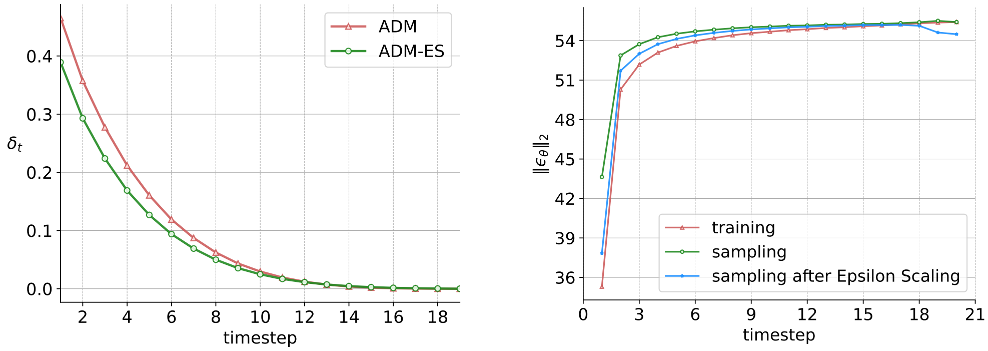
We now replicate these experiments on other datasets using the same base model ADM and 20-step sampling. Fig. 9 and Fig. 10 display the corresponding results on CIFAR-10 and FFHQ 128×128 datasets. Similar to the phenomenon on LSUN tower 64×64 (Fig. 5 and Fig. 4) , Epsilon Scaling consistently obtains a smaller exposure bias δ_t and pushes the sampling trajectory to the vector field learned in the training stage.

A.10 Qualitative Comparison

In Section 5.6, we have presented the sample quality comparison between the base model sampling and Epsilon Scaling sampling on FFHQ 128×128 dataset. Applying the same experimental settings, we show more qualitative contrasts between ADM and ADM-ES on the dataset CIFAR-10 32×32 (Fig. 11), LSUN tower 64×64 (Fig. 12), ImageNet 64×64 (Fig. 13) and ImageNet 128×128 (Fig. 14). Also, we provide the qualitative comparison between LDM and LDM-ES on the dataset CelebA-HQ 256×256 (Fig. 15). These sample comparisons clearly state that Epsilon Scaling effectively improves the sample quality from various perspectives, including illumination, colour, object coherence, background details and so on.

Table 12: Epsilon Scaling schedule $\lambda_t = kt + b, (k = 0)$ we used on EDM baseline. We keep the FID results in the table for comparisons and remark b underneath FIDs

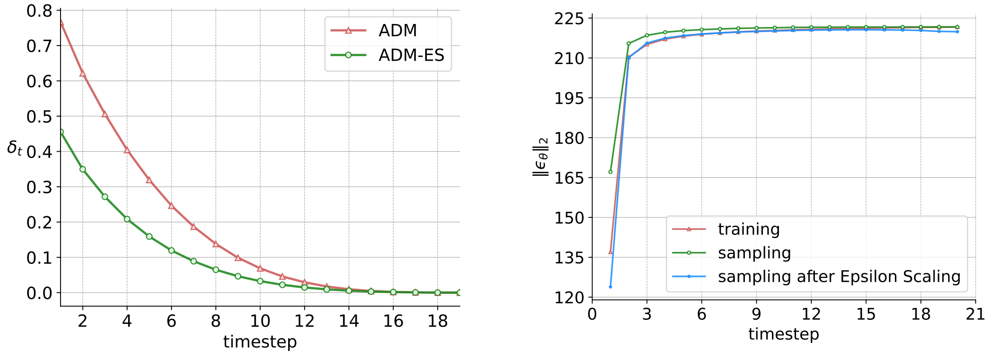
T'	Model	Unconditional		Conditional	
		Heun	Euler	Heun	Euler
35	EDM	1.97	3.81	1.82	3.74
	EDM-ES	1.95 **b=1.0005**	2.80 **b=1.0034**	1.80 **b=1.0006**	2.59 **b=1.0035**
21	EDM	2.33	6.29	2.17	5.91
	EDM-ES	2.24 **b=0.9985**	4.32 **b=1.0043**	2.08 **b=0.9983**	3.74 **b=1.0045**
13	EDM	7.16	12.28	6.69	10.66
	EDM-ES	6.54 **b=1.0060**	8.39 **b=1.0048**	6.16 **b=1.0070**	6.59 **b=1.0051**



(a) Exposure bias measured by δ_t on CIFAR-10

(b) L2-norm of $\boldsymbol{\epsilon_\theta}(\cdot)$ on CIFAR-10

Figure 9: Left: Epsilon Scaling achieves a smaller δ_t at the end of sampling ($t = 1$). Right: after applying Epsilon Scaling, the sampling $\|\boldsymbol{\epsilon_\theta}\|_2$ (blue) gets closer to the training $\|\boldsymbol{\epsilon_\theta}\|_2$ (red)



(a) Exposure bias measured by δ_t on FFHQ 128×128

(b) L2-norm of $\boldsymbol{\epsilon_\theta}(\cdot)$ on FFHQ 128×128

Figure 10: Left: Epsilon Scaling achieves a smaller δ_t at the end of sampling ($t = 1$). Right: after applying Epsilon Scaling, the sampling $\|\boldsymbol{\epsilon_\theta}\|_2$ (blue) gets closer to the training $\|\boldsymbol{\epsilon_\theta}\|_2$ (red).

