AC-SAMPLER: ACCELERATE AND CORRECT DIFFU-SION SAMPLING WITH METROPOLIS-HASTINGS ALGO-RITHM

Anonymous authorsPaper under double-blind review

000

001

002

004

006

008 009 010

011 012

013

014

016

017

018

019

021

025

026

027 028 029

030

032

033

034

037

038

040

041

042 043

044

046

047

048

051

052

ABSTRACT

Diffusion-based generative models have recently achieved state-of-the-art performance in high-fidelity image synthesis. These models learn a sequence of denoising transition kernels that gradually transform a simple prior distribution into a complex data distribution. However, requiring many transitions not only slows down sampling but also accumulates approximation errors. We introduce the Accelerator-Corrector Sampler (AC-Sampler), which accelerates and corrects diffusion sampling without fine-tuning. It generates samples directly from intermediate timesteps using the Metropolis-Hastings (MH) algorithm while correcting them to target the true data distribution. We derive a tractable density ratio for arbitrary timesteps with a discriminator, enabling computation of MH acceptance probabilities. Theoretically, our method yields samples better aligned with the true data distribution than the original model distribution. Empirically, AC-Sampler achieves FID 2.38 with only 15.8 NFEs, compared to the base sampler's FID 3.23 with 17 NFEs on unconditional CIFAR-10. On CelebA-HQ 256×256, it attains FID 6.6 with 98.3 NFEs. AC-Sampler can be combined with existing acceleration and correction techniques, demonstrating its flexibility and broad applicability.

1 Introduction

Diffusion-based generative models (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song et al., 2021a) have become one of the most popular approaches in recent years due to their strong ability to generate diverse types of data such as high-fidelity images (Dhariwal & Nichol, 2021; Rombach et al., 2022) and videos (Ho et al., 2022; Voleti et al., 2022). Building on these models, strong pre-trained variants have emerged (Rombach et al., 2022; Karras et al., 2022), followed by many studies aiming to further improve them (Kim et al., 2023; Na et al., 2024; Xu et al., 2023a). In spite of many variations, these models share a fundamental structure: they start by sampling from a simple prior distribution and iteratively transform the samples through a series of learned transition kernels to approximate the complex data distribution. This iterative generation causes two problems. First, the sampling process is slow due to the large number of kernel transition calculations required (Song et al., 2021b; Zhang & Chen, 2023). Second, errors can accumulate during the sampling process if the kernel transition does not accurately reflect the true reverse diffusion process (Xu et al., 2023a).

Speed and accuracy are usually considered separate research topics, as improving both simultaneously is often challenging. Some approaches for acceleration diffusion sampling (Kim & Ye, 2023; Zheng et al., 2023) focus on reducing the NFE while maintaining image quality, but they lack theoretical analysis for converging to the true data distribution. On the contrary, previous correction methods (Kim et al., 2023; Na et al., 2024; Xu et al., 2023a) maintain or even increase the NFE, which makes the methods unscalable in real-world services.

To address both challenges, we propose **Accelerator-Corrector Sampler (AC-Sampler)**. It accelerates and corrects the diffusion sampling process without any fine-tuning of the pre-trained model. Instead of sampling from the prior distribution, AC-Sampler directly proposes samples at intermediate timesteps, which enables acceleration. Using Metropolis-Hastings correction, these proposals are guaranteed to theoretically follow the true marginal distribution. Since the pre-trained model approximates the score function, we can construct an effective proposal distribution using

Langevin dynamics (Grenander & Miller, 1994). For computing the acceptance probability, we only train a time-dependent discriminator (Kim et al., 2023; Na et al., 2024), which can be learned at a much lower cost than the diffusion model itself.

Our contributions are as follows:

- We propose the **AC-Sampler**, which accelerates diffusion sampling by generating samples from intermediate timesteps rather than the initial prior distribution. It also corrects the accumulated error in the sampling process with Metropolis-Hastings (MH) algorithms.
- We provide a theoretical analysis showing that training the discriminator and following the MH chain in our method leads to a tighter bound on the data distribution compared to that of a pre-trained diffusion model. Furthermore, we provide a theoretic analysis of the expected reduction in the number of function evaluations (NFE).
- We validate our theoretical claim through experiments on benchmark datasets and toy settings, and demonstrate that our method applies effectively to diverse pre-trained models in both unconditional and conditional settings.
- Our contribution is orthogonal to advances in training-free samplers, so the two gains are complementary and can be realized simultaneously. Also, the utilize discriminator is simple and does not require ad-hoc structures on the pretrained diffusion model.

2 RELATED WORK

Due to the high cost of training or fine-tuning pre-trained diffusion models, a growing body of work focuses on keeping such models fixed and instead accelerating and correcting the sampling process.

Acceleration Methods To reduce the computational burden associated with additional training, acceleration methods have been developed to speed up the sampling process without modifying the original diffusion model. DDIM (Song et al., 2021a) reformulates the reverse diffusion as a deterministic ODE, achieving significant speedups with fewer steps while preserving the pre-trained network. Building on this foundation, various works have further improved ODE solvers through high-order numerical methods and exponential integration, leading to significant gains in sampling efficiency (Liu et al., 2022; Lu et al., 2022a;b; Dockhorn et al., 2022a; Karras et al., 2022; Zhang & Chen, 2023; Zhao et al., 2023; Zheng et al., 2023).

In parallel, other lines of work have explored fundamentally different perspectives on diffusion model acceleration. For example, PDS (Ma et al., 2022) treats diffusion sampling as an Markov Chain Monte Carlo (MCMC) process, incorporating frequency-domain preconditioning to improve high-frequency details. DLG (Kim & Ye, 2023) formulates the sampling process over the product space of data and time, enabling joint Langevin-based Gibbs sampling. This sampling process identifies the intermediate perturbed data with low noise for the initialization of the reverse process, which shortens the subsequent diffusion trajectory. However, as shown in Appendix A.1, DLG lacks theoretical convergence guarantees, leaving room for improvement.

Correction Methods Several studies have been conducted to improve the sampling quality of pre-trained diffusion models. DG (Kim et al., 2023) proposes a correction method using a time-dependent discriminator when score estimation is inaccurate, thereby improving the accuracy of the transition kernel. Restart (Xu et al., 2023a) theoretically demonstrates that repeating forward and backward steps within a fixed time interval $[t_{\min}, t_{\max}]$ in a pre-trained model can reduce sampling error. DiffRS (Na et al., 2024) aims to sample from the true distribution by applying a rejection sampling scheme with a time-dependent discriminator. ES (Ning et al., 2024) proposes a training-free correction schedule to compensate for the scale gap in score norms between the training and sampling phases. However, there is no theoretical guarantee that simply matching the norms leads to distributional equivalence. While these methods focus on sampling correction, they do not reduce the base NFE and may even increase it, leading to slower sampling.

3 PRELIMINARY

3.1 METROPOLIS-HASTINGS ALGORITHM AND LANGEVIN PROPOSAL

The Metropolis-Hastings algorithm (MH algorithm) (Metropolis et al., 1953; Hastings, 1970) is a MCMC method used to sample from a target distribution when direct sampling is not possible. It constructs a Markov chain whose stationary distribution is the target distribution by satisfying the detailed balance condition. Based on this condition, the algorithm defines an acceptance probability, which is used to determined whether to accept proposals drawn from a simple proposal distribution.

Given a target distribution $q(\cdot)$ and a proposal distribution $p_{\text{proposal}}(\cdot \mid \cdot)$, the acceptance probability for a proposed sample $\tilde{\mathbf{x}}$ and current sample \mathbf{x} is defined as:

$$\alpha = \min\left(1, \frac{q(\tilde{\mathbf{x}}) p_{\text{proposal}}(\mathbf{x} \mid \tilde{\mathbf{x}})}{q(\mathbf{x}) p_{\text{proposal}}(\tilde{\mathbf{x}} \mid \mathbf{x})}\right), \tag{1}$$

which guarantees that $q(\cdot)$ is the stationary distribution of the Markov chain.

Though the MH algorithm allows to sample from a complex distribution, a poor proposal distribution leads to slow convergence to the target distribution. To improve mixing and convergence, gradient-based proposals have been studied (Parisi, 1988; Neal et al., 2011), i.e. the Langevin proposal, which leverages the gradient of the target distribution. Specifically, the proposal is derived from the Euler–Maruyama discretization of the overdamped Langevin dynamics (Roberts & Tweedie, 1996). Formally, with the target distribution $q(\mathbf{x})$, the Langevin proposal is defined as:

$$\tilde{\mathbf{x}} = \mathbf{x} + \frac{\eta}{2} \nabla_{\mathbf{x}} \log q(\mathbf{x}) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \tag{2}$$

where η is the step size and $\nabla_{\mathbf{x}} \log q(\mathbf{x})$ is the score function of $q(\mathbf{x})$. This method is commonly referred to as the Metropolis-Adjusted Langevin Algorithm (MALA) (Grenander & Miller, 1994).

3.2 Diffusion Models

Diffusion models (Ho et al., 2020; Song et al., 2021b) are probabilistic generative models that approximate data distributions by adding noise and reversing this process. They consist of a forward process that corrupts data into noise and a reverse process that removes noise to generate samples.

Let $q_0(\mathbf{x}_0)$ denote the true data distribution and $p_0^{\boldsymbol{\theta}}(\mathbf{x}_0)$ denote the distribution of generated samples from the model. In particular, the forward process is a fixed Markov chain where Gaussian noise is added using a pre-defined variance schedule. This creates a sequence of random variables $\mathbf{x}_{1:T}$:

$$q_{1:T|0}(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q_{t|t-1}(\mathbf{x}_t|\mathbf{x}_{t-1}), \tag{3}$$

where each $q_{t|t-1}(\mathbf{x}_t|\mathbf{x}_{t-1})$ is a Gaussian transition with increasing noise levels. This process transforms the data distribution into a tractable prior distribution (e.g., standard Gaussian) as $t \to T$. The reverse process is modeled as a Gaussian distribution that denoises a prior distribution $p_T(\mathbf{x}_T)$ iteratively:

$$p_{0:T}(\mathbf{x}_{0:T}) := p_T(\mathbf{x}_T) \prod_{t=1}^T p_{t-1|t}^{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t), \tag{4}$$

where $p_{t-1|t}^{\theta}$ is the transition kernel that generate data from prior distribution p_T . The model is trained by maximizing a variational bound on the log-likelihood of the data. In practice, this can be achieved via denoising score matching loss (Vincent, 2011; Ho et al., 2020), given by:

$$\mathcal{L}_{\text{DSM}}(\boldsymbol{\theta}) = \mathbb{E}_{t \sim U[0,1], \mathbf{x}_0 \sim q(\mathbf{x}_0), \mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_0)} \left[\left\| \mathbf{s}^{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q_{t|0}(\mathbf{x}_t | \mathbf{x}_0) \right\|_2^2 \right], \tag{5}$$

With this loss, $\mathbf{s}^{\theta}(\mathbf{x}_t, t)$ optimizes to follow the true score $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$. The transition kernel is parameterized as a Gaussian distribution whose mean is a function of the score function:

$$p_{t-1|t}^{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_t(\mathbf{x}_t, \mathbf{s}^{\boldsymbol{\theta}}(\mathbf{x}_t, t)), \sigma_t^2 \mathbf{I}),$$
(6)

where $\sigma_t^2 \mathbf{I}$ denotes time-dependent variance. After training, samples are generated by iteratively applying the reverse kernel from t = T to t = 0.

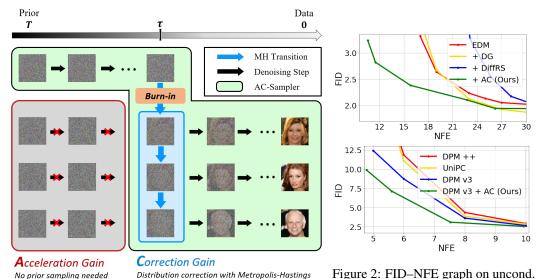


Figure 1: Overall figure of AC-Sampler.

Figure 2: FID–NFE graph on uncond. CIFAR-10: (Top) Correction methods (Bottom) Acceleration methods.

We assume that we have access to a pre-trained diffusion model¹, denoted by $\boldsymbol{\theta}$. Let $q_t(\mathbf{x}_t)$ and $p_t^{\boldsymbol{\theta}}(\mathbf{x}_t)$ denote the marginal distributions at timestep t, defined by forward diffusion processes starting from $q_0(\mathbf{x}_0)$ and $p_0^{\boldsymbol{\theta}}(\mathbf{x}_0)$, respectively. Pre-trained diffusion model provides $\mathbf{s}^{\boldsymbol{\theta}}(\mathbf{x}_t,t) \approx \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$. The mean of the transition kernel depends on both \mathbf{x}_t and the score function $\mathbf{s}^{\boldsymbol{\theta}}(\mathbf{x}_t,t)$. Therefore, we treat the score function and the transition kernel $p_{t-1|t}^{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ as equivalent parametrizations, since both are derived from the same model.

Due to this structure, diffusion models are inherently limited by slow sampling speed from the large number of transition steps, and by the accumulation of approximation errors in the transition kernels.

4 METHOD

To address the two key challenges previously discussed, we propose **AC-Sampler**, a novel diffusion sampling framework. The overall sampling procedure is as follows. First, we perform denoising from the prior distribution up to a target timestep τ , which serves as the initial sample of the MCMC chain. Starting from this initial sample, we repeatedly generate new candidates using a score-based proposal distribution. At each step, MH correction is applied. These steps are performed as described in Algorithm. 1. After sufficient burn-in period, the resulting samples are corrected to true marginal distribution, q_{τ} . Finally, each accepted sample is further denoised to obtain the final outputs. The overall sampling process is illustrated in Fig. 1.

