Multi-Step Denoising Scheduled Sampling: Towards Alleviating Exposure Bias for Diffusion Models

Zhiyao Ren[1], **Yibing Zhan**[2], **Liang Ding**[2], **Gaoang Wang**[3], **Chaoyue Wang**[1], **Zhongyi Fan**[2], **Dacheng Tao**[1]

[1]The University of Sydney, Australia,
[2]JD Explore Academy, China,
[3]Zhejiang University, China
zren0130@uni.sydney.edu.au, zhanyibing@jd.com, liangding.liam@gmail.com, gaoangwang@intl.zju.edu.cn, chaoyue.wang@outlook.com, zyfanzy@foxmail.com, dacheng.tao@sydney.edu.au

Abstract

Denoising Diffusion Probabilistic Models (DDPMs) have achieved significant success in generation tasks. Nevertheless, the exposure bias issue, *i.e.*, the natural discrepancy between the training (the output of each step is calculated individually by a given input) and inference (the output of each step is calculated based on the input iteratively obtained based on the model), harms the performance of DDPMs. To our knowledge, few works have tried to tackle this issue by modifying the training process for DDPMs, but they still perform unsatisfactorily due to 1) partially modeling the discrepancy and 2) ignoring the prediction error accumulation. To address the above issues, in this paper, we propose a multi-step denoising scheduled sampling (MDSS) strategy to alleviate the exposure bias for DDPMs. Analyzing the formulations of the training and inference of DDPMs, MDSS 1) comprehensively considers the discrepancy influence of prediction errors on the output of the model (the Gaussian noise) and the output of the step (the calculated input signal of the next step), and 2) efficiently models the prediction error accumulation by using multiple iterations of a mathematical formulation initialized from one-step prediction error obtained from the model. The experimental results, compared with previous works, demonstrate that our approach is more effective in mitigating exposure bias in DDPM, DDIM, and DPM-solver. In particular, MDSS achieves an FID score of 3.86 in 100 sample steps of DDIM on the CIFAR-10 dataset, whereas the second best obtains 4.78. The code will be available on GitHub.

Introduction

Denoising Diffusion Probabilistic Models (DDPMs) (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) are generative models, which first destruct data by progressively adding noise and then learn the reverse process for sample generation (Yang et al. 2022). Due to the advantage of unrestricted model structure and stable training, DDPMs have swiftly gained substantial attention and become state-of-the-art approaches in generative tasks, including image generation (Dhariwal and Nichol 2021), text-to-image generation (Nichol et al. 2022; Ramesh et al. 2022; Saharia et al. 2022), text-to-video generation (Singer et al. 2022), and audio generation (Mittal et al. 2021). Recent researches on DDPMs have been primarily focused on augmenting the classical method (Ho, Jain, and Abbeel 2020) in three key areas: efficient sampling (Song, Meng, and Ermon 2021; Kong and Ping 2021; Lu et al. 2022; Salimans and Ho 2022), improved likelihood estimation (Nichol and Dhariwal 2021; Bao et al. 2022; Kingma et al. 2021), and handling multi-modal tasks (Nichol et al. 2022; Mittal et al. 2021). Nevertheless, the exposure bias issue of DDPMs has been generally overlooked.

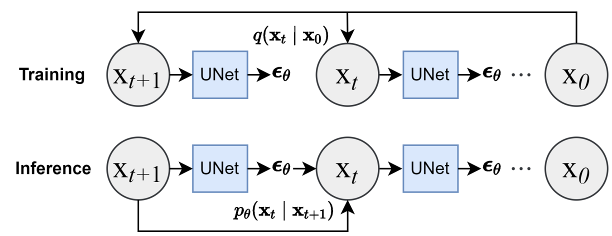


Figure 1: The discrepancy between training and inference process. In the training process, the input at time step t is derived from the forward diffusion process of $\mathbf{x}_0$. Contrarily, in the inference process, the input at time step t is obtained from the output of the previous step.

The exposure bias problem is a prevalent issue, leading to suboptimal performance, in the domain of recurrent processes, such as autoregressive text generation (Ranzato et al. 2016), arising from the disparity between the training and inference processes (Bengio et al. 2015; Zhang et al. 2019; Schmidt 2019). As shown in Fig. 1, in the training process of DDPMs, a real sample $\mathbf{x}_0$ is corrupted by introducing Gaussian noise as a Markov chain. The input to the model at step t during training is obtained based on the real sample $\mathbf{x}_0$, noise schedule α_t, and a random standard Gaussian noise ϵ: $q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)=\mathcal{N}\left(\mathbf{x}_t ; \sqrt{\bar{\alpha}_t} \mathbf{x}_0,\left(1-\bar{\alpha}_t\right) I\right)$. In contrast, in the inference process, the input to the model comes from the output of the previous steps $p_\theta\left(\mathbf{x}_t \mid \mathbf{x}_{t+1}\right)=\mathcal{N}\left(\mathbf{x}_t ; \mu_\theta\left(\mathbf{x}_{t+1}, t+1\right), \Sigma_\theta\left(\mathbf{x}_{t+1}, t+1\right)\right)$. The training process sources its input directly from the ground truth, while the inference process derives its input from model predictions with potential errors. The discrepancy between the training and inference, *i.e.*, the exposure bias issue, harms the performance of DDPMs (Deng, Kojima, and

Rush 2022; Ning et al. 2023; Li et al. 2023).