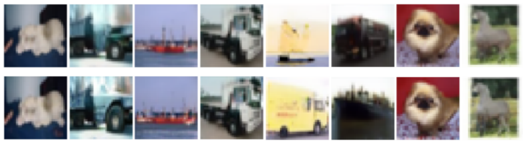


Figure 11: Qualitative comparison between ADM (first row) and ADM-ES (second row) on CIFAR-10 32×32

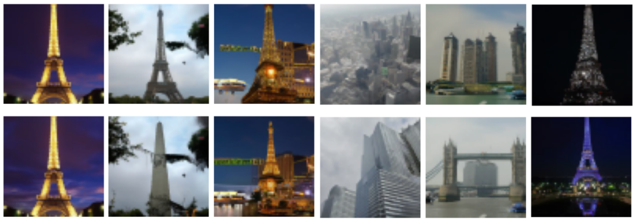


Figure 12: Qualitative comparison between ADM (first row) and ADM-ES (second row) on LSUN tower 64×64

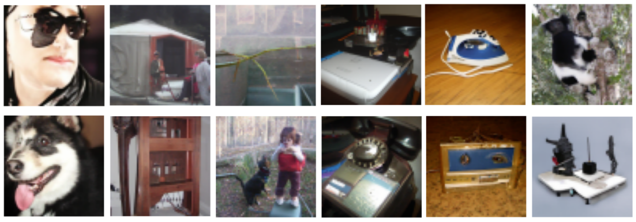


Figure 13: Qualitative comparison between ADM (first row) and ADM-ES (second row) on ImageNet 64×64

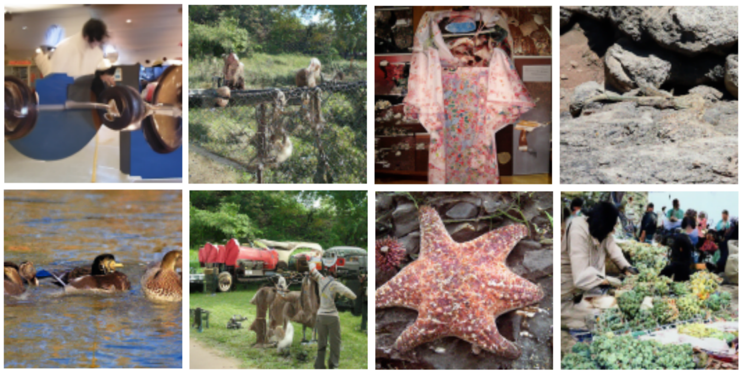


Figure 14: Qualitative comparison between ADM (first row) and ADM-ES (second row) on ImageNet 128×128

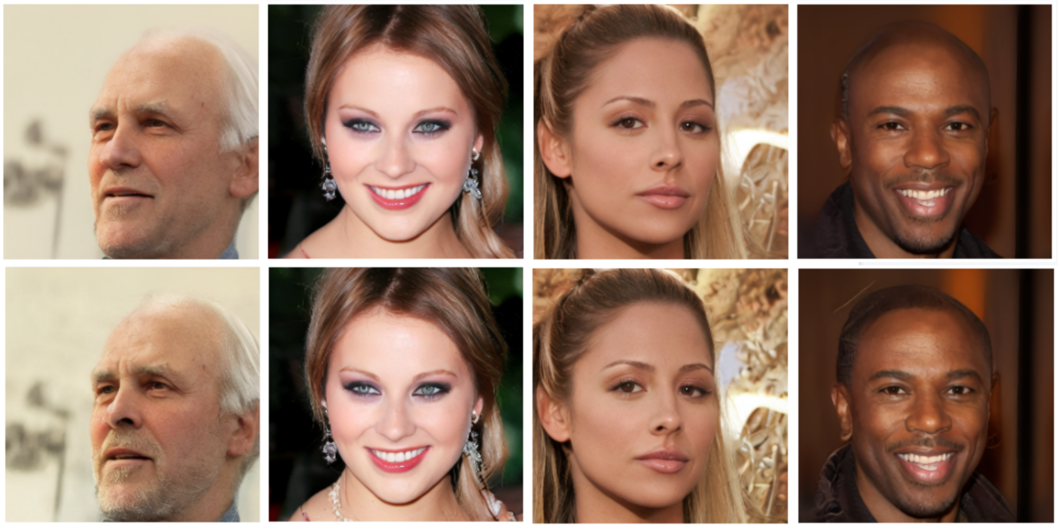


Figure 15: Qualitative comparison between LDM (first row) and LDM-ES (second row) on CelebA-HQ 256×256