This approach has two advantages. First, samples are created directly at τ without denoising from T, which enables faster sampling (denoted as Acceleration~Gain in Fig. 1). Second, because of MH correction, the resulting samples follow a true marginal distribution, which yields more accurate samples(denoted as Correction~Gain in Fig. 1). Our method accelerates and corrects the sampling process without requiring any fine-tuning of the underlying diffusion model. To implement this process, we require the design of the proposal distribution (Sec. 4.1) and the computation of the acceptance probability (Sec. 4.2).

4.1 PROPOSAL DISTRIBUTION

To sample from the intermediate timestep t, we use Metropolis-Adjusted Langevin Algorithm (MALA). We set the target distribution to be $q_t(\mathbf{x}_t)$, and construct the proposal distribution using its score function. Since pre-trained score model $\mathbf{s}^{\theta}(\mathbf{x}_t,t)$ approximates the score $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$, we

¹This assumption reflects realistic applications, since many pre-trained diffusion models can readily incorporate AC-Sampler without retraining.

Algorithm 1 MALAOneStep

Input: Target timestep τ , Previous sample \mathbf{x}_{τ} , Score output $\mathbf{s} := \mathbf{s}^{\theta}(\mathbf{x}_{\tau}, \tau)$, Likelihood ratio $L_{\tau}^{\phi} := \frac{d^{\phi}(\mathbf{x}_{\tau}, \tau)}{1 - d^{\phi}(\mathbf{x}_{\tau}, \tau)}$, Score network \mathbf{s}^{θ} , Discriminator d^{ϕ}

Output: Next sample $\tilde{\mathbf{x}}_{\tau}$

1: repeat

- 2: Propose $\tilde{\mathbf{x}}_{\tau}$ from proposal distribution $p_{\text{proposal},\tau}^{\boldsymbol{\theta}}(\cdot \mid \mathbf{x}_{\tau})$ (Eq. 7)
- 3: Get score $\tilde{\mathbf{s}} \leftarrow \mathbf{s}^{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{\tau}, \tau)$, and likelihood ratio $\tilde{L}_{\tau}^{\boldsymbol{\phi}} \leftarrow \frac{d^{\boldsymbol{\phi}}(\tilde{\mathbf{x}}_{\tau}, \tau)}{1 d^{\boldsymbol{\phi}}(\tilde{\mathbf{x}}_{\tau}, \tau)}$
- 4: Calculate acceptance probability $\alpha \leftarrow \hat{\alpha}(\mathbf{x}_{\tau}, \tilde{\mathbf{x}}_{\tau}, \mathbf{s}, \tilde{\mathbf{s}}, L_{\tau}^{\phi}, \tilde{L}_{\tau}^{\phi})$ (Eq. 11)
- 5: Sample $u \sim \mathcal{U}(0,1)$
- 6: **until** $u < \alpha$
- 7: **return** $\tilde{\mathbf{x}}_{\tau}, \tilde{\mathbf{s}}, \tilde{L}_{\tau}^{\boldsymbol{\phi}}$

leverage $\mathbf{s}^{\theta}(\mathbf{x}_t, t)$ to construct the proposal distribution as follows:

$$p_{\text{proposal},t}^{\boldsymbol{\theta}}(\cdot|\mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_t + \frac{\eta}{2}\mathbf{s}^{\boldsymbol{\theta}}(\mathbf{x}_t, t), \eta \mathbf{I}\right), \tag{7}$$

where η is Langevin step size. We adaptively set the value of η to maintain a constant signal-to-noise ratio (SNR) during sampling. An important advantage of our framework is that both the denoising step and the MCMC-based proposal distribution rely on the same score value. As a result, a single network evaluation is sufficient for both operations, enabling efficient integration of denoising and exploration within AC-Sampler.

4.2 ACCEPTANCE PROBABILITY

The target distribution is $q_t(\cdot)$, so the acceptance probability is $\alpha = \min\left(1, \frac{q_t(\tilde{\mathbf{x}}_t)p_{\text{proposal},t}^{\boldsymbol{\theta}}(\mathbf{x}_t|\tilde{\mathbf{x}}_t)}{q_t(\mathbf{x}_t)p_{\text{proposal},t}^{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_t|\mathbf{x}_t)}\right)$,

where $\tilde{\mathbf{x}}_t$ is a sample from the proposal distribution $p_{\text{proposal},t}^{\theta}(\cdot|\mathbf{x}_t)$. To make the acceptance probability tractable, we first decompose $q_t(\tilde{\mathbf{x}}_t)/q_t(\mathbf{x}_t)$ as stated in the following theorem.

Theorem 4.1. Let \mathbf{x}_t and $\tilde{\mathbf{x}}_t$ be two arbitrary samples at diffusion timestep t. Then, for any fixed \mathbf{x}_{t-1} , the density ratio of the true marginal distribution q_t is given by:

$$\frac{q_t(\tilde{\mathbf{x}}_t)}{q_t(\mathbf{x}_t)} = \frac{q_{t|t-1}(\tilde{\mathbf{x}}_t \mid \mathbf{x}_{t-1})}{q_{t|t-1}(\mathbf{x}_t \mid \mathbf{x}_{t-1})} \cdot \frac{L_t(\tilde{\mathbf{x}}_t, t)}{L_t(\mathbf{x}_t, t)} \cdot \frac{p_{t-1|t}^{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{p_{t-1|t}^{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_t)}, \tag{8}$$

where $L_t(\mathbf{x}_t, t) := q_t(\mathbf{x}_t)/p_t^{\theta}(\mathbf{x}_t)$ denotes the likelihood ratio between the data and model marginal distributions at timestep t.

Proof is provided in Appendix A.3. Let $\mu_t(\mathbf{x}_t, \mathbf{s}^{\theta}(\mathbf{x}_t, t))$ denote the mean of the reverse transition kernel $p_{t-1|t}^{\theta}$. Since Theorem 4.1 holds for arbitrary \mathbf{x}_{t-1} , choose

$$\hat{\mathbf{x}}_{t-1} := \frac{1}{2} \left(\boldsymbol{\mu}_t(\mathbf{x}_t, \mathbf{s}^{\boldsymbol{\theta}}(\mathbf{x}_t, t)) + \boldsymbol{\mu}_t(\tilde{\mathbf{x}}_t, \mathbf{s}^{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_t, t)) \right).$$

With this choice, the transition kernel related terms in Eq. 8 are $\frac{p_{t-1|t}^{\theta}(\hat{\mathbf{x}}_{t-1}|\mathbf{x}_t)}{p_{t-1|t}^{\theta}(\hat{\mathbf{x}}_{t-1}|\tilde{\mathbf{x}}_t)}$, which cancel in the density ratio because both kernels are Gaussian with the same variance and $\hat{\mathbf{x}}_{t-1}$ is equidistant from their means. We therefore obtain the acceptance probability as follows:

$$\alpha = \min \left(1, \frac{q_{t|t-1}(\tilde{\mathbf{x}}_t \mid \hat{\mathbf{x}}_{t-1})}{q_{t|t-1}(\mathbf{x}_t \mid \hat{\mathbf{x}}_{t-1})} \cdot \frac{L_t(\tilde{\mathbf{x}}_t, t)}{L_t(\mathbf{x}_t, t)} \cdot \frac{p_{\text{proposal}, t}^{\boldsymbol{\theta}}(\mathbf{x}_t \mid \tilde{\mathbf{x}}_t)}{p_{\text{proposal}, t}^{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_t \mid \mathbf{x}_t)} \right)$$
(9)

To access $L_t(\mathbf{x}_t)$, we use time-dependent discriminator d^{ϕ} , following the approach of DG (Kim et al., 2023). The discriminator is trained to distinguish between q_t and p_t^{θ} at all timesteps. To achieve this, weighted binary cross-entropy loss is used for training the discriminator:

$$\mathcal{L}_{BCE}(\boldsymbol{\phi}) = \int \lambda(t) \left[\mathbb{E}_{\mathbf{x}_t \sim q_t} \left[-\log d^{\boldsymbol{\phi}}(\mathbf{x}_t, t) \right] + \mathbb{E}_{\mathbf{x}_t \sim p_t^{\boldsymbol{\theta}}} \left[-\log(1 - d^{\boldsymbol{\phi}}(\mathbf{x}_t, t)) \right] \right] dt, \quad (10)$$

The optimal discriminator satisfies $d^{\phi^*}(\mathbf{x}_t,t) = \frac{q_t(\mathbf{x}_t)}{q_t(\mathbf{x}_t) + p_t^{\theta}(\mathbf{x}_t)}$, so the density ratio $\frac{q_t(\mathbf{x}_t)}{p_t^{\theta}(\mathbf{x}_t)}$ becomes $\frac{d^{\phi^*}(\mathbf{x}_t,t)}{1-d^{\phi^*}(\mathbf{x}_t,t)}$. Having access to $\frac{q_t(\mathbf{x}_t)}{p^{\theta}(\mathbf{x}_t)} \approx \frac{d^{\phi}(\mathbf{x}_t,t)}{1-d^{\phi}(\mathbf{x}_t,t)} =: L_t^{\phi}(\mathbf{x}_t,t)$, the acceptance probability can be

$$\hat{\alpha}(\mathbf{x}_{t}, \tilde{\mathbf{x}}_{t}, \mathbf{s}, \tilde{\mathbf{s}}, L, \tilde{L}) = \min \left(1, \underbrace{\frac{q_{t|t-1}(\tilde{\mathbf{x}}_{t} \mid \hat{\mathbf{x}}_{t-1})}{q_{t|t-1}(\mathbf{x}_{t} \mid \hat{\mathbf{x}}_{t-1})}}_{\text{Forward term}} \cdot \underbrace{\frac{\tilde{L}}{L}}_{\text{Likelihood ratio}} \cdot \underbrace{\frac{p_{\text{proposal}, t}^{\boldsymbol{\theta}}(\mathbf{x}_{t} \mid \tilde{\mathbf{x}}_{t})}{p_{\text{proposal}, t}^{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t} \mid \mathbf{x}_{t})}}_{\text{Proposal term}} \right)$$
(11)

where $\mathbf{s}, \tilde{\mathbf{s}}, L, \tilde{L}$ denotes $\mathbf{s}^{\theta}(\mathbf{x}_t, t), \mathbf{s}^{\theta}(\tilde{\mathbf{x}}_t, t), L_t^{\phi}(\mathbf{x}_t, t), L_t^{\phi}(\tilde{\mathbf{x}}_t, t)$, respectively. The acceptance probability consists of three terms. The forward and proposal terms are tractable Gaussian distributions, and the likelihood ratio is computed using a discriminator. Together, these components make the acceptance probability fully tractable. Note that with any tractable proposal distribution, the acceptance probability also remains tractable.

4.3 THEORETICAL ANALYSIS

We prove that our method can theoretically achieve sampling acceleration and correction.

Proposition 4.2. Let the reverse diffusion process have total timestep T and the AC-sampler target timestep be τ . Let NFE_R be the average NFE reduction per sample with AC-Sampler. If the acceptance probability satisfies $\alpha > \frac{1}{T-\tau+1}$, then, $\lim_{l\to\infty} \mathbb{E}[NFE_R] > 0, \quad \lim_{l\to\infty} \mathrm{Var}(NFE_R) = 0$

$$\lim_{l \to \infty} \mathbb{E}[NFE_R] > 0, \quad \lim_{l \to \infty} \text{Var}(NFE_R) = 0 \tag{12}$$

where l is the length of Markov Chain at timestep τ

Since, in practice, $T-\tau$ is pretty big, the threshold $\alpha>\frac{1}{T-\tau+1}$ is mild and typically easy to satisfy. Moreover, because our proposals are guided by the score network, acceptance rates are sufficiently high that substantial NFE reduction can be achieved even with a small chain length l. Further discussion of the acceptance probability is provided in Appendix C.

To show that our method not only accelerates sampling but also corrects errors, we theoretically demonstrate that the data distribution induced by our sampler is closer to the true data distribution than that of the baseline model.

Theorem 4.3. Let p_0^{θ} , p_0^{θ,ϕ^*} denote the model distribution and refined distribution by AC-Sampler with optimal discriminator ϕ^* , respectively. Then, the KL divergence between the true data distribution q_0 and the refined distribution p_0^{θ,ϕ^*} is bounded by:

$$D_{KL}(q_0(\mathbf{x}_0)||p^{\boldsymbol{\theta},\boldsymbol{\phi}^*}(\mathbf{x}_0)) \le D_{KL}(q_0(\mathbf{x}_0)||p^{\boldsymbol{\theta}}(\mathbf{x}_0))$$
(13)

Theorem 4.4. Let T_{τ} be the transition kernel of MALA at timestep τ . Also, $p_{\tau}^{\theta,\phi^*,(l)}$ denotes marginal distribution at timestep τ after the l-th MALA transition from p_{τ}^{θ} , and $p_{0}^{\theta,\phi^{*},(l)}$ denotes the data distribution generated from $p_{\tau}^{\theta,\phi^*,(l)}$ with denoising transition kernel $p_{t-1|t}^{\theta}$. \mathcal{L}^p denotes a space of function which satisfies $(\int_{\mathbb{R}} |f|^p d\mathbf{x})^{\frac{1}{p}} < \infty$. If $q_{\tau}(\mathbf{x}_{\tau}) \in \mathcal{L}^{\alpha}$, $\log\left(\frac{p_{\tau}^{\theta, \phi^*, (l)}(\mathbf{x}_{\tau})}{q_{\tau}(\mathbf{x}_{\tau})}\right) \in \mathcal{L}^{\beta}$, $T_{\tau} \in \mathcal{L}^{\gamma}$, where $\alpha, \beta, \gamma \in [1, \infty]$ satisfy $\frac{1}{\alpha} + \frac{1}{\beta} + \frac{1}{\gamma} = 1$, then the KL divergence between the true data distribution q_0 and the refined distribution $p_0^{\theta, \phi^*, (l+1)}$ is bounded by:

$$D_{KL}(q_0(\mathbf{x}_0)||p_0^{\boldsymbol{\theta},\phi^*,(l+1)}(\mathbf{x}_0)) \le D_{KL}(q_0(\mathbf{x}_0)||p_0^{\boldsymbol{\theta},\phi^*,(l)}(\mathbf{x}_0))$$
(14)

The proofs of each theoretical result and detailed analysis are provided in Appendix A.4 and A.5, respectively. When our model has sufficiently converged, Theorem 4.3 suggests that it can generate samples that are closer to the true data distribution than those produced by the baseline model. Theorem 4.4 further shows that applying more MALA steps progressively moves the samples closer to the true data distribution. Note that l=0 denotes the base diffusion model, without MALA.