A few works (Ning et al. 2023; Deng, Kojima, and Rush 2022) have recently attempted to address the exposure bias in the training of DDPMs, but they still suffer from two problems. First, they partially model the discrepancy between training and inference of DDPMs. Specifically, the Input Perturbation (IP) method (Ning et al. 2023) introduces perturbation in the ground truth samples to simulate the inference prediction errors. Nevertheless, IP's perturbation, *e.g.,* Gaussian noise, is predefined and not equal to the true distribution of noise obtained from the model. Deng, Kojima, and Rush directly employ the scheduled sampling (SS) method (Bengio et al. 2015), originally proposed for autoregressive text generation tasks. However, the scheduled sampling method cannot fully depict the discrepancy since the input of the next step of DDPMs is calculated based on the input and output of the model (Gaussian Noise). The input of the next step of SS still contains noise from previous prediction errors. Second, IP and SS only calculate prediction errors in one step for efficiency, and they ignore that the prediction errors would be accumulated through the interaction process and further side influence the performance (2023; 2023).

In light of the above issues, in this paper, we propose a multi-step denoising scheduled sampling (MDSS) strategy to alleviate exposure bias for DDPMs. To comprehensively alleviate the influence of prediction errors, MDSS considers the exposure bias from two aspects, requiring the output of the model in the current step, *e.g.*, the Gaussian noise, to be accurately predicted, and the noise influence on the input of the next step to be reduced. To mitigate the prediction error accumulation influence, we model the accumulated prediction errors time-efficiently by using multiple iterations of a mathematical formulation initialized from the one-step prediction error obtained from the model. The process starts with a one-step model prediction to introduce the model noise and uses multiple iterations of the mathematical formulation to model the prediction error accumulation as similarly as possible. In addition, to further reduce the implementation complexity, we validate that our MDSS could improve the performance by finetuning a well-trained DDPM with small retraining steps. We conduct extensive experiments on CIFAR-10 (Krizhevsky, Hinton et al. 2009), ImageNet 64×64 (Deng et al. 2009), and LSUN 64×64 (Yu et al. 2015) datasets. Compared to previous methods: IP and SS, our MDSS exhibits better generation quality improvements in DDPM, DDIM, and DPM-Solver.

Our contributions are summarised as follows:

- We detailed analyze the discrepancy between training and inference of DDPMs and propose an effective multi-step denoising scheduled sampling (MDSS) strategy to alleviate the exposure bias for DDPMs.
- MDSS comprehensively considers the discrepancy influence of prediction errors on both the output of the model and the output of the calculated input signal per step and efficiently models accumulated prediction error by using multiple iterations of mathematical formulation initialized from the one-step prediction error of the model.
- Extensive experiments were conducted to compare the performance of current works for solving exposure bias in DDPMs. The experimental results demonstrate that our MDSS performs the best.

Preliminary Knowledge

Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models (DDPMs) consist of two processes: the forward process corrupts the data through the addition of Gaussian noise, and the reverse process reverts the forward process and generates data from standard Gaussian noise (Ho, Jain, and Abbeel 2020; Nichol and Dhariwal 2021).

Given data distribution $q(\mathbf{x}_0)$ and the noise schedule $\beta_1, \beta_2, \ldots, \beta_T$, the forward process corrupts the data as a Markov chain:

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

$$q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)=\prod_{t=1}^T q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) \tag{2}$$

When T is large enough, we can achieve $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. As mentioned in (Ho, Jain, and Abbeel 2020), we can sample to any time step directly using input $\mathbf{x}_0 \sim q(\mathbf{x}_0)$:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)=\mathcal{N}\left(\mathbf{x}_t ; \sqrt{\bar{\alpha}_t} \mathbf{x}_0,\left(1-\bar{\alpha}_t\right) I\right) \tag{3}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. Utilizing the reparameter skill, we are able to sample any step $\mathbf{x}_t$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\mathbf{x}_t=\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{1-\bar{\alpha}_t} \epsilon \tag{4}$$

Using Bayes theorem, we can obtain the posterior reverse process distribution $q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)$:

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)=\mathcal{N}\left(\mathbf{x}_{t-1} ; \tilde{\mu}\left(x_t, x_0\right), \tilde{\beta}_t \mathbf{I}\right) \tag{5}$$

$$\tilde{\mu}_t\left(\mathbf{x}_t, \mathbf{x}_0\right):=\frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0+\frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} \mathbf{x}_t \tag{6}$$

$$\tilde{\beta}_t:=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t \tag{7}$$

In the inference process, $\mathbf{x}_0$ is not available and $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ depends on the entire data distribution. Consequently, the reverse process is defined as a parameterized process:

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)=\mathcal{N}\left(\mathbf{x}_{t-1} ; \mu_\theta\left(\mathbf{x}_t, t\right), \Sigma_\theta\left(\mathbf{x}_t, t\right)\right) \tag{8}$$

$$p_\theta\left(\mathbf{x}_{0: T}\right)=p\left(\mathbf{x}_T\right) \prod_{t=1}^T p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) \tag{9}$$

Instead of learning the mean of reverse process, Ho, Jain, and Abbeel find that predicting the noise ϵ is a better option. Empirically, they propose simplifying the loss function as follows:

$$L_{\text{simple}}=\mathbb{E}_{\mathbf{x}_0, t, \epsilon \sim \mathcal{N}(0, \mathbf{I})}\left[\left\|\epsilon-\epsilon_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right] \tag{10}$$

The training and inference algorithms are described in Alg. 1 and Alg. 2, respectively.