Table 1: Performance on unconditional CIFAR-10 generation. Values that are better compared to the baseline are highlighted in bold.

Model	FID↓	NFE↓
Unconditional Generation		
VDM (Kingma et al., 2021)	7.41	1000
DDPM (Ho et al., 2020)	3.17	1000
iDDPM (Nichol & Dhariwal, 2021)	2.90	1000
DDIM (Song et al., 2021a)	4.16	100
ScoreSDE (Song et al., 2021b)	2.20	2000
Soft Truncation (Kim et al., 2022b)	2.33	2000
STF (Xu et al., 2022)	1.90	35
CLD-SGM (Dockhorn et al., 2022b)	2.25	312
INDM (Kim et al., 2022a)	2.28	2000
LSGM (Vahdat et al., 2021)	2.10	138
PFGM++ (Xu et al., 2023b)	1.93	35
PSLD (Pandey & Mandt, 2023)	2.10	246
Flow Matching (Lipman et al., 2023)	6.35	142
Rectified Flow (Liu et al., 2023)	2.58	127
ES (Ning et al., 2024)	1.95	35
EDM (Heun) (Karras et al., 2022)	2.01	35
EDM (Heun) + AC (Ours)	1.97	26.19
DDO (Heun)(Zheng et al., 2025)	1.42	35
DDO + AC (Ours)	1.41	29.41
Acceleration Method		
DPM ++ (Lu et al., 2022a)	24.54	5
UniPC (Zhao et al., 2023)	23.52	5
DPM-v3 (Zheng et al., 2023)	12.41	5
DPM-v3 + AC (Ours)	9.88	4.78
Correction Method		
Restart (Xu et al., 2023a)	1.95	43
DiffRS (Na et al., 2024)	2.02	30.73
DG (Kim et al., 2023)	1.93	27
DG + AC (Ours)	1.84	26.19

Table 2: Performance on unconditional CIFAR-10 generation with (Top) correction and (Bottom) acceleration methods.

Method	FID↓	NFE↓	FID↓	NFE↓	FID↓	NFE↓
EDM (Heun)	2.05	27	2.23	23	3.23	17
+DiffRS (Na et al., 2024)	2.17	28.15	3.26	23.13	7.79	19.87
+DG (Kim et al., 2023)	1.93	27	2.12	23	3.62	17
+AC (Ours)	1.97	26.19	2.10	22.78	2.38	15.81
Method	FID	↓ NFE↓	. FID↓	NFE↓	FID↓	NFE↓
Method DPM++ (Lu et al., 2022b)	FID:	•	. FID↓	NFE↓	FID↓ 2.91	NFE↓ 10
	+	5 6	+ -			•
DPM++ (Lu et al., 2022b)	11.8	5 6 0 6	4.36	8	2.91	10

Table 3: FID and NFE on unconditional CelebA-HQ 256 generation.

Method	FID↓	NFE↓	FID↓	NFE↓	FID↓	NFE↓
ScoreSDE (KAR1)	121.27	40	122.74	98	125.15	198
+DLG (Kim & Ye, 2023)	20.19	21.21	29.12	47.21	30.72	107.21
+AC (Ours)	15.13	15.94	22.55	44.06	15.69	87.26
ScoreSDE (KAR2)	83.21	40	57.28	98	29.74	198
+DLG (Kim & Ye, 2023)	17.92	21.21	12.12	47.21	8.14	107.21
+AC (Ours)	8.45	20.05	9.55	40.05	6.60	98.27

Table 4: Performance on conditional ImageNet (Top) 64×64, (Bottom) 256×256 generation.

Method	FID↓	$NFE\!\!\downarrow$	FID↓	NFE	↓ FID↓	$NFE\!\!\downarrow$
EDM (SDE)	2.30	61	1.78	127	1.43	511
+AC (Ours)	2.25	58.75	1.77	121.9	8 1.42	483.86
Method			FID↓	NFE↓	Precision [†]	Recall†
Method DiT (DDPM) (Pee	bles & X	ie, 2023)		NFE↓ 250	Precision↑ 0.829	Recall↑ 0.576

5 EXPERIMENTS

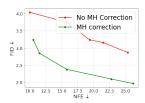
In this section, we present experimental results to validate the effectiveness of our method empirically.

Experimental setting We employ our methods on various pre-trained networks trained on CIFAR-10 (Krizhevsky, 2009), ImageNet64×64 and 256×256 (Deng et al., 2009), CelebA-HQ 256×256 (Karras et al., 2017). We report the Fréchet Inception Distance (FID) (Heusel et al., 2017), Precision/Recall (Kynkäänniemi et al., 2019). We use Heun, SDE sampler Karras et al. (2022), KAR1(deterministic), KAR2(stochastic) sampler (Kim & Ye, 2023), and DDPM sampler (Ho et al., 2020). We highlight the best-performing results compare to baseline model in bold. Detailed hyperparameters and experimental settings are provided in Appendix C and G.

5.1 Unconditional Generation

CIFAR-10 The upper part of Table 1 shows results for unconditional generation on CIFAR-10. Our method is compatible with both EDM and DDO checkpoints. Both methods were re-tested without applying seed fixing as in the EDM setting. For the EDM checkpoint, our sampler improves the FID from 2.01 with 35 NFE to 1.97 with only 26.19 NFE. For the stronger DDO checkpoint, we achieve 1.41 FID while reducing NFE from 35 to 29.41. Although the gain in FID is marginal for highly capable pre-trained models, our method consistently reduces the NFE, demonstrating its efficiency.

CelebA-HQ 256×**256** We employ the pre-trained time classifier released by DLG to reproduce their reported performance. For the AC-Sampler, we extend the MALA algorithm in the joint space of $(\mathbf{x}_{\tau}, \tau)$, as detailed in the Appendix B, to support effective sampling in high-dimensional benchmark



379

380

381

382

384

385

386

387 388

389

390 391

392

393

394

395

396

397

398

399 400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415 416 417

418

419

420

421

422

423

424 425

426

427

428

429

430

431

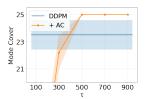


Table 5: Performance of AC-Sampler apply on EDM. Time denotes the wall-clock seconds required to generate 100 images.

Figure 3: Effect of an Figure 4: Mode cover MH correction on AC-Sampler.

with different τ in 25-Gaussian toy experiment

Method	FID↓	NFE↓	Precision [†]	Recall↑	Time↓
EDM (Heun) +AC	2.01 1.97	35 26.19	0.704 0.703	0.627 0.628	6.46 5.26
EDM (Heun)	2.24	23	0.703	0.625	4.30
+AC	2.21	20.67	0.707	0.632	4.15
EDM (Heun) +AC	3.32 2.41	17 15.78	0.683 0.699	0.622 0.623	3.20 3.19
TAC	2.71	15.70	0.077	0.023	3.17

datasets. As shown in Table 3, the AC-Sampler demonstrates a clear improvement in FID compared to other methods with lower NFE.

Method Compatibility The lower part of Table 1 reports results when applying AC-Sampler on top of existing acceleration and correction methods. When combined with DPM-v3, which originally yields 12.41 FID at 5 NFEs, our method improves performance to 9.88 FID with only 4.78 NFEs. Similarly, DG achieves 1.93 FID at 27 NFEs, while AC-Sampler applied to DG further reduces this to 1.84 FID at 26.19 NFEs. These experiments highlight that our method is orthogonal to existing acceleration and correction methods, and can flexibly enhance them. Table 2 and Figure 2 present the FID-NFE trade-offs of our method compared to existing acceleration and correction techniques. We observe that our method achieves better trade-offs in most NFE regimes.

5.2 CLASS-CONDITIONAL GENERATION

ImageNet Table 4 presents the results on conditional ImageNet 64×64 and 256×256 generation. For ImageNet 64×64, we employed a conditional score network, while for ImageNet 256×256, we used classifierfree guided generation with a CFG scale of 1.5. In both cases, we find effective improvements in NFE and slight improvements in FID. Although class-conditioned settings inherently limit the length of the MCMC chain, our results demonstrate that the pro-



Figure 5: AC-Sampler on CelebA-HQ 256×256 (Top, unconditional) and ImageNet 256×256 (Bottom, condition on "Weasel"). Panel presents the final results of the MALA chains, ordered from left to right.

posed method can still be applied effectively under this constraint. We provide a further discussion of the class diversity and related experiments in Appendix F.2.

5.3 ABLATION STUDIES

Distribution Alignment In our method, the acceptance probability is computed using the density ratio provided by the discriminator. If this density ratio is inaccurate, effective distribution alignment through MH correction becomes difficult. To examine this, we retained the overall framework of our method but removed the MH accept-reject step. As shown in Fig. 3, without MH correction, the FID degrades significantly. This result demonstrates that our method effectively corrects the distribution through MH correction.

Sample Diversity Because our method relies on MH, successive samples can be correlated. Thus, assessing whether it still produces diverse samples is crucial. In Table 5, the recall metric, which is an indicator of sample diversity, is comparable to the base sampler. These results indicate that, our method can preserve sample diversity. Together with the improved FID and reduced NFE, these results demonstrate both the effectiveness and efficiency of our approach. The MALA chains on benchmark datasets are shown in Fig. 5, where we observe that our method successfully generates diverse and high-quality images. Further analysis is provided in Appendix D.

Toy experiment We applied our method to a 25-Gaussian toy task: we generated 100 samples and define a mode covered if any sample lay within 2σ of its mean. Over 10 trials, the baseline DDPM covered on average 23.5 modes, while ours consistently covered all 25 modes as in Fig. 4. The solid lines indicate the mean, and the shaded areas are the standard deviation. This confirms that our method effectively performs distribution correction and ensures diverse mode coverage. When low- τ regimes, more samples are required for sufficient cover. Fig. 6 shows the results with 100 generated samples. Red circles denote uncovered modes. Ours reduces NFE while fully covering all modes of the 25-Gaussian mixture.



(a) DDPM (NFE=1000) (b) +AC (NFE=504.5)

Figure 6: Toy experiment results. Red circles denote uncovered mode.

Hyperparameter We primarily tune two parameters: the target timestep τ and the signal-to-noise ratio (SNR) of the Langevin proposal. The SNR controls the proposal step size η . Larger SNR yields larger steps. With T=18, we vary τ and SNR and report the resulting FID, NFE and Recall. As τ decreases (i.e., closer to the data distribution), the distribution becomes sharper, and MALA mixing deteriorates. Also, if the SNR is too small, proposals change little from the current sample, also leading to slow exploration. Given the constraints of limited number of generation, the choice of the target timestep τ and the proposal SNR is crucial. A detailed analysis of hyperparameter is provided in Appendix C.

Нуре	rparameter	Metrics				
τ	SNR	FID ↓	NFE↓	Recall ↑		
	0.10	2.89	25.13	0.550		
13	0.20	2.06	25.65	0.620		
	0.23	1.97	26.19	0.628		
	0.10	8.77	15.21	0.200		
8	0.20	6.14	15.75	0.422		
	0.23	6.73	16.70	0.441		
	0.10	62.67	5.29	0		
3	0.20	46.60	5.75	0		
	0.23	43.96	6.26	0		

Table 6: Hyperparameter analysis.

Faster sampling: Wall-clock time Since our method employs an additional discriminator at inference time, the wall-clock time could potentially be slower even with the same NFE. To evaluate this, we measured the average time (in seconds) required to generate 100 samples using both the base sampler and our method on a single RTX 3090. As shown in Table 5, our approach not only improved sample quality but also achieved faster wall-clock time. Unlike conventional approaches that reduce NFE by enlarging the time step and thereby increasing discretization error, our method generates intermediate samples without coarsening the time grid. As a result, the improvement in sample quality is particularly pronounced in the low-NFE regime.

Jump Markov Chain In conventional Metropolis-Hastings, rejected proposals are also retained as part of the chain, which ensures detailed balance but is inefficient when the goal is sample generation under limited capacity. As shown in Table 7, this often manifests as reduced class diversity, making the standard formulation impractical in generative settings. To address this, in Algorithm 1 we adopt a propose-until-accept design: proposals are repeatedly drawn until one is accepted, and only the accepted

Table 7: Comparison of conventional MH (MH $_C$) and Alg. 1.

Method	FID↓	NFE↓	Recall†
EDM (Base)	2.05	27	0.627
+AC with MH _C	3.22	25.08	0.580
+AC with Alg. 1	1.97	26.19	0.628

sample is recorded. This prevents duplicate retention of the same sample and promotes greater diversity in the generated outputs. This variant can be formally understood as a Jump Markov chain (Rosenthal et al., 2021), and further details are provided in Appendix E.

6 Conclusion

We introduced AC-Sampler, which accelerates and corrects diffusion sampling via Metropolis–Hastings with a Langevin proposal. By sampling from intermediate timesteps and using time-dependent discriminators for density ratio estimation, it improves sample quality and provides faster inference speed without retraining. AC-Sampler theoretically reduces KL divergence at each refinement step and empirically achieves better FID with fewer NFEs across datasets. It also integrates smoothly with prior acceleration and correction methods.

REFERENCES

- Haldun Akoglu. User's guide to correlation coefficients. *Turkish journal of emergency medicine*, 18 (3):91–93, 2018.
- Theodore G Birdsall, Kurt Metzger, Matthew A Dzieciuch, and John Spiesberger. Integrated autocorrelation phase at one period lag. *The Journal of the Acoustical Society of America*, 96(4): 2353–2356, 1994.
 - Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.
 - Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.
 - Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
 - Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), Advances in Neural Information Processing Systems, volume 35, pp. 30150–30166. Curran Associates, Inc., 2022a.
 - Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. In *International Conference on Learning Representations*, 2022b.
 - Ulf Grenander and Michael I Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(4):549–581, 1994.
 - WK Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
 - 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. *Advances in neural information processing systems*, 30, 2017.
 - Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
 - Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.
 - O. Hölder. Über ein mittelwertsatz. Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse, pp. 38–47, 1889.
 - Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.
 - Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.
 - Beomsu Kim and Jong Chul Ye. Denoising mcmc for accelerating diffusion-based generative models. In *International Conference on Machine Learning*, pp. 16955–16977. PMLR, 2023.
 - Dongjun Kim, Byeonghu Na, Se Jung Kwon, Dongsoo Lee, Wanmo Kang, and Il-chul Moon. Maximum likelihood training of implicit nonlinear diffusion model. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 32270–32284. Curran Associates, Inc., 2022a.
 - Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. In *International Conference on Machine Learning*, pp. 11201–11228. PMLR, 2022b.

Dongjun Kim, Yeongmin Kim, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Refining generative process with discriminator guidance in score-based diffusion models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 16567–16598. PMLR, 23–29 Jul 2023.

- Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.
- A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.
- Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.
- Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.
- Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PlKWVd2yBkY.
- Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.
- 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. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022a.
- Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.
- Hengyuan Ma, Li Zhang, Xiatian Zhu, and Jianfeng Feng. Accelerating score-based generative models with preconditioned diffusion sampling. In *European conference on computer vision*, pp. 1–16. Springer, 2022.
- Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.
- Byeonghu Na, Yeongmin Kim, Minsang Park, Donghyeok Shin, Wanmo Kang, and Il-Chul Moon. Diffusion rejection sampling. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 37097–37121, 2024.
- Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.
- Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.
- Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.
- Kushagra Pandey and Stephan Mandt. A complete recipe for diffusion generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4261–4272, 2023.
- Giorgio Parisi. Numerical simulations: Old and new problems. *Nonperturbative Quantum Field Theory*, pp. 427–432, 1988.

- William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.
 - Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. 1996.
 - Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.
 - Jeffrey S Rosenthal, Aki Dote, Keivan Dabiri, Hirotaka Tamura, Sigeng Chen, and Ali Sheikholeslami. Jump markov chains and rejection-free metropolis algorithms. *Computational Statistics*, 36(4): 2789–2811, 2021.
 - Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.
 - Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.
 - Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 11918–11930, 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 *International Conference on Learning Representations*, 2021b.
 - Dimiter Tsvetkov, Lyubomir Hristov, and Ralitsa Angelova-Slavova. On the convergence of the metropolis-hastings markov chains. *Serdica Math. J*, 43(2):93–110, 2017.
 - Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.
 - Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.
 - Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*, 35:23371–23385, 2022.
 - Yilun Xu, Shangyuan Tong, and Tommi S Jaakkola. Stable target field for reduced variance score estimation in diffusion models. In *The Eleventh International Conference on Learning Representations*, 2022.
 - Yilun Xu, Mingyang Deng, Xiang Cheng, Yonglong Tian, Ziming Liu, and Tommi Jaakkola. Restart sampling for improving generative processes. *Advances in Neural Information Processing Systems*, 36:76806–76838, 2023a.
 - Yilun Xu, Ziming Liu, Yonglong Tian, Shangyuan Tong, Max Tegmark, and Tommi Jaakkola. PFGM++: Unlocking the potential of physics-inspired generative models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38566–38591. PMLR, 23–29 Jul 2023b.
 - Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023.
 - Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Kaiwen Zheng, Yongxin Chen, Huayu Chen, Guande He, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Direct discriminative optimization: Your likelihood-based visual generative model is secretly a gan discriminator. In *Forty-second International Conference on Machine Learning*, 2025.

A PROOFS AND MATHEMATICAL EXPLANATIONS

In this section, we provide a mathematical explanation and formal derivation of the theorems presented in the main body of the paper.

A.1 THEORETICAL ANALYSIS OF DLG (KIM & YE, 2023)

Proposition A.1. Let $q(\mathbf{x},t)$ be the true joint distribution over data \mathbf{x} and diffusion timestep t, and $p^{\theta}(\mathbf{x},t)$ be the model joint distribution.

Suppose that there is an optimal time classifier ψ^* , i.e., $p^{\psi^*}(t \mid \mathbf{x}) = q(t \mid \mathbf{x}), \forall \mathbf{x}, t$. If the model marginal distribution $p^{\theta}(\mathbf{x})$ does not match the true marginal distribution, i.e., $\exists \mathbf{x} \text{ s.t. } p^{\theta}(\mathbf{x}) \neq q(\mathbf{x})$, then the Markov chain defined by Gibbs sampling between $p^{\theta}(\mathbf{x} \mid t)$ and $p^{\psi^*}(t \mid \mathbf{x})$ does not have $q(\mathbf{x}, t)$ as its stationary distribution.

Proof. Let's assume that the true joint distribution $q(\mathbf{x}_t,t)$ is a stationary distribution of Gibbs Sampling, whose transition kernel is $T((\mathbf{x},t) \to (\mathbf{x}',t')) = p^{\theta}(\mathbf{x}' \mid t)p^{\psi^*}(t' \mid \mathbf{x}')$. Let $(\mathbf{x},t) \sim q(\cdot,\cdot)$, and (\mathbf{x}',t') is a sample drawn from the transition kernel $T((\mathbf{x},t) \to (\cdot,\cdot))$. Since the stationary distribution should satisfy the invariance condition, for arbitrary \mathbf{x}' the following equation holds:

$$q(\mathbf{x}',t') = \iint q(\mathbf{x},t)T((\mathbf{x},t) \to (\mathbf{x}',t')) d\mathbf{x} dt$$
(15)

$$= \iint q(\mathbf{x}, t) p^{\boldsymbol{\theta}}(\mathbf{x}' \mid t) p^{\boldsymbol{\psi}^*}(t' \mid \mathbf{x}') d\mathbf{x} dt$$
 (16)

$$= \iint q(\mathbf{x}, t) p^{\theta}(\mathbf{x}' \mid t) q(t' \mid \mathbf{x}') d\mathbf{x} dt$$
 (17)

$$= \int q(t)p^{\theta}(\mathbf{x}' \mid t)q(t' \mid \mathbf{x}') dt$$
 (18)

$$= \int q(t) \frac{p^{\theta}(t \mid \mathbf{x}')p^{\theta}(\mathbf{x}')}{p^{\theta}(t)} q(t' \mid \mathbf{x}') dt$$
(19)

$$= \int p^{\theta}(t \mid \mathbf{x}')p^{\theta}(\mathbf{x}')q(t' \mid \mathbf{x}') dt$$
 (20)

$$= p^{\theta}(\mathbf{x}')q(t'\mid \mathbf{x}') \int p^{\theta}(t\mid \mathbf{x}') dt$$
 (21)

$$= p^{\theta}(\mathbf{x}')q(t'\mid \mathbf{x}') \tag{22}$$

Note that the marginal distribution $q(t) = p^{\theta}(t)$ for every t is an uniform distribution. To satisfy the invariance, $\forall \mathbf{x}', p^{\theta}(\mathbf{x}') = q(\mathbf{x}')$ so the proof holds due to the contradiction.

Proposition A.1 states that even if the time-classifier in DLG(Kim & Ye, 2023) is optimal, it is impossible to sample from the true distribution. So we leverage Metropolis-Hastings Correction with this proposal distribution. Detailed explanation is given in Appendix B.

A.2 PROOF OF PROPOSITION 4.2

Proposition 4.2. Let the reverse diffusion process have total timestep T and the AC-sampler target timestep be τ . Let NFE_R be the average NFE reduction per sample with AC-Sampler. If the acceptance probability satisfies $\alpha > \frac{1}{T-\tau+1}$, then,

$$\lim_{l \to \infty} \mathbb{E}[NFE_R] > 0, \quad \lim_{l \to \infty} \text{Var}(NFE_R) = 0$$
 (12)

where l is the length of Markov Chain at timestep τ .

Proof. Let's assume that the acceptance probability α is fixed in (0,1], and that there is no burn-in process in the AC-Sampler for simplicity.²

²We can get same result with burn-in process easily

Let R denote the total NFE required at timestep τ for the MCMC step. At timestep τ , Algorithm 1 runs l times. Since we make proposals until it is accepted, R follows the Negative Binomial distribution, i.e., $R \sim \mathrm{NB}(l,\alpha)$. Let NFE $_T$ denote the total NFE required in the AC-Sampler sampling step. Then the following decomposition holds:

$$NFE_T = \underbrace{T - \tau}_{\text{Initial denoising step}} + \underbrace{R}_{\text{MALA step}} + \underbrace{l(\tau - 1)}_{\text{Denoising step after MALA}}.$$
 (23)

Since we obtain the score value of each proposal at the MALA step and denoise total of l samples, the denoising step after MALA is $l(\tau-1)$. Since R follows a negative binomial distribution, we have $\mathbb{E}[\text{NFE}_T] = T - \tau + \frac{l}{\alpha} + l(\tau-1)$ and $\text{Var}(\text{NFE}_T) = \frac{l(1-\alpha)}{\alpha^2}$. Then, the mean of NFE_R is as follows:

$$\mathbb{E}[NFE_R] = T - \frac{\mathbb{E}[NFE_T]}{l}$$
 (24)

$$=T-\frac{T-\tau}{l}-\frac{1}{\alpha}-(\tau-1) \tag{25}$$

$$=T-\tau+1-\frac{1}{\alpha}-\frac{T-\tau}{l}.$$
 (26)

Taking limits on both sides yields

$$\lim_{l \to \infty} \mathbb{E}[NFE_R] = T - \tau + 1 - \frac{1}{\alpha}$$
(27)

$$> 0 \qquad \left(\because \alpha > \frac{1}{T - \tau + 1}\right)$$
 (28)

Moreover, $Var(NFE_R) = \frac{1-\alpha}{\alpha^2} \cdot \frac{1}{l}$, and thus taking limits gives $\lim_{l\to\infty} Var(NFE_R) = 0$, which concludes the proof.

A.3 Proof of Theorem 4.1

Theorem 4.1. Let \mathbf{x}_t and $\tilde{\mathbf{x}}_t$ be two arbitrary samples at diffusion timestep t. Then, for any fixed \mathbf{x}_{t-1} , the density ratio of the true marginal distribution q_t is given by:

$$\frac{q_t(\tilde{\mathbf{x}}_t)}{q_t(\mathbf{x}_t)} = \frac{q_{t|t-1}(\tilde{\mathbf{x}}_t \mid \mathbf{x}_{t-1})}{q_{t|t-1}(\mathbf{x}_t \mid \mathbf{x}_{t-1})} \cdot \frac{L_t(\tilde{\mathbf{x}}_t, t)}{L_t(\mathbf{x}_t, t)} \cdot \frac{p_{t-1|t}^{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{p_{t-1|t}^{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_t)}, \tag{8}$$

where $L_t(\mathbf{x}_t, t) := q_t(\mathbf{x}_t)/p_t^{\theta}(\mathbf{x}_t)$ denotes the likelihood ratio between the data and model marginal distributions at timestep t.