Algorithm 1: DDPMs Standard Training Process

1: **repeat**
2: $\quad x_0 \sim q(x_0)$;
3: $\quad t \sim \mathbb{U}(\{1, \cdots T\})$;
4: $\quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
5: $\quad$ Take gradient descent step on
$$\nabla_\theta \left\| \epsilon - \epsilon_\theta \left(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}, t \right) \right\|^2 ;$$
6: **until** converged

Algorithm 2: DDPM Standard Inference Process

1: $X_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
2: **for** $t = T, \cdots, 1$ **do**
3: $\quad z \sim \mathcal{N}(\mathbf{0}, I)$ if $t > 1$, else $z = 0$;
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta \left(\mathbf{x}_t, t \right) \right) + \sigma_t z$;
5: **end for**
6: **return** $\mathbf{x}_0$

Fast Inference Methods: DDIM and DPM-Solver

Many methods demonstrate improvement in the outcomes of fewer steps inference, with DDIM and DPM-Solver being the most prevalent. DDIM (2021) proposes a more generalized non-Markov process with the same margianl distribution of $\mathbf{x}_t$. The inference process changes such that the model first predicts the normal sample $\mathbf{x}_0$, and then, the normal sample $\mathbf{x}_0$ is used to estimate the next step in the chain. The reverse process can be sampled as follows:

$$\begin{aligned} \mathbf{x}_{t-1} = & \sqrt{\bar{\alpha}_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t} \epsilon_\theta^{(t)} \left(\mathbf{x}_t \right)}{\sqrt{\bar{\alpha}_t}} \right) \\ & + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)} \left(\mathbf{x}_t \right) + \sigma_t \epsilon_t \end{aligned} \tag{11}$$

where $\sigma_t = \sqrt{\left(1-\bar{\alpha}_{t-1}\right) / \left(1-\bar{\alpha}_t\right)} \sqrt{1-\bar{\alpha}_t / \bar{\alpha}_{t-1}}$.

DPM-Solver (Lu et al. 2022) proposes an exact formulation of the solution of diffusion ODEs by Taylor expansion (first order to third order):

$$\begin{aligned} \boldsymbol{x}_{t_{i-1} \to t_i} = & \frac{\sqrt{\bar{\alpha}_{t_i}}}{\sqrt{\bar{\alpha}_{t_{i-1}}}} \tilde{\boldsymbol{x}}_{t_{i-1}} - \sqrt{\bar{\alpha}_{t_i}} \sum_{n=0}^{k-1} \hat{\boldsymbol{\epsilon}}_\theta^{(n)} \left(\hat{\boldsymbol{x}}_{\lambda_{t_{i-1}}}, \lambda_{t_{i-1}} \right) \\ & \int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^{-\lambda} \frac{\left(\lambda - \lambda_{t_{i-1}} \right)^n}{n!} \mathrm{d}\lambda + \mathcal{O}\left(h_i^{k+1} \right) \end{aligned} \tag{12}$$

where $\lambda_t = log(\sqrt{\bar{\alpha}_t} / \sqrt{1-\bar{\alpha}})$ (one half of the log-SNR) and $h_i = \lambda_{t_i} - \lambda_{t_{i-1}}$.

More introductions of DDIM and DPM-Solver can be found in Appendix.

Multi-step Denosing Scheduled Sampling

This section presents our multi-step denosing scheduled sampling (MDSS). As shown in line 5 in Alg. 1 and line 4 in Alg. 2, the inputs of the training and inference are different. Specifically, the input of the training process originates from the forward process. When $\mathbf{x_0}$, noise schedule, time step t, and Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ are deterministic, $\mathbf{x}_t$ is obtained by Eq. 4 and is thus also deterministic. However, the input of the inference process comes from the sampling results of previous steps, containing non-negligible errors, and is not exposed to the model during training. Besides, the subsequent multiple-step inference process will continuously amplify errors, further impacting the final sampling outcomes. Therefore, we design MDSS to alleviate the exposure bias issue in training by exposing and denoising the accumulated prediction errors of the inference process.

In the remaining part, we first elaborate on the scheduled sampling with denoising for DDPMs and then present the modeling of prediction errors with accumulation. Next, we introduce our algorithm, and last, we compare our MDSS with current methods towards exposure bias for DDPMs, *i.e.*, IP and SS.

Scheduled Sampling with Denoising

We first formulate and analyze the influence of prediction errors in one step. For simplicity, we suppose the input with prediction errors is represented as:

$$\hat{\mathbf{x}}_t = \mathbf{x}_t + \xi \tag{13}$$

where $\hat{\mathbf{x}}_t$ is the input of the current step with noise, $\mathbf{x}_t$ is the ground-truth input without noises, and ξ is the prediction errors modeling from the inference process. Here, we only consider additive noise following IP and SS. The discussion of other types of noises remains a challenge for future work.

According to the equation of inference process in line 4 of Alg. 2, we can obtain subsequent inference step with noise:

$$\begin{aligned} \hat{\mathbf{x}}_{t-1} & = \frac{1}{\sqrt{\alpha_t}} \left(\hat{\mathbf{x}}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta \left(\hat{\mathbf{x}}_\mathbf{t}, t \right) \right) \\ & = \frac{1}{\sqrt{\alpha_t}} \left(\underbrace{\mathbf{x}_\mathbf{t} + \xi}_{\text{Input Signal}} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \underbrace{\boldsymbol{\epsilon}_\theta \left(\mathbf{x}_\mathbf{t} + \xi, t \right)}_{\text{Model Prediction}} \right) \end{aligned} \tag{14}$$

where we ignore the variance item for simplicity since it is given directly and devoid of model prediction noise in most of works (Ho, Jain, and Abbeel 2020).

It can be observed from Eq. 14 that there are two types of influence for the output of current step: the model prediction and the input signal. Previous methods only consider model prediction. In contrast, we comprehensively mitigate both influences in the sampling process.