Proof. We derive the marginal density ratio as follows:

$$\frac{q_t(\tilde{\mathbf{x}}_t)}{q_t(\mathbf{x}_t)} = \frac{q_{t-1}(\mathbf{x}_{t-1})}{q_{t-1}(\mathbf{x}_{t-1})} \cdot \frac{q_{t|t-1}(\tilde{\mathbf{x}}_t \mid \mathbf{x}_{t-1})}{q_{t|t-1}(\mathbf{x}_t \mid \mathbf{x}_{t-1})} \cdot \frac{q_{t-1|t}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{q_{t-1|t}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_t)}$$
(29)

$$= \frac{q_{t|t-1}(\tilde{\mathbf{x}}_t \mid \mathbf{x}_{t-1})}{q_{t|t-1}(\mathbf{x}_t \mid \mathbf{x}_{t-1})} \cdot \frac{q_{t-1|t}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{q_{t-1|t}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_t)}$$
(30)

$$= \frac{q_{t|t-1}(\tilde{\mathbf{x}}_t \mid \mathbf{x}_{t-1})}{q_{t|t-1}(\mathbf{x}_t \mid \mathbf{x}_{t-1})} \cdot \frac{q_{t-1|t}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{p_{t-1|t}^{\theta}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)} \cdot \frac{p_{t-1|t}^{\theta}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_t)}{q_{t-1|t}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_t)} \cdot \frac{p_{t-1|t}^{\theta}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{p_{t-1|t}^{\theta}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_t)}$$
(31)

Following the derivation process of DiffRS (Na et al., 2024), Eq. 31 can be expressed as follows:

$$= \frac{q_{t|t-1}(\tilde{\mathbf{x}}_{t} \mid \mathbf{x}_{t-1})}{q_{t|t-1}(\mathbf{x}_{t} \mid \mathbf{x}_{t-1})} \cdot \frac{L_{t-1}(\mathbf{x}_{t-1}, t-1)}{L_{t}(\mathbf{x}_{t}, t)} \cdot \frac{L_{t}(\tilde{\mathbf{x}}_{t}, t)}{L_{t-1}(\mathbf{x}_{t-1}, t-1)} \cdot \frac{p_{t-1|t}^{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_{t})}{p_{t-1|t}^{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_{t})}$$
(32)

$$= \underbrace{\frac{q_{t|t-1}(\tilde{\mathbf{x}}_{t} \mid \mathbf{x}_{t-1})}{q_{t|t-1}(\mathbf{x}_{t} \mid \mathbf{x}_{t-1})}}_{\text{Forward term}} \cdot \underbrace{\frac{L_{t}(\tilde{\mathbf{x}}_{t}, t)}{L_{t}(\mathbf{x}_{t}, t)}}_{\text{Likelihood ratio}} \cdot \underbrace{\frac{p_{t-1|t}^{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_{t})}{p_{t-1|t}^{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \tilde{\mathbf{x}}_{t})}}_{\text{Transition kernel term}}$$
(33)

The Forward term and Transition kernel term are tractable since it is a gaussian distribution. By approximating the Likelihood ratio by a discriminator, we can derive a tractable form of the acceptance probability.

A.4 PROOF OF THEOREM 4.3

Theorem 4.3. Let p_0^{θ} , p_0^{θ,ϕ^*} denote the model distribution and refined distribution by AC-Sampler with optimal discriminator ϕ^* , respectively. Then, the KL divergence between the true data distribution q_0 and the refined distribution p_0^{θ,ϕ^*} is bounded by:

$$D_{KL}(q_0(\mathbf{x}_0)||p^{\boldsymbol{\theta},\boldsymbol{\phi}^*}(\mathbf{x}_0)) \le D_{KL}(q_0(\mathbf{x}_0)||p^{\boldsymbol{\theta}}(\mathbf{x}_0))$$
(13)

Proof. First, let τ be a timestep that MALA occurs in AC-Sampler framework. From (Ho et al., 2020), the upper bound of KL divergence between the true data distribution and the model distribution can be written as follows:

$$D_{\mathrm{KL}}[q_0 \parallel p_0^{\boldsymbol{\theta}}] = \mathbb{E}_{q_0} \left[\log \frac{q_0(\mathbf{x}_0)}{p_0^{\boldsymbol{\theta}}(\mathbf{x}_0)} \right]$$
(34)

$$= \mathbb{E}_{q_0}[-\log p_0^{\boldsymbol{\theta}}(\mathbf{x}_0)] - H(q_0) \tag{35}$$

$$= \mathbb{E}_{q_0} \left[-\log \int q_{1:\tau|0}(\mathbf{x}_{1:\tau} \mid \mathbf{x}_0) \frac{p_{0:\tau}^{\boldsymbol{\theta}}(\mathbf{x}_{0:\tau})}{q_{1:\tau|0}(\mathbf{x}_{1:\tau} \mid \mathbf{x}_0)} d\mathbf{x}_{1:\tau} \right] - H(q_0)$$
 (36)

$$\leq \mathbb{E}_{q_0} \left[-\int q_{1:\tau|0}(\mathbf{x}_{1:\tau} \mid \mathbf{x}_0) \log \frac{p_{0:\tau}^{\boldsymbol{\theta}}(\mathbf{x}_{0:\tau})}{q_{1:\tau|0}(\mathbf{x}_{1:\tau} \mid \mathbf{x}_0)} d\mathbf{x}_{1:\tau} \right] - H(q_0)$$
(37)

$$= \mathbb{E}_{q_{0:\tau}} \left[-\log p_{\tau}^{\boldsymbol{\theta}}(\mathbf{x}_{\tau}) - \sum_{i=1}^{\tau} \log \frac{p_{i-1|i}^{\boldsymbol{\theta}}(\mathbf{x}_{i-1} \mid \mathbf{x}_{i})}{q_{i-1|i}(\mathbf{x}_{i-1} \mid \mathbf{x}_{i})} \cdot \frac{q_{i}(\mathbf{x}_{i})}{q_{i-1}(\mathbf{x}_{i-1})} \right] - H(q_{0}) \quad (38)$$

$$= \mathbb{E}_{q_{0:\tau}} \left[\log \frac{q_{\tau}(\mathbf{x}_{\tau})}{p_{\tau}^{\boldsymbol{\theta}}(\mathbf{x}_{\tau})} - \log q_{0}(\mathbf{x}_{0}) - \sum_{i=1}^{\tau} \log \frac{p_{i-1\mid i}^{\boldsymbol{\theta}}(\mathbf{x}_{i-1} \mid \mathbf{x}_{i})}{q_{i-1\mid i}(\mathbf{x}_{i-1} \mid \mathbf{x}_{i})} \right] - H(q_{0})$$
(39)

$$= D_{\mathrm{KL}}[q_{\tau} \parallel p_{\tau}^{\boldsymbol{\theta}}] + \mathbb{E}_{q_{0:\tau}} \left[\sum_{i=1}^{\tau} \log \frac{q_{i-1|i}(\mathbf{x}_{i-1} \mid \mathbf{x}_{i})}{p_{i-1|i}^{\boldsymbol{\theta}}(\mathbf{x}_{i-1} \mid \mathbf{x}_{i})} \right]$$
(40)

$$= D_{\mathrm{KL}}[q_{\tau} \| p_{\tau}^{\theta}] + \sum_{i=1}^{\tau} \mathbb{E}_{q_{i}} \left[D_{\mathrm{KL}}[q_{i-1}|\mathbf{x}_{i-1} | \mathbf{x}_{i}) \| p_{i-1}^{\theta}(\mathbf{x}_{i-1} | \mathbf{x}_{i})] \right]$$
(41)

By substituting $p_{ au}^{m{ heta}}$ with $p_{ au}^{m{ heta}, m{\phi}^*}$, the following bounded relation also holds:

$$D_{\mathrm{KL}}[q_0 \parallel p_0^{\boldsymbol{\theta}, \phi^*}] \le D_{\mathrm{KL}}[q_\tau \parallel p_\tau^{\boldsymbol{\theta}, \phi^*}] + \sum_{i=1}^{\tau} \mathbb{E}_{q_i} \left[D_{\mathrm{KL}}[q_{i-1|i}(\mathbf{x}_{i-1} \mid \mathbf{x}_i) \parallel p_{i-1|i}^{\boldsymbol{\theta}, \phi^*}(\mathbf{x}_{i-1} \mid \mathbf{x}_i)] \right]$$
(42)

Since $p_{i-1|i}^{\theta,\phi^*}(\mathbf{x}_{i-1}\mid\mathbf{x}_i)=p_{i-1|i}^{\theta}(\mathbf{x}_{i-1}\mid\mathbf{x}_i)\ \forall i\leq \tau$, it is sufficient to show

$$D_{\mathrm{KL}}[q_{\tau} \parallel p_{\tau}^{\boldsymbol{\theta}, \boldsymbol{\phi}^*}] \le D_{\mathrm{KL}}[q_{\tau} \parallel p_{\tau}^{\boldsymbol{\theta}}] \tag{43}$$

When the discriminator is optimal and the burn-in process has been sufficiently performed, $p_{\tau}^{\theta,\phi^*} = q_{\tau}$, so $D_{\mathrm{KL}}[q_{\tau} \parallel p_{\tau}^{\theta,\phi^*}] = 0$. Since the KL-Divergence is non-negative, the proof is complete.

The stationary distribution at timestep τ is $p_{\tau}^{\theta}L_{\tau}^{\phi}$. The gap between the two distributions in Eq. 43 is as follows:

$$D_{\mathrm{KL}}[q_{\tau} \parallel p_{\tau}^{\boldsymbol{\theta}}] - D_{\mathrm{KL}}[q_{\tau} \parallel p_{\tau}^{\boldsymbol{\theta}, \boldsymbol{\phi}}] = \int q_{\tau} \log \frac{q_{\tau}}{p_{\tau}^{\boldsymbol{\theta}}} d\mathbf{x}_{\tau} - \int q_{\tau} \log \frac{q_{\tau}}{p_{\tau}^{\boldsymbol{\theta}, \boldsymbol{\phi}}} d\mathbf{x}_{\tau}$$
(44)

$$= \int q_{\tau} \log \frac{q_{\tau}}{p_{\tau}^{\theta}} d\mathbf{x}_{\tau} - \int q_{\tau} \log \frac{q_{\tau}}{p_{\tau}^{\theta} L_{\tau}^{\phi}} d\mathbf{x}_{\tau}$$
 (45)

$$= \int q_{\tau} \log L_{\tau}^{\phi} d\mathbf{x}_{\tau} \tag{46}$$

If the discriminator cannot distinguish between the two distributions p_t^{θ} , q_t at all, i.e., $L_t^{\phi}(\mathbf{x}_t, t) = 1$ for all \mathbf{x}_t , then the target distribution of Metropolis-Hastings algorithm becomes p_{τ}^{θ} . As a result the gap between the two KL divergences in Eq. 46 becomes 0. By training the discriminator, the gap converges to $D_{\text{KL}}[q_{\tau}||p_{\tau}^{\theta}||(\geq 0)$, indicating that the bound becomes tighter.

A.5 PROOF OF THEOREM 4.4

 Theorem 4.4. Let T_{τ} be the transition kernel of MALA at timestep τ . Also, $p_{\tau}^{\boldsymbol{\theta}, \phi^*, (l)}$ denotes marginal distribution at timestep τ after the l-th MALA transition from $p_{\tau}^{\boldsymbol{\theta}}$, and $p_0^{\boldsymbol{\theta}, \phi^*, (l)}$ denotes the data distribution generated from $p_{\tau}^{\boldsymbol{\theta}, \phi^*, (l)}$ with denoising transition kernel $p_{t-1|t}^{\boldsymbol{\theta}}$. \mathcal{L}^p denotes a space of function which satisfies $(\int_{\mathbb{R}} |f|^p d\mathbf{x})^{\frac{1}{p}} < \infty$. If $q_{\tau}(\mathbf{x}_{\tau}) \in \mathcal{L}^{\alpha}$, $\log\left(\frac{p_{\tau}^{\boldsymbol{\theta}, \phi^*, (l)}(\mathbf{x}_{\tau})}{q_{\tau}(\mathbf{x}_{\tau})}\right) \in \mathcal{L}^{\beta}$, $T_{\tau} \in \mathcal{L}^{\gamma}$, where $\alpha, \beta, \gamma \in [1, \infty]$ satisfy $\frac{1}{\alpha} + \frac{1}{\beta} + \frac{1}{\gamma} = 1$, then the KL divergence between the true data distribution q_0 and the refined distribution $p_0^{\boldsymbol{\theta}, \phi^*, (l+1)}$ is bounded by:

$$D_{KL}(q_0(\mathbf{x}_0)||p_0^{\boldsymbol{\theta},\phi^*,(l+1)}(\mathbf{x}_0)) \le D_{KL}(q_0(\mathbf{x}_0)||p_0^{\boldsymbol{\theta},\phi^*,(l)}(\mathbf{x}_0))$$
(14)

Proof. As in Theorem 4.3, it suffices to show that

$$D_{\mathrm{KL}}[q_{\tau} \parallel p_{\tau}^{\boldsymbol{\theta}, \boldsymbol{\phi}^*, (l+1)}] \leq D_{\mathrm{KL}}[q_{\tau} \parallel p_{\tau}^{\boldsymbol{\theta}, \boldsymbol{\phi}^*, (l)}]$$

Then the below equation holds. We refer to the proof procedure in (Tsvetkov et al., 2017).

Then the below equation holds. We fellet to the proof procedure in (Tsverkov et al., 2017).

$$D_{\text{KL}}[q_{\tau} \parallel p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l+1)}] = \int q_{\tau}(\mathbf{x}_{\tau}) \log \frac{q_{\tau}(\mathbf{x}_{\tau})}{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l+1)}(\mathbf{x}_{\tau})}$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ \int p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l+1)}(\tilde{\mathbf{x}}_{\tau}) T_{\tau}(\tilde{\mathbf{x}}_{\tau} \to \mathbf{x}_{\tau}) d\tilde{\mathbf{x}}_{\tau} \right\} d\mathbf{x}_{\tau} - H(q_{\tau})$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ \int \frac{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} q_{\tau}(\tilde{\mathbf{x}}_{\tau}) T_{\tau}(\tilde{\mathbf{x}}_{\tau} \to \mathbf{x}_{\tau}) d\tilde{\mathbf{x}}_{\tau} \right\} d\mathbf{x}_{\tau} - H(q_{\tau})$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ \int \frac{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} q_{\tau}(\mathbf{x}_{\tau}) T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) d\tilde{\mathbf{x}}_{\tau} \right\} d\mathbf{x}_{\tau} - H(q_{\tau})$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ \int \frac{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} q_{\tau}(\mathbf{x}_{\tau}) T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) d\tilde{\mathbf{x}}_{\tau} \right\} d\mathbf{x}_{\tau} - H(q_{\tau})$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ q_{\tau}(\mathbf{x}_{\tau}) \int \frac{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) d\tilde{\mathbf{x}}_{\tau} \right\} d\mathbf{x}_{\tau} - H(q_{\tau})$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ q_{\tau}(\mathbf{x}_{\tau}) \int \frac{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) d\tilde{\mathbf{x}}_{\tau} \right\} d\mathbf{x}_{\tau} - H(q_{\tau})$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ q_{\tau}(\mathbf{x}_{\tau}) \int \frac{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) d\tilde{\mathbf{x}}_{\tau} \right\} d\mathbf{x}_{\tau} - H(q_{\tau})$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ q_{\tau}(\mathbf{x}_{\tau}) \int \frac{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) d\tilde{\mathbf{x}}_{\tau} \right\} d\mathbf{x}_{\tau} - H(q_{\tau})$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ q_{\tau}(\mathbf{x}_{\tau}) \int \frac{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) d\tilde{\mathbf{x}}_{\tau} \right\} d\mathbf{x}_{\tau} - H(q_{\tau})$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ q_{\tau}(\mathbf{x}_{\tau}) \int \frac{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) d\tilde{\mathbf{x}}_{\tau} \right\} d\mathbf{x}_{\tau} - H(q_{\tau})$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ q_{\tau}(\mathbf{x}_{\tau}) \int \frac{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} T_$$

$$= -\int q_{\tau}(\mathbf{x}_{\tau}) \log \left\{ \int \frac{p_{\tau}^{\boldsymbol{\theta}, \boldsymbol{\phi}^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) d\tilde{\mathbf{x}}_{\tau} \right\} d\mathbf{x}_{\tau}$$
 (53)

$$\leq -\iint q_{\tau}(\mathbf{x}_{\tau}) \left[\log \left\{ \frac{p_{\tau}^{\boldsymbol{\theta}, \phi^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} \right\} T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) \right] d\tilde{\mathbf{x}}_{\tau} d\mathbf{x}_{\tau}$$
 (54)

Since $\frac{1}{\alpha} + \frac{1}{\beta} + \frac{1}{\gamma} = 1$, we can apply Hölder's inequality (Hölder, 1889) in Eq. 54

$$\iint q_{\tau}(\mathbf{x}_{\tau}) \left[\log \left\{ \frac{p_{\tau}^{\boldsymbol{\theta}, \boldsymbol{\phi}^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} \right\} T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) \right] d\tilde{\mathbf{x}}_{\tau} d\mathbf{x}_{\tau} \tag{55}$$

$$\leq \left(\iint |q_{\tau}(\mathbf{x}_{\tau})|^{\alpha} d\tilde{\mathbf{x}}_{\tau} d\mathbf{x}_{\tau} \right)^{\frac{1}{\alpha}} \left(\iint \left| \log \left\{ \frac{p_{\tau}^{\boldsymbol{\theta}, \boldsymbol{\phi}^{*}, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} \right\} \right|^{\beta} d\tilde{\mathbf{x}}_{\tau} d\mathbf{x}_{\tau} \right)^{\frac{1}{\beta}} \left(\iint |T_{\tau}(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau})|^{\gamma} d\tilde{\mathbf{x}}_{\tau} d\mathbf{x}_{\tau} \right)^{\frac{1}{\gamma}} \tag{56}$$

Since there exist α, β , and γ such that $q_{\tau}(\mathbf{x}_{\tau}) \in \mathcal{L}^{\alpha}$, $\log\left(\frac{p_{\tau}^{\theta, \phi^*, (l)}(\mathbf{x}_{\tau})}{q_{\tau}(\mathbf{x}_{\tau})}\right) \in \mathcal{L}^{\beta}$, and $T_{\tau} \in \mathcal{L}^{\gamma}$ holds, Eq. 54 is absolute convergence. Therefore, by Fubini's theorem, the order of integration can be interchanged.

$$= -\int \log \left(\frac{p_{\tau}^{\boldsymbol{\theta}, \boldsymbol{\phi}^*, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} \right) \left[\int q_{\tau}(\mathbf{x}_{\tau}) T(\mathbf{x}_{\tau} \to \tilde{\mathbf{x}}_{\tau}) d\mathbf{x}_{\tau} \right] d\tilde{\mathbf{x}}_{\tau}$$
 (57)

$$= -\int \log \left(\frac{p_{\tau}^{\boldsymbol{\theta}, \boldsymbol{\phi}^*, (l)}(\tilde{\mathbf{x}}_{\tau})}{q_{\tau}(\tilde{\mathbf{x}}_{\tau})} \right) q_{\tau}(\tilde{\mathbf{x}}_{\tau}) d\tilde{\mathbf{x}}_{\tau}$$
(58)

$$= D_{\mathrm{KL}}[q_{\tau} \parallel p_{\tau}^{\boldsymbol{\theta}, \boldsymbol{\phi}^*, (l)}] \tag{59}$$

The assumption in Theorem 4.4 is made solely to satisfy Fubini's theorem, and we note that the theorem is commonly adopted in prior works (De Bortoli et al., 2021; Lipman et al., 2023).

B METROPOLIS HASTINGS ALGORITHM IN JOINT SPACE

For the CelebA-HQ 256×256 dataset, we observed that the method which performed well on low-dimensional datasets did not perform well. We provide an analysis on this issue based on the following points:

- In high-dimensional data, the data manifold is more complex compared to low-dimensional cases. This makes it difficult to estimate the true score accurately.
- As a result, the proposal distribution becomes misaligned with the target distribution, and Langevin dynamics can easily drift away from the data manifold at timestep τ.

In DLG (Kim & Ye, 2023), a time classifier was proposed to detect whether a sample had left the manifold after Langevin dynamics at timestep t. The proposal distribution of DLG is as follows:

$$p_{\text{proposal}}^{\boldsymbol{\theta}, \boldsymbol{\psi}}(\tilde{\mathbf{x}}, \tilde{t} \mid \mathbf{x}, t) = p_{\text{proposal}}^{\boldsymbol{\theta}}(\tilde{\mathbf{x}} \mid \mathbf{x}, t) \cdot p_{\text{proposal}}^{\boldsymbol{\psi}}(\tilde{t} \mid \mathbf{x}, t, \tilde{\mathbf{x}}) = p_{\text{proposal}}^{\boldsymbol{\theta}}(\tilde{\mathbf{x}} \mid \mathbf{x}, t) \cdot p_{\text{proposal}}^{\boldsymbol{\psi}}(\tilde{t} \mid \tilde{\mathbf{x}})$$
(60)

First, given (\mathbf{x}, t) , sample $\tilde{\mathbf{x}}$ using one step of Langevin dynamics. After that, sample \tilde{t} using the time classifier conditioned on $\tilde{\mathbf{x}}$. The proposal distribution in the joint space depends not only on the score network but also on the time classifier.

However, as we showed in Proposition A.1, this approach cannot converge to the true joint distribution even when the time classifier is optimal, i.e., $p^{\psi^*}(t|\mathbf{x}) = q(t|\mathbf{x})$. To address this issue, we perform the Metropolis-Hastings algorithm in the joint space of time and data. To compute the acceptance probability in the joint distribution, we extend the density ratio formulation presented in Theorem 4.1. This extension is proposed in the following corollary.

Corollary B.1. Let $\mathbf{x}, \tilde{\mathbf{x}}$ be arbitrary samples at diffusion timesteps t, τ , respectively, and let \mathbf{x}_{τ} be any fixed point at timestep τ . If $\tau < \min(t, \tilde{t})$ is satisfied, the density ratio of the true joint distribution $q(\cdot, \cdot)$ is given by:

$$\frac{q(\tilde{\mathbf{x}}, \tilde{t})}{q(\mathbf{x}, t)} = \frac{q_{\tilde{t}|\tau}(\tilde{\mathbf{x}}|\mathbf{x}_{\tau})}{q_{t|\tau}(\mathbf{x}|\mathbf{x}_{\tau})} \cdot \frac{L_{\tilde{t}}(\tilde{\mathbf{x}}, \tilde{t})}{L_{t}(\mathbf{x}, t)} \cdot \frac{p_{\tau|t}^{\boldsymbol{\theta}}(\mathbf{x}_{\tau}|\mathbf{x})}{p_{\tau|\tilde{t}}^{\boldsymbol{\theta}}(\mathbf{x}_{\tau}|\tilde{\mathbf{x}})}$$
(61)

Algorithm 2 JointMALAOneStep($\mathbf{x}, t, \mathbf{s}, L_t, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}$)

Input: $\mathbf{x}, t, \mathbf{s}, L_t$

Output: Accepted sample $\tilde{\mathbf{x}}_{\tau}$

1: repeat

- $$\begin{split} & \text{Sample } \tilde{\mathbf{x}} \sim p_{\text{proposal},t}^{\boldsymbol{\theta}}(\cdot \mid \mathbf{x},t) \\ & \text{Sample } \tilde{t} \sim p_{\text{proposal}}^{\boldsymbol{\psi}}(\cdot \mid \tilde{\mathbf{x}}) \\ & \text{Compute } \tilde{\mathbf{s}} = \mathbf{s}^{\boldsymbol{\theta}}(\tilde{\mathbf{x}},\tilde{t}) \text{ and } \tilde{L}_{\tilde{t}} = L_{\tilde{t}}^{\phi}(\tilde{\mathbf{x}},\tilde{t}) \end{split}$$
- Compute acceptance probability: $\hat{\alpha}_{joint}(\mathbf{x}, \tilde{\mathbf{x}}, \mathbf{s}, \tilde{\mathbf{s}}, L_t, \tilde{L}_t, \tau)$
- Sample $u \sim \mathcal{U}(0,1)$
- 7: **until** $u < \alpha$
- 8: **return** $\tilde{\mathbf{x}}, \tilde{t}, \tilde{\mathbf{s}}, \tilde{L}_{\tilde{t}}$

Proof. For $\forall \mathbf{x}_{\tau}$ with $\tau < t, \tilde{t}$, the below equation holds.

$$\frac{q(\tilde{\mathbf{x}}, \tilde{t})}{q(\mathbf{x}, t)} = \frac{q(\tilde{\mathbf{x}} \mid \tilde{t})}{q(\mathbf{x} \mid t)} \cdot \frac{q(\tilde{t})}{q(t)}$$
(62)

$$=\frac{q_{\tilde{t}}(\tilde{\mathbf{x}})}{q_t(\mathbf{x})}\tag{63}$$

$$= \frac{q_{\tilde{t}|\tau}(\tilde{\mathbf{x}} \mid \mathbf{x}_{\tau})}{q_{t|\tau}(\mathbf{x} \mid \mathbf{x}_{\tau})} \cdot \frac{q_{\tau|t}(\mathbf{x}_{\tau} \mid \mathbf{x})}{q_{\tau|\tilde{t}}(\mathbf{x}_{\tau} \mid \tilde{\mathbf{x}})} \cdot \frac{q_{\tau}(\mathbf{x}_{\tau})}{q_{\tau}(\mathbf{x}_{\tau})}$$
(64)

$$= \frac{q_{\tilde{t}|\tau}(\tilde{\mathbf{x}} \mid \mathbf{x}_{\tau})}{q_{t|\tau}(\mathbf{x} \mid \mathbf{x}_{\tau})} \cdot \frac{q_{\tau|t}(\mathbf{x}_{\tau} \mid \mathbf{x})}{q_{\tau|\tilde{t}}(\mathbf{x}_{\tau} \mid \tilde{\mathbf{x}})}$$

$$(65)$$

$$= \frac{q_{\tilde{t}|\tau}(\tilde{\mathbf{x}} \mid \mathbf{x}_{\tau})}{q_{t|\tau}(\mathbf{x} \mid \mathbf{x}_{\tau})} \cdot \frac{q_{\tau|t}(\mathbf{x}_{\tau}|\mathbf{x})}{p_{\tau|t}^{\theta}(\mathbf{x}_{\tau}|\mathbf{x})} \cdot \frac{p_{\tau|\tilde{t}}^{\theta}(\mathbf{x}_{\tau}|\tilde{\mathbf{x}})}{q_{\tau|\tilde{t}}(\mathbf{x}_{\tau}|\tilde{\mathbf{x}})} \cdot \frac{p_{\tau|t}^{\theta}(\mathbf{x}_{\tau}|\mathbf{x})}{p_{\tau|\tilde{t}}^{\theta}(\mathbf{x}_{\tau}|\tilde{\mathbf{x}})}$$
(66)

$$= \frac{q_{\tilde{t}|\tau}(\tilde{\mathbf{x}} \mid \mathbf{x}_{\tau})}{q_{t|\tau}(\mathbf{x} \mid \mathbf{x}_{\tau})} \cdot \frac{L_{\tau}(\mathbf{x}_{\tau}, \tau)}{L_{t}(\mathbf{x}, t)} \cdot \frac{L_{\tilde{t}}(\tilde{\mathbf{x}}, \tilde{t})}{L_{\tau}(\mathbf{x}_{\tau}, \tau)} \cdot \frac{p_{\tau|t}^{\theta}(\mathbf{x}_{\tau} \mid \mathbf{x})}{p_{\tau|\tilde{t}}^{\theta}(\mathbf{x}_{\tau} \mid \tilde{\mathbf{x}})}$$
(67)

$$= \frac{q_{\tilde{t}|\tau}(\tilde{\mathbf{x}} \mid \mathbf{x}_{\tau})}{q_{t|\tau}(\mathbf{x} \mid \mathbf{x}_{\tau})} \cdot \frac{L_{\tilde{t}}(\tilde{\mathbf{x}}, \tilde{t})}{L_{t}(\mathbf{x}, t)} \cdot \frac{p_{\tau|t}^{\theta}(\mathbf{x}_{\tau} | \mathbf{x})}{p_{\tau|\tilde{t}}^{\theta}(\mathbf{x}_{\tau} | \tilde{\mathbf{x}})}$$
(68)