For influence within the model prediction, we mitigate it by training with noise-free training objective. We utilize $\mathbf{x}_t$, which contains no noise, to obtain the training objective. $\mathbf{x}_t$ can be derived by sampling from the posterior distribution $q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)$. By replacing $\mathbf{x}_0$ using ground truth ϵ_{t+1} in Eq. 6, we can obtain:

$$\begin{aligned} \mathbf{x}_t & = \tilde{\mu}_{t+1} \left(\mathbf{x}_{t+1}, \mathbf{x}_0 \right) \\ & = \frac{\sqrt{\alpha_{t+1}} \left(1-\bar{\alpha}_t \right)}{1-\bar{\alpha}_{t+1}} \mathbf{x}_{t+1} + \frac{\sqrt{\bar{\alpha}_t} \beta_{t+1}}{1-\bar{\alpha}_{t+1}} \mathbf{x}_0 \\ & = \frac{1}{\sqrt{\alpha_{t+1}}} \left(x_{t+1} - \frac{\beta_{t+1}}{\sqrt{\left(1-\bar{\alpha}_{t+1}\right)}} \boldsymbol{\epsilon}_{t+1} \right) \end{aligned} \tag{15}$$

The meaning of posterior distribution is sampling to next step, when the model prediction $\boldsymbol{\epsilon}_\theta$ equals the ground truth $\boldsymbol{\epsilon}$. Hence, $\mathbf{x}_t$ derived from the posterior distribution represents the noise-free sampling result. Utilizing the variant of Eq. 4, we obtain the noise-free training object with $\mathbf{x}_t$ and $\mathbf{x}_0$:

$$\epsilon_t = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} \tag{16}$$

For influence within the input signal, we incorporate an extra denoising channel into the model's output to achieve the denoising task. During the training process, the additional channel is trained to predict the noise from the input containing noise. The training objective is the different between input from model prediction and posterior distribution calculation:

$$\xi_t = \hat{\mathbf{x}}_t - \mathbf{x}_t \tag{17}$$

During the inference process, the input signal is denoised by subtracting the output of the denoising channel.

In summary, the loss function for the entire training process is given as:

$$\left\|\epsilon_t - \epsilon_\theta\left(\hat{\mathbf{x}}_t, t\right)\right\|^2 + \left\|\xi_t - \xi_\theta\left(\hat{\mathbf{x}}_t, t\right)\right\|^2 \tag{18}$$

More details of scheduled sampling with denoising can be found in Appendix.

Prediction Errors with Multi-step Accumulation

Previous methods (Ning et al. 2023; Deng, Kojima, and Rush 2022) generally ignore the accumulation of prediction errors: IP uses Gaussian noise, and SS only considers one-step prediction error for time efficiency. Our proposed MDSS tries to model the prediction errors with multi-step accumulations. One possible solution is to obtain the signal of time t based on the iteration process of Alg. 2. However, such an intuitive manner requires much calculation and is ineffectively applied to the training process. Therefore, MDSS models the accumulated prediction errors as similarly as possible by using multiple iterations of mathematical formulation initialized from one-step prediction error obtained from the model. Here, the one-step prediction error from the model is used to introduce model noises, and the multiple iterations of the mathematical formulation are used to model the accumulation quickly.

Specifically, we first obtain the one-step prediction error from the model by using:

$$\mathbf{x}_{t+k+1} = \sqrt{\bar{\alpha}_{t+k+1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t+k+1}}\epsilon \tag{19}$$

$$\begin{aligned}\mathbf{x}_{t+k} = &\frac{1}{\sqrt{\alpha_{t+k+1}}}\left(\mathbf{x}_{t+k+1}\right.\\ &\left.- \frac{1-\alpha_{t+k+1}}{\sqrt{1-\bar{\alpha}_{t+k+1}}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_{t+k+1}, t+k+1\right)\right)\end{aligned} \tag{20}$$

Then, we simulate the error accumulation by using the posterior distribution, which conducts the reverse process without model prediction. For further simulating of one step, we can sample by posterior directly. In order to calculate multi-step sample of posterior, we can sample as:

$$\mathbf{x}_t = \gamma_t\mathbf{x}_{t+k} + \omega_t\mathbf{x}_0 + \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}\mathbf{x}_0 \tag{21}$$

Algorithm 3: Multi-step Denoising Scheduled Sampling

- **repeat**
  - $x_0 \sim q(x_0)$;
  - $t \sim \mathbb{U}(\{1, \cdots T-k-1\})$;
  - $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
  - $\mathbf{x}_{t+k+1} = \sqrt{\bar{\alpha}_{t+k+1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t+k+1}}\boldsymbol{\epsilon}$;
  - $\hat{\mathbf{x}}_{t+k} = \frac{1}{\sqrt{\alpha_{t+k+1}}}\left(\mathbf{x_{t+k+1}} - \frac{1-\alpha_{t+k+1}}{\sqrt{1-\bar{\alpha}_{t+k+1}}}\epsilon_\theta\left(\mathbf{x}_{t+k+1}, t+k+1\right)\right)$
  - $\mathbf{x}_{t+k} = \frac{1}{\sqrt{\alpha_{t+k+1}}}\left(\mathbf{x_{t+k+1}} - \frac{1-\alpha_{t+k+1}}{\sqrt{1-\bar{\alpha}_{t+k+1}}}\boldsymbol{\epsilon}\right)$;
  - **if** $k > 0$ **then**
    - $\hat{\mathbf{x}}_t = \gamma_t\hat{\mathbf{x}}_{t+k} + \omega_t\mathbf{x}_0 + \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}\mathbf{x}_0$
    - $\mathbf{x}_t = \gamma_t\mathbf{x}_{t+k} + \omega_t\mathbf{x}_0 + \frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}\mathbf{x}_0$
  - **end if**
  - $\boldsymbol{\xi}_t = \hat{\mathbf{x}}_t - \mathbf{x}_t$;
  - $\epsilon_t = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}$;
  - Take gradient descent step on $\nabla_\theta(\left\|\epsilon_t - \epsilon_\theta\left(\hat{\mathbf{x}}_t, t\right)\right\|^2 + \left\|\xi_t - \xi_\theta\left(\hat{\mathbf{x}}_t, t\right)\right\|^2)$
- **until** converged