The density of the proposal distribution, $p_{\text{proposal}}^{\theta}(\tilde{\mathbf{x}} \mid \mathbf{x}, t) \cdot p_{\text{proposal}}^{\psi}(\tilde{t} \mid \tilde{\mathbf{x}})$, is tractable. In detail, $p_{\text{proposal}}^{\theta}(\tilde{\mathbf{x}} \mid \mathbf{x}, t)$ is a Langevin proposal, which follows a Gaussian distribution and $p_{\text{proposal}}^{\psi}(\tilde{t} \mid \tilde{\mathbf{x}})$ can be evaluated using the output of the time classifier. Therefore, the acceptance probability in the joint space can be computed. The acceptance probability is given as follows:

$$\hat{\alpha}_{joint}(\mathbf{x}, \tilde{\mathbf{x}}, \mathbf{s}, \tilde{\mathbf{s}}, L, L, \tau)$$

$$= \min \left(1, \underbrace{\frac{q_{\tilde{t}|\tau}(\tilde{\mathbf{x}} \mid \hat{\mathbf{x}}_{\tau})}{q_{t|\tau}(\mathbf{x} \mid \hat{\mathbf{x}}_{\tau})}}_{\text{Forward term}} \cdot \underbrace{\frac{\tilde{L}}{L}}_{\text{Likelihood ratio}} \cdot \underbrace{\frac{p_{\tau|t}^{\theta}(\mathbf{x}_{\tau}|\mathbf{x})}{p_{\tau|\tilde{t}}^{\theta}(\mathbf{x}_{\tau}|\tilde{\mathbf{x}})}}_{\text{Transition kernel term}} \cdot \underbrace{\frac{p_{\text{proposal}}^{\theta}(\mathbf{x} \mid \tilde{\mathbf{x}}, \tilde{t}) \cdot p_{\text{proposal}}^{\psi}(t \mid \mathbf{x})}{p_{\text{proposal}}^{\theta}(\tilde{\mathbf{x}} \mid \tilde{\mathbf{x}}, t) \cdot p_{\text{proposal}}^{\psi}(\tilde{t} \mid \tilde{\mathbf{x}})} \right)$$
(69)

 $\tau, \hat{\mathbf{x}}_{\tau}$ can be any point. We choose the value of τ such that it does not deviate significantly from the original timestep t. In our experiments, we empirically set τ so that the standard deviation of $q_{\tau|0}(\mathbf{x}_{\tau}\mid\mathbf{x}_0)$ differs from that of $q_{\min(t,\tilde{t})\mid0}(\mathbf{x}_{\min(t,\tilde{t})\mid}\mid\mathbf{x}_0)$ by 0.1, based on the VESDE(Song et al., 2021b) parameterization. We set $\hat{\mathbf{x}}_{\tau} = \frac{1}{2} (\boldsymbol{\mu}_{t}(\mathbf{x}, \mathbf{s}^{\theta}(\mathbf{x}, t)) + \boldsymbol{\mu}_{\tilde{t}}(\tilde{\mathbf{x}}, \mathbf{s}^{\theta}(\tilde{\mathbf{x}}, \tilde{t})))$. The detailed process is in Algorithm 2.

Since the timestep is proposed for every update of Alg. 2, we need to reassign the starting timestep t for denoising. We first fix the total number of steps T, and perform T-t steps of denoising

Table 8: FID and NFE on unconditional CelebA-HQ 256×256 generation with ScoreSDE (Song et al., 2021b), DLG (Kim & Ye, 2023), AC (marginal), and AC (joint).

	FID↓	NFE↓	FID↓	NFE↓	FID↓	NFE↓
ScoreSDE (KAR1)	121.27	40	122.74	98	125.15	198
+DLG	20.19	21.21	29.12	47.21	30.72	107.21
+AC (marginal)	103.81	19.78	75.51	41.83	87.40	84.13
+AC (joint)	15.13	15.94	22.55	44.06	15.69	87.26
ScoreSDE (KAR2)	83.21	40	57.28	98	29.74	198
+DLG	17.92	21.21	12.12	47.21	8.14	107.21
+AC (marginal)	45.74	19.55	19.48	42.97	9.45	94.34
+AC (joint)	8.45	20.05	9.55	40.05	6.60	98.27

from t down to the proposed timestep \tilde{t} . Then, we perform t steps of denoising from \tilde{t} to 0. Our methodology generalizes the approach of DLG (Kim & Ye, 2023). While DLG generates samples from the joint space of time and data using Gibbs sampling, we introduce the Metropolis-Hastings algorithm to correct samples toward the true data distribution by additionally training a time-dependent discriminator. We adopt the time classifier from the official code of DLG and use the argmax of the classifier output as the proposed timestep, following their original approach. Since using the argmax results in a deterministic time proposal distribution, we set $p^{\psi}(t \mid \mathbf{x}) = 1$ when computing the acceptance probability.

For fair comparison, we reproduced the experimental setting of DLG. We first obtained the best parameters for both the KAR1 and KAR2 samplers as reported in DLG, and then reproduced their performance using these optimal settings. Subsequently, we increased the number of denoising steps while keeping the remaining parameters unchanged.

C HYPERPARAMETER DETAIL

As described in the main text, our method treats the MH target diffusion timestep τ and the Langevin step size (controlled by the signal-to-noise ratio, SNR) as the primary parameters. In addition, we employ several auxiliary hyperparameters: the number of skipped steps $n_{\rm skip}$, the burn-in length $n_{\rm burn-in}$, and the number of parallel chains $n_{\rm chain}$. Their roles are summarized as follows:

- $n_{\rm skip}$: Controls how many intermediate steps are skipped between proposals.
- $n_{\text{burn-in}}$: Specifies the number of initial iterations discarded to reduce initialization bias.
- n_{chain} : Denotes the length of MCMC chains. With one initial point, we can get n_{chain} samples.

Among the hyperparameters, we regard the choice of τ as the most critical. As τ decreases—i.e., as the state approaches the data distribution—the marginal distribution becomes sharper. This sharpness increases the computational burden of moving across the space via MCMC. While smaller τ brings the chain closer to the true data distribution (see proof of Theorem 4.2), it also requires a larger number of samples to sufficiently cover the support. Conversely, if τ is set too low, the effective reduction in NFE diminishes and distributional alignment becomes less pronounced. Therefore, selecting an appropriate τ is essential. Empirically, we found that setting τ between $\frac{1}{2}T$ and $\frac{3}{4}T$ achieves the most effective trade-off.

The second key parameter is the SNR, which controls the step size η of the Langevin proposal:

$$\sqrt{\eta} = \text{SNR} \times \left(\frac{2 \cdot |\epsilon|}{|\mathbf{s}|}\right).$$
 (70)

A too-small SNR yields excessively small step sizes, limiting sample diversity, while a too-large SNR hampers convergence of the MH correction. Based on prior works that adopted Langevin sampling in diffusion models (e.g., Song & Ermon (2019); Song et al. (2021b)), we set the SNR in the range of 0.1–0.25.

1082

1084

1086

1087

1088

1089 1090

1093

1095

1103

1112

1113

1114 1115

1116

1117 1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129 1130

1131 1132

1133

Experimental results on varying these two key parameters are reported in Table 9. We conducted experiments with the total number of timesteps fixed at 18, while keeping $n_{\rm skip}$, $n_{\rm burn-in}$, and $n_{\rm chain}$ constant. As shown in the Table, when τ is set too small, the MCMC chain tends to remain in a limited region of the space for a long time. Consequently, covering the entire distribution requires significantly higher computational cost, which is reflected in the degraded FID and Recall metrics. Furthermore, as the SNR increases, the acceptance probability gradually decreases. Across our overall experimental setup, the SNR satisfies $SNR \le 0.25$, which allows us to maintain an acceptance probability of approximately $\alpha \gtrsim 0.25$ (see Table 9). Because α is sufficiently large, as argued in the main text, substantial NFE reduction can be achieved even with a small chain length l. These experimental results support the preceding analysis in Proposition 4.2.

Method	τ	SNR	FID↓	NFE↓	Recall↑	C.I of α
EDM (Base)		-	2.01	35	0.627	
EDINI (Buse)		0.1	2.89	25.13	0.550	0.9150 ± 0.0023
		0.2	2.06	25.65	0.620	0.6248 ± 0.0033
+AC	13	0.23	1.97	26.19	0.628	0.4703 ± 0.0030
		0.27	2.09	28.11	0.625	0.2493 ± 0.0019
		0.1	8.77	15.21	0.200	0.8689 ± 0.0027
	0	0.2	6.14	15.75	0.422	0.6560 ± 0.0039
+AC	8	0.23	6.73	16.70	0.441	0.4671 ± 0.0034
		0.27	9.45	18.66	0.448	0.2733 ± 0.0022
		0.1	62.67	5.29	0	0.8525 ± 0.0046
	3	0.2	46.60	5.75	0	0.6590 ± 0.0054
+AC	3	0.23	43.96	6.26	0	0.4812 ± 0.0054
		0.27	39.44	9.13	0	0.2703 ± 0.0022

Table 9: Results for different τ values and SNR settings, including FID, NFE, Recall, and acceptance probabilities with 95% confidence intervals.

Method	$n_{\rm skin}$	FID↓	NFEL	Recall↑	Method	n	FID	NFEL	Recall↑	Method	$n_{ m burn-in}$	FID↓	NFE↓	Recall↑
EDM		2.01	35	0.627		$n_{\rm chain}$	· ·	· •		EDM	-	2.01	35	0.627
EDM	-				EDM	-	2.01	35	0.627		0	1.99	36.23	0.640
	0	1.97	26.19	0.628		10	2.03	29.28	0.622		1	2.02	37.46	0.624
	1	1.98	28.25	0.631		50	2.00	26.69	0.630		2	2.01	38.69	0.633
	2	1.94	30.28	0.634							2			
+ AC	2	1.97	32.30	0.638	+ AC	100	2.07	26.38	0.624	+ AC	5	1.99	42.37	0.632
+ AC	3					300	1.97	26.19	0.629		10	1.97	48.54	0.629
	4	2.00	34.39	0.640							20	1.97	60.81	0.632
	5	2.02	36.38	0.623		500	2.02	26.12	0.625		50	1.98	97.58	0.629
											50	1.98	97.38	0.029

Table 10: FID and NFE for dif- Table 11: FID and NFE for differ- Table 12: FID and NFE for difent chain steps. ferent skip steps.

ferent burn-in steps.

The following reports the results of varying each auxiliary parameter. Tables 10, 11, and 12 present the outcomes for changing n_{skip} , n_{chain} , and $n_{\text{burn-in}}$, respectively. Unless otherwise noted, all experiments are conducted with T=18, $\tau=13$.

Table 10 reports the effect of varying n_{skip} while fixing SNR = 0.23, n_{chain} = 300, and $n_{\text{burn-in}}$ = 10. The parameter $n_{\rm skip}$ helps reduce autocorrelation between samples; however, excessively large values increase the NFE, limiting the achievable acceleration gain. Empirically, we set $n_{\rm skip} = 0 \sim 1$ for CIFAR-10 and ImageNet, and maximum 4 for CelebA-HQ 256×256 .

Table 11 shows the results obtained by varying n_{chain} while fixing SNR = 0.23 and $n_{\text{burn-in}}$ = 10. When SNR is too small, recall may vary with n_{chain} , but under reasonable SNR values the recall is largely insensitive to $n_{\rm chain}$. Nevertheless, setting $n_{\rm chain}$ too small can hinder effective NFE reduction.

Table 12 investigates the role of the burn-in process by varying $n_{\text{burn-in}}$ while fixing SNR = 0.16 and $n_{\text{chain}} = 1$. We set $n_{\text{chain}} = 1$ in order to isolate and examine the effect of correction on each sample. We observe that after about 10 burn-in steps, the chain sufficiently converges, indicating that the score-based proposal distribution indeed allows proper convergence. In practice, we set $n_{\text{burn-in}} \leq 10$.

D MCMC MIXING

Images are high-dimensional data, which makes direct statistical evaluation of Markov chain mixing challenging. To address this, we assess mixing indirectly by analyzing the class labels of generated

images. We trained a ResNet based classifier that achieves 95% accuracy on the CIFAR-10 test set, and used it to assign class labels to each generated image, thus forming a class sequence along the MCMC chain. We constructed Markov chains of length 300.

We measured Integrated Autocorrelation Time(IACT) (Birdsall et al., 1994) 30 times with a maximum lag of 100. With Table 14 we observed trends consistent with Recall metrics. Lower IACT values indicate better mixing, suggesting that our method yields well-mixed samples. However, we note that IACT is originally defined for continuous variables, and applying it to categorical class labels can be limiting.

To complement IACT, we also computed Cramér's V (Akoglu, 2018) 30 times to assess class autocorrelation in the discrete label space in Table 13. Under the best-performing setting ($\tau=5, {\rm SNR}=0.23$), the value at lag 1 shows a relatively strong correlation (Akoglu, 2018), which is expected since our sampler proposes candidates based on local gradients. Nevertheless, both improvements in the Recall metric and our toy experiment 5.3 demonstrate that, despite such correlations, the chain is able to generate sufficiently diverse samples.