$$\gamma_t = \prod_{i=t}^{t+k-1}\frac{\sqrt{\alpha_{i+1}}(1-\bar{\alpha}_i)}{1-\bar{\alpha}_{i+1}} \tag{22}$$

$$\omega_t = \sum_{j=t}^{t+k-2}\frac{\prod_{n=t}^{j}\left[\sqrt{\alpha_{n+1}}(1-\bar{\alpha}_n)\right]\sqrt{\bar{\alpha}_{j+1}}\beta_{j+2}}{\prod_{m=t+1}^{j+2}\left(1-\bar{\alpha}_m\right)} \tag{23}$$

Through above process, step $t+k$ with model errors is quickly simulated for k steps errors accumulation to obtain, step t, the inputs for training. The disparity between the input with noise and the ground truth expends, mimicking the noise accumulation observed in the sampling process.

Algorithm

The training process by introducing the prediction errors with multi-step accumulation in the scheduled sampling with denoising, the algorithm of MDSS is described in Alg. 3. The training process still follows scheduled sampling, which introduces prediction errors by the scheduled ratio. For brevity, the algorithm only outlines the steps of introducing errors. More details can be found in Appendix. k represents the number of accumulation using mathmatical formulation. When $k = 0$, it defaults to a Single-step denoising scheduled sampling (SDSS).

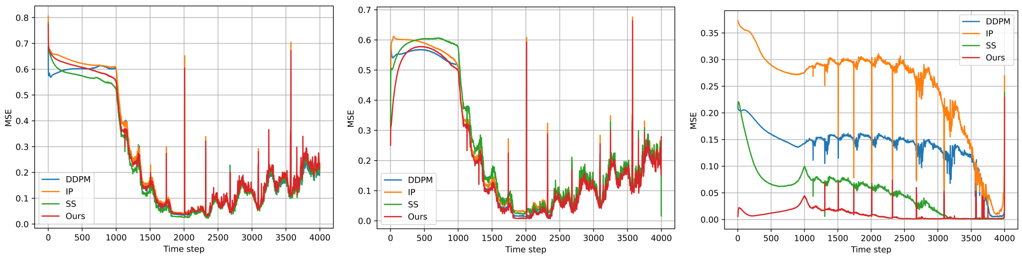
Furthermore, we validate our MDSS could be applied to a well-trained model. In such a manner, only requiring a small number of retraining steps, MDSS can improve the performance of a given DDPM, saving a lot of time and computational resources when compared with training from scratch.

Discussion

In this subsection, we compare our proposed approach with IP and SS, two state-of-the-art methods for addressing exposure bias for DDPMs, highlighting the reasons behind the superior sampling outcomes achieved by our method.

	Input	Denoise model prediction	Denoise input signal
IP	$\hat{\mathbf{x}}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\left(\boldsymbol{\epsilon}+\gamma_t\boldsymbol{\varepsilon}\right)$ (✗)	Given ϵ (✓)	✗
SS	$\hat{\mathbf{x}}_t = \frac{1}{\sqrt{\alpha_{t+1}}}\left(\mathbf{x}_{t+1} - \frac{1-\alpha_{t+1}}{\sqrt{1-\bar{\alpha}_{t+1}}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_{t+1},t\right)\right)$ (✓)	$\epsilon = \frac{\hat{\mathbf{x}}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}$ (✗)	✗
MDSS	$\hat{\mathbf{x}}_t = \gamma\hat{\mathbf{x}}_{t+k}+\omega\mathbf{x}_0+\frac{\sqrt{\bar{\alpha}_t}\beta_{t+1}}{1-\bar{\alpha}_{t+1}}\mathbf{x}_0$ (✓)	$\epsilon = \frac{\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}$ (✓)	$\xi = \hat{\mathbf{x}}_t - \mathbf{x}_t$

Table 1: The comparison of MDSS with IP and SS. The symbol ✓denotes that the process is analytically accurate and involved, while ✗ indicates that the process is either analytically incorrect or not covered.



(a) Input without errors

(b) Input with one-step errors

(c) Input with previous accumulated errors

Figure 2: The Mean Squared Errors between the model prediction and the ground truth for different methods.

Table. 1 presents the comparison. For the training input, IP uses Gaussian noise to emulate model prediction noise, which may be different from the actual noise in the inference process. For the model prediction, SS ignores the influence of the input signal, as shown in Eq. 14, and only such a model prediction cannot fully address the exposure bias and may introduce new noises. Only MDSS considers the influence of input signal and prediction error accumulation.

We conduct three experiments with well-trained models to demonstrate the efficacy of our method further. In our experiments, we employ various inputs and measure the Mean Squared Error (MSE) between the model prediction and ground truth at every step. Firstly, the model's input is calculated by ground truth and does not include errors. Fig. 2(a) shows that all methods perform nearly the same when the model inputs are free of noise. Secondly, the input of the model comes from one step of sampling of the previous step through the model. Fig. 2(b) reveals that when the input incorporates one step of model prediction noise, MDSS yields the most accurate results. Last, the input of the model comes from the sample of previous steps, which contains the accumulation errors. Fig. 2(c) shows that the error is prominent in the initial stages, likely due to the data distribution being close to standard. As the sampling process progresses, the model error diminishes rapidly. After hundreds of steps, the errors of prediction increase due to the exposure bias problem. However, using MDSS, the model prediction error did not show a trend of increasing during the sampling process, which reflects that we have well-alleviated exposure bias.

Experiment

Experimental Setup

We evaluate our method across unconditional image generation tasks on three datasets: CIFAR-10 (Krizhevsky, Hinton et al. 2009), ImageNet 64×64 (Deng et al. 2009), and LSUN tower 64×64 (Yu et al. 2015). For the CIFAR-10 and ImageNet 64×64 datasets, we fine-tune on the well-trained iDDPM (Nichol and Dhariwal 2021) models, and for LSUN tower 64×64, we keep training on the ADM (Ho and Salimans 2022) model. We employ the Frechet Inception Distance (FID) (Heusel et al. 2017) to evaluate the quality of the generated images. In order to visually show the effect of MDSS on image synthesis, we set the same random seed in the sampling phase to ensure a similar trajectory for all methods. More details regarding training hyperparameters, network architecture, FID evaluating settings, CLIP-FID (Kynkäänniemi et al. 2022; Rangwani et al. 2023) results and qualitative comparison can be found in Appendix.

Main Comparison

In this section, we compare MDSS with IP and SS on DDPM, DDIM, and DPM-Solver. For DDPM and DDIM, we sample 30, 100, and 250 steps. For DPM-Solver, we sample 10, 20, and 50 steps. In the following results, we highlight the best result and underline the second-best result.

The results of DDPM are shown in Table. 2. We can draw the following conclusions: 1) Exposure bias drops the generation performance of DDPMs, and MDSS outperforms SS and yields competitive results with IP. 2) We find that SDSS outperforms MDSS when sampling steps are larger, while MDSS performs exceptionally well with fewer sampling steps. This may be because the noise accumulation in multi-step is more effective at mitigating the fewer-step sampling process, which is prone to more significant errors.

The results of DDIM are shown in Table. 3. We can draw conclusions that: 1) Our method significantly enhances the sampling results and achieves a greater FID improvement than DDPM sampling. It is important to note that the IP method yielded subpar results in DDIM. This validates that

Dataset	Steps	DDPM	IP	SS	SDSS	MDSS
CIFAR-10	30	7.81	6.40	7.95	7.92	**5.54**
	100	3.72	**3.25**	3.62	3.47	3.49
	250	3.23	**3.02**	3.17	3.08	3.62
ImageNet	30	34.35	34.52	32.60	32.32	**31.76**
	100	25.32	24.88	23.86	**23.22**	24.28
	250	24.88	24.24	22.82	**22.48**	23.69
LSUN	30	5.98	**5.52**	5.76	5.72	5.57
	100	2.32	2.27	2.25	**2.24**	2.26
	250	2.22	1.95	1.93	**1.90**	1.92

Table 2: DDPM results on CIFAR-10, ImageNet, and LSUN tower with varying inference steps.

Dataset	Steps	DDPM	IP	SS	SDSS	MDSS
CIFAR-10	30	7.47	5.35	7.02	6.92	**5.25**
	100	4.99	6.95	4.78	4.53	**3.86**
	250	4.48	8.60	4.11	3.94	**3.92**
ImageNet	30	28.11	43.74	27.82	27.47	**27.18**
	100	25.33	49.73	24.81	**24.75**	24.96
	250	24.29	55.22	23.69	**23.44**	23.86
LSUN	30	4.26	37.38	4.01	3.94	**3.84**
	100	2.96	27.20	2.79	**2.61**	2.67
	250	4.48	32.92	4.38	4.56	**4.13**

Table 3: DDIM results on CIFAR-10, ImageNet, and LSUN tower with varying inference steps.

a predefined noise, such as the Gaussian Noise of IP, may contain a gap when compared with prediction errors of the inference process. MDSS and SDSS perform better than SS, partly because MDSS and SDSS model more accurately the prediction errors. 2) MDSS achieves better sampling results than SDSS partly because the non-markov process of DDIM might make it more susceptible to error accumulation.

We conducted experiments using DPM-Solver-1, 2, and 3 for the DPM-Solver sampling method. The CIFAR-10 results are shown in Table. 4, while results for ImageNet and LSUN are available in Appendix. We can draw the following observations: 1) DPM-Solver-1 is fundamentally similar to DDIM; therefore, our method can achieve superior sampling results compared to other approaches. 2) In DPM-Solver-2 and 3, while our approach outperforms IP and SS, there is minimal or no improvement compared to the DDPM baseline. One reason could be that DDPM and DDIM require a single model prediction during sampling, whereas DPM Solver undergoes two or three model predictions. The modeling of prediction errors should be adjusted based on DPM Solver inference, which is one of our future works.

Ablation Study

In this subsection, we conduct extensive ablation studies on the CIFAR-10 dataset to elucidate the impact of methodological components within our method.

The effect of denosing. We first assess the impact of denoise the input signal. We compare the performance of SDSS and SDSS without denoising the input signal (SDSS w/o). The results of DDPM and DDIM are shown in Table. 5. We can conclude that using an additional model channel to remove noise from the input signal can further mitigate exposure bias and enhance generated results. Nevertheless, the improvement is slight compared with the model prediction denoising effect. This might be because exposure bias predominantly stems from the noise introduced by model prediction. The input containing noise can lead to large deviations in model predictions.

Methods	Steps	DDPM	IP	SS	SDSS	MDSS
DPM-Solver-1	10	26.08	41.96	25.39	24.96	**12.11**
	20	11.46	20.79	11.17	11.04	**5.41**
	50	6.03	14.43	5.99	5.89	**4.84**
DPM-Solver-2	10	**10.43**	42.03	13.04	11.98	12.149
	20	**3.55**	7.43	4.65	4.48	5.319
	50	**3.28**	11.39	4.04	4.02	4.845
DPM-Solver-3	10	**5.99**	33.22	6.43	6.34	7.80
	20	4.07	13.32	4.23	**4.01**	5.51
	50	4.01	12.36	4.00	**3.94**	5.33

Table 4: DPM-Solver-1,2,3 on CIFAR-10 with varying inference processes.

	DDPM SDSS w/o	DDPM SDSS	DDIM SDSS w/o	DDIM SDSS
30 steps	7.93	**7.92**	6.93	**6.92**
100 steps	3.55	**3.47**	4.60	**4.53**
250 steps	3.09	**3.08**	4.05	**3.94**

Table 5: Comparison of denoising input signal on DDPM and DDIM.

The effect of multi-step training. We discuss the effection of prediction error accumulation by conducting experiments with varying steps, k: 4, 10, 20, and 50. The DDPM and DDIM inference results are presented in Table. 6 and Table. 7. In DDPM, the multi-step training approach only offers improvements at fewer inference steps. This might be due to the accumulation of errors in fewer inferences is more significant and can be mitigated by MDSS. Incorporating more steps in MDSS also leads to worse DDPM results. For DDIM, performance is enhanced using multi-step training. Significant results can also be obtained at certain sampling steps using longer multi-steps. This indicates that there is more significant noise in DDIM, and therefore MDSS can be used for more sampling steps and longer error accumulation. Choosing the number of multi-step training steps requires careful consideration to prevent exacerbating exposure bias. We advise beginning with a conservative number of steps, such as four steps, and incrementally increasing it.

Number of iterations in continue training. In our experiments, we fine-tune a well-trained DDPM using MDSS. In this section, we conduct experiments on the CIFAR-10 dataset, assessing the result of different training iterations. We calculate the FID of 250 sampling steps on every 5,000 iterations, and the result is shown in Fig. 3. The FID experiences a sharp decline in the initial phases of training and begins to converge after approximately 20,000 iterations, but we do not notice a decline if MDSS is not used.

	single-step	4 steps	10 steps	20 steps	50 steps
30 steps	7.92	**5.54**	5.92	6.70	8.31
100 steps	3.47	**3.49**	4.20	3.64	4.12
250 steps	**3.08**	3.62	3.75	3.58	3.32

Table 6: Comparison of multi-step training using different steps on DDPM.

	single-step	4 steps	10 steps	20 steps	50 steps
30 steps	6.92	5.25	**4.45**	4.52	6.98
100 steps	4.53	**3.86**	4.04	4.05	4.61
250 steps	3.94	3.92	3.96	**3.79**	4.09

Table 7: Comparison of multi-step training using different steps on DDIM.

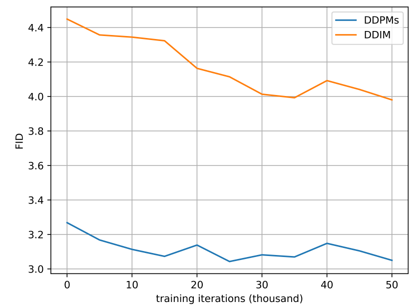


Figure 3: FID scores with respect to the number of training iterations. Each FID result is computed with 250 inference steps.

	SDSS	DDIM SDSS	MDSS	DDIM MDSS
30 steps	6.92	**5.42**	5.25	**4.82**
100 steps	4.53	**4.21**	**3.86**	3.94
250 steps	**3.94**	4.51	**3.92**	4.63

Table 8: Comparison of SDSS/MDSS with/without modification based on DDIM sampling.

This demonstrates that our method can achieve convergence results with fewer training iterations. Compared to training from scratch, which requires around 200,000 iterations, our method reduces the training time by approximately a factor of 10. More results and explanations of training from scratch can be found in Appendix.

Discussion of Modifying MSDD Based on DDIM

Many methods modify the inference process to achieve improved results when utilizing fewer sampling steps. For example, while DDIM and DDPM undergo identical training process, their inference methods are a little distinct. Analytically, by utilizing DDIM in Eq. 11 instead of DDPM in scheduled sampling, we can achieve a noise distribution that more closely mirrors the actual DDIM inference process. We compare the effect of SDSS and MDSS with the DDIM revised version, and the results are shown in Table. 8. It can be observed that with a modification, the performance of SDSS/MDSS is further improved when sampling steps are few. We suggest that When leveraging DDIM for quick sampling with minimal steps, we can employ the DDIM scheduled sampling to boost the quality of the results. We can adopt a similar approach with the DPM-Solver and we leave this as our feature work.

Related Work

Denoising Diffusion Probabilistic Models Several enhancements have been proposed based on the Denoising Diffusion Probabilistic Model (DDPM) (2020). For instance, Nichol and Dhariwal introduced the cosine noise schedule and a method for learning variances Σ_θ. Dhariwal and Nichol proposed additional classifier guidance and an improved U-net model, demonstrating that the diffusion model can achieve superior image sample quality compared to GANs. Ho and Salimans proposed a classifier-free guidance that can achieve state-of-the-art results without necessitating the training of an additional classifier.

Exposure bias Exposure bias is a prevalent issue in recurrent processes, arising due to the teacher-forcing training method (2015; 2016; 2019; 2019). Throughout the entire training process, the model is not exposed to its own predictions but given ground truth. However, during the sampling phase, the word predicted at a previous moment is used to predict the subsequent word. This discrepancy between training and sampling leads to inaccurate sampling. The Data As Demonstrator (DAD) (2015) approach tackles this issue by feeding both ground truth words and predicted words during the training process. Scheduled sampling (2015), on the other hand, replaces the teacher-forcing training method with a biased sampling approach that emulates the sampling process based on its own predictions. This paper discusses the Exposure bias in DDPMs.

To avoid the exposure bias problem caused by autoregressive factorization, another potential way is switching to non-autoregressive generation which has been validated in the field of natural language processing (2018; 2021b; 2021a; 2022) and will be explored in the future.

Conclusion

In this paper, we propose a novel training method called multi-step denoising scheduled sampling (MDSS) to mitigate the exposure bias issue. Specifically, MDSS 1) comprehensively denoises the errors in model prediction and input signal and 2) efficiently models the prediction error accumulation by mathematical formulation. Our method can be plugged into any existing DDPMs, requiring merely a few additional training iterations on the well-trained model. The experiments showcase that MDSS achieves better results in alleviating exposure bias problems compared with state-of-the-art works: IP and SS.

Even though our method achieves excellent results in both DDPM and DDIM inference, it is not well-compatible with the DPM-Solver method. Besides, we assume that the noise of model prediction is additive. We leave the discussion of DPM-Solver and other types of noises as our future work.

References

Bao, F.; Li, C.; Zhu, J.; and Zhang, B. 2022. Analytic-DPM: an Analytic Estimate of the Optimal Reverse Variance in Diffusion Probabilistic Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 1171–1179.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Deng, Y.; Kojima, N.; and Rush, A. M. 2022. Markup-to-Image Diffusion Models with Scheduled Sampling. In *The Eleventh International Conference on Learning Representations*.

Dhariwal, P.; and Nichol, A. Q. 2021. Diffusion Models Beat GANs on Image Synthesis. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 8780–8794.

Ding, L.; Wang, L.; Liu, X.; Wong, D. F.; Tao, D.; and Tu, Z. 2021a. Progressive Multi-Granularity Training for Non-Autoregressive Translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Ding, L.; Wang, L.; Liu, X.; Wong, D. F.; Tao, D.; and Tu, Z. 2021b. Understanding and Improving Lexical Choice in Non-Autoregressive Translation. In *International Conference on Learning Representations*.

Ding, L.; Wang, L.; Shi, S.; Tao, D.; and Tu, Z. 2022. Redistributing low-frequency words: Making the most of monolingual data in non-autoregressive translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Gu, J.; Bradbury, J.; Xiong, C.; Li, V. O.; and Socher, R. 2018. Non-Autoregressive Neural Machine Translation. In *International Conference on Learning Representations*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *ArXiv preprint*, abs/2207.12598.

Kingma, D.; Salimans, T.; Poole, B.; and Ho, J. 2021. Variational diffusion models. *Advances in neural information processing systems*, 34: 21696–21707.

Kong, Z.; and Ping, W. 2021. On fast sampling of diffusion probabilistic models. *ArXiv preprint*, abs/2106.00132.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Kynkäänniemi, T.; Karras, T.; Aittala, M.; Aila, T.; and Lehtinen, J. 2022. The Role of ImageNet Classes in Fr\'echet Inception Distance. *ArXiv preprint*, abs/2203.06026.

Li, M.; Qu, T.; Sun, W.; and Moens, M.-F. 2023. Alleviating Exposure Bias in Diffusion Models through Sampling with Shifted Time Steps. *ArXiv preprint*, abs/2305.15583.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35: 5775–5787.

Mittal, G.; Engel, J.; Hawthorne, C.; and Simon, I. 2021. Symbolic music generation with diffusion models. *ArXiv preprint*, abs/2103.16091.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved Denoising Diffusion Probabilistic Models. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 8162–8171. PMLR.

Nichol, A. Q.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 16784–16804. PMLR.

Ning, M.; Sangineto, E.; Porrello, A.; Calderara, S.; and Cucchiara, R. 2023. Input Perturbation Reduces Exposure Bias in Diffusion Models. *ArXiv preprint*, abs/2301.11706.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *ArXiv preprint*, abs/2204.06125.

Rangwani, H.; Bansal, L.; Sharma, K.; Karmali, T.; Jampani, V.; and Babu, R. V. 2023. NoisyTwins: Class-Consistent and Diverse Image Generation through StyleGANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5987–5996.

Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2016. Sequence Level Training with Recurrent Neural Networks. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494.

Salimans, T.; and Ho, J. 2022. Progressive Distillation for Fast Sampling of Diffusion Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Schmidt, F. 2019. Generalization in Generation: A closer look at Exposure Bias. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, 157–167. Hong Kong: Association for Computational Linguistics.

Singer, U.; Polyak, A.; Hayes, T.; Yin, X.; An, J.; Zhang, S.; Hu, Q.; Yang, H.; Ashual, O.; Gafni, O.; et al. 2022. Make-a-video: Text-to-video generation without text-video data. *ArXiv preprint*, abs/2209.14792.

Sohl-Dickstein, J.; Weiss, E. A.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In Bach, F. R.; and Blei, D. M., eds., *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, 2256–2265. JMLR.org.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Venkatraman, A.; Hebert, M.; and Bagnell, J. A. 2015. Improving Multi-Step Prediction of Learned Time Series Models. In Bonet, B.; and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, 3024–3030. AAAI Press.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Shao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2022. Diffusion models: A comprehensive survey of methods and applications. *ArXiv preprint*, abs/2209.00796.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *ArXiv preprint*, abs/1506.03365.

Zhang, W.; Feng, Y.; Meng, F.; You, D.; and Liu, Q. 2019. Bridging the Gap between Training and Inference for Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4334–4343. Florence, Italy: Association for Computational Linguistics.