Lag	Cramér's V \pm std
1	0.360 ± 0.035
2	0.259 ± 0.037
3	0.214 ± 0.044
4	0.182 ± 0.042
5	0.172 ± 0.036

Table 13: Cramér's V across lags.

Method	τ	SNR	FID↓	Recall ↑	IACT of class sequence ↓
EDM (Base)	-	-	2.01	0.627	-
		0.1	2.89	0.550	8.71 ± 10.02
	13	0.2	2.06	0.620	2.45 ± 3.23
+AC	13	0.23	1.97	0.628	1.53 ± 1.70
		0.27	2.09	0.625	1.89 ± 1.35
		0.1	8.77	0.200	29.47 ± 26.86
	8	0.2	6.14	0.422	21.65 ± 24.84
+AC	o	0.23	6.73	0.441	20.92 ± 24.50
		0.27	9.45	0.448	24.39 ± 25.55
		0.1	62.67	0	38.45 ± 27.39
	3	0.2	46.60	0	34.93 ± 27.83
+AC	3	0.23	43.96	0	22.51 ± 23.15
		0.27	39.44	0	24.45 ± 32.90

Table 14: IACT of class sequence for different τ values and SNR settings.

E METROPOLIS-HASTINGS ALGORITHM AND ALGORITHM 1

Algorithm 1 employs a propose-until-accept update: at each step, proposals are repeatedly drawn and subjected to the MH accept—reject test until one is accepted, and the accepted proposal is then emitted as the next sample. We adopted this design for empirical reasons, namely to mitigate stagnation and preserve sample diversity. In canonical Metropolis—Hastings, however, a rejection corresponds to a self-transition, which is essential for preserving detailed balance. Eliminating self-transitions by proposing until acceptance alters the transition kernel and can introduce stationary bias.

This variant can be interpreted as a Jump Markov chain (Rosenthal et al., 2021). In such chains, the target distribution is implicitly modified because the rejection mechanism no longer permits self-transitions. Following Rosenthal et al. (2021), the stationary distribution of the jump chain, denoted $\hat{\pi}$, can be expressed in terms of the original stationary distribution π as

$$\hat{\pi}(x) = c\alpha(x)\pi(x),\tag{71}$$

where $\alpha(x) := 1 - P_{\text{transition}}(x|x)$ is the escape probability at state x, and $c = \mathbb{E}_{y \sim \pi}[\alpha(y)]^{-1}$ is a normalizing constant. Here $P_{\text{transition}}(\cdot|\cdot)$ denotes the transition probability of the original MH chain. The KL divergence between π and $\hat{\pi}$ is then

$$D_{\mathrm{KL}}[\pi||\hat{\pi}] = \mathbb{E}_{x \sim \pi} \left[\log \frac{\pi(x)}{\hat{\pi}(x)} \right] = \mathbb{E}_{x \sim \pi} \left[\log \frac{1}{c\alpha(x)} \right] = \mathbb{E}_{x \sim \pi} \left[\log \frac{\mathbb{E}_{y \sim \pi}[\alpha(y)]}{\alpha(x)} \right]. \tag{72}$$

This formulation shows that the jump chain introduces a KL divergence bias. When $\alpha(x)$ is constant over the support of π , no bias arises; otherwise, the deviation can be non-negligible. Despite the strong empirical performance of our method, a distributional gap remains. We leave a rigorous theoretical analysis of this gap to future work.

F ADDITIONAL EXPERIMENT

F.1 EXTEND AC-SAMPLER TO CORRECT EACH SAMPLES: MULTI STEP CORRECTION & REFINED PROPOSAL WITH A DISCRIMINATOR

After the burn-in process of the Metropolis-Hastings algorithm, the samples generated from our method can be regarded as samples drawn from the true distribution. This demonstrates that the Metropolis-Hastings algorithm can be used not only to accelerate sampling, but also to correct intermediate samples to better match the target distribution.

Focusing solely on the correction perspective, our proposed framework naturally incorporates the following methodological components: *multi-step correction* and *refined proposal with a discriminator*. As discussed in the main text, we initially present our algorithm using a single-step formulation for simplicity. However, applying our method in a multi-step setting is straightforward and does not pose any conceptual or technical difficulties. Therefore, we also conducted experiments under the multi-step setting.

Furthermore, DG (Kim et al., 2023) proposed correcting the score network using the gradient information from a discriminator. Since we adopt exactly the same training scheme for the discriminator as in DG, it is reasonable to apply a refined proposal based on the corrected score network. This implies that the discriminator trained at timestep τ not only provides a likelihood ratio estimate, but also enables refining the proposal distribution $p_{\text{proposal},\tau}^{\theta}$. It is possible to use

Table 15: Comparison of FID and sampling settings under different configurations

Sampling	FID↓	NFE↓	T	au	SNR	$n_{ m burn-in}$	n_{chain}
EDM	1.97	35	18	-	_	_	-
	2.18	51	18	1, 3, 5, 7	0.16	3	1
EDM + PC	2.13	51	18	7, 9, 11, 13	0.16	3	1
	2.00	51	18	11, 13, 15, 17	0.16	3	1
	1.94	54.66	18	1, 3, 5, 7	0.16	3	1
EDM + AC	1.96	55.33	18	7, 9, 11, 13	0.16	3	1
	1.93	54.91	18	11, 13, 15, 17	0.16	3	1
	1.98	54.65	18	1, 3, 5, 7	0.16	3	1
$EDM + AC + DG_p$	1.87	54.56	18	7, 9, 11, 13	0.16	3	1
I	1.92	54.17	18	11, 13, 15, 17	0.16	3	1

 $\overline{\mathrm{DG}_p}$ in accelerating, but $\overline{\mathrm{DG}_p}$ needs gradient calculation and this made sampling speed slow.

We present the results of both extensions in Table 15. We denote this discriminator-guided proposal scheme as DG_p . The result demonstrates the effect of MALA correction across various choices of the correction timestep τ . Although the correction timestep increases while maintaining the same SNR, AC-Sampler either improves or maintains the baseline FID score. Furthermore, we observe that incorporating the DG scheme into AC-Sampler leads to a meaningful reduction in FID with shorter length of chain. In contrast, PC-Sampler (Song et al., 2021b) often fails to correct samples at large correction timesteps.

In the table, setting $n_{\rm chain}=1$ indicates that Metropolis-Hastings correction is applied for each individual sample. While this setting does not reduce the number of function evaluations (NFE), it effectively corrects each intermediate sample.

F.2 DISCUSSION ABOUT TRADE-OFF IN CLASS-CONDITIONAL GENERATION

When performing class-conditional generation, the number of independent class samples plays an important role. For FID evaluation, we use 50K images. With $n_{\rm chain}=10$, the baseline involves 50,000 independent class samplings, whereas only 5,000 samplings occur with our method, which may lead to class imbalance. If this effect did not exist, the NFE reduction could be even more effective. To evaluate our method fairly under this setting, we generate 250K samples with $n_{\rm chain}=5$ (this setting makes total of 50K independent class samplings) and compute the FID five times using randomly selected subsets of 50K samples. We report the mean and stan-

Table 16: FID and NFE results on ImageNet 64×64 across different (SNR, τ) settings.

Method	SNR	τ	FID↓	NFE↓
EDM	-	-	2.30	61
+AC	0.12	8	2.28±0.05	52.91
+AC	0.12	9	2.29 ± 0.03	51.30

dard deviation of the resulting FID. Table 16 presents this analysis, showing that our method can significantly reduce NFE while maintaining a comparable FID.

F.3 CIFAR-10 WITH SCORESDE

We also conduct experiments with Score-SDE (Song et al., 2021b) and DLG (Kim & Ye, 2023). Table 17 and Figure 7 reports our reproductions of the base models (Score-SDE and DLG) alongside our method. Because the publicly released Score-SDE checkpoint is configured for sampling with roughly 1,000 NFE, achieving strong performance at substantially lower NFE is inherently challenging. Following the evaluation protocol described in the DLG paper, our reproduced results improved over the base checkpoint but did not exactly match the values reported in the original work. Under the same setting, applying our method yielded consistent distribution correction and quality improvements, even in the low-NFE regime.

	FID↓	NFE↓	FID↓	NFE↓	FID↓	NFE↓
ScoreSDE (Base)	27.35	16	26.58	26	26.72	36
+DLG				21.23		52.23
+AC (Ours)	25.18	10.95	23.83	19.13	23.14	29.07

Table 17: Experiment results on CIFAR-10 with ScoreSDE

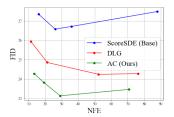


Figure 7: ScoreSDE base experiment on CIFAR-10

G EXPERIMENT SETTING

G.1 EXPERIMENTAL SETUP

Setups. We evaluate on CIFAR-10, CelebA-HQ 256×256 , ImageNet 64×64 , and ImageNet 256×256 . On CIFAR-10, we assess EDM (Karras et al., 2022) and DDO (Zheng et al., 2025) using the Heun sampler as in EDM, ScoreSDE (Song et al., 2021b) adopting samplers KAR1 (deterministic) and KAR2 (stochastic) (Kim & Ye, 2023). For ScoreSDE, refer Appendix F.3. On CelebA-HQ, we use the ScoreSDE (Song et al., 2021b) checkpoint within the DLG codebase (Kim & Ye, 2023) with KAR1, KAR2 sampler. On ImageNet 64×64 , we use the EDM checkpoint with the SDE sampler from (Karras et al., 2022); on ImageNet 256×256 , we use the DiT checkpoint (Peebles & Xie, 2023) with a DDPM sampler (Ho et al., 2020).

Codebases and checkpoints. Our experiments use the official repositories of EDM³, DLG⁴, DG⁵, ⁶, DDO⁷, ScoreSDE⁸, and DiT⁹.

Discriminator training. We train a time-dependent discriminator per network following DG (Kim et al., 2023), using the pre-trained ADM classifier (Dhariwal & Nichol, 2021) as the feature extractor. Compared to training a diffusion model, discriminator training is substantially cheaper.

Metrics. We report FID and the mean number of function evaluations (NFE) of the score network (as in DLG (Kim & Ye, 2023)), since NFE varies across samples in our method. FID is computed on 50K generated samples against the 50K test images; for CelebA-HQ 256×256, we report 10K FID. Also we report Precision / Recall metric to assess both the fidelity and diversity of generated images. The computation of FID follows the official implementation provided by DG (Kim et al., 2023). We measure the Precision and Recall using the ADM codebase (Dhariwal & Nichol, 2021).

https://github.com/NVlabs/edm

⁴https://github.com/1202kbs/DMCMC

⁵https://github.com/aailabkaist/DG

⁶https://github.com/alsdudrla10/DG_imagenet

⁷https://github.com/NVlabs/DDO

⁸https://github.com/yang-song/score_sde_pytorch

⁹https://github.com/facebookresearch/DiT

All experiments were conducted on NVIDIA RTX 3090, 4090 GPU and A100 GPU using Python 3.8, PyTorch 1.12, and CUDA 11.4.

Table 18: Sampling configuration and performance metrics (FID / NFE) with various diffusion and sampler combinations.

Dataset	Task	Base Model	Base Sampler	T	SNR	n_{chain}	$n_{ m burn-in}$	$n_{ m skip}$	au	FID	NFE
CIFAR-10	Uncond.	EDM	EDM (Heun)	18	0.23	50	10	0	11	2.10	22.78
CH7HC-10	Cheona.	LDM	LDW (Heun)	18	0.23	300	10	0	13	1.97	26.19
				18	0.23	500	10	0	13	2.02	26.12
				18	0.23	3,4,5	5	4	13,14,15	1.93	44.40
				14	0.2	50	0	0	6	2.38	15.82
				10	0.2	11	0	0	5	3.24	10.57
CIFAR-10	Uncond.	DDO	EDM (Heun)	16	0.175	2	0	2	13	1.41	29.41
CIFAR-10	Uncond.	EDM	DPM-Solver-v3	6	0.16	10	0	0	5	7.12	5.62
				8	0.16	10	0	0	7	3.09	7.54
				8	0.1	7	0	0	4	9.88	4.78
				10	0.15	3	0	0	9	2.55	9.93
CIFAR-10	Uncond.	ScoreSDE	KAR1	18	0.23	250	10	1	13	23.14	29.08
				18	0.23	250	10	1	11	23.80	25.18
				18	0.23	250	10	0	14	23.83	19.13
ImageNet 64×64	Cond.	EDM	EDM (SDE)	32	0.16	2	1	0	26	2.25	58.75
				64	0.18	2	5	1	50	1.77	121.98
				256	0.1	2	5	1	225	1.42	483.86
CelebA-HQ 256×256	Uncond.	ScoreSDE	KAR1	20	0.16	25	10	0	13	15.13	15.94
				49	0.16	100	10	4	30	22.55	44.07
				99	0.16	25	10	3	60	15.69	87.26
CelebA-HQ 256×256	Uncond.	ScoreSDE	KAR2	20	0.16	25	10	1	12	8.45	20.05
				49	0.16	50	10	4	33	9.55	40.05
				99	0.18	25	10	3	55	6.60	98.27
ImageNet 256×256	Cond.	DiT	DDPM	250	0.12	2	190	0	10	2.31	234.38

G.2 DISCRIMINATOR DETAILS

To implement the time-dependent discriminator, we directly used the official DG codebase and followed their approach. On CIFAR-10, we used the discriminator checkpoint provided by DG only when the base diffusion model was EDM and the NFE of the EDM (Heun) sampler was set to 35. For all other cases, we trained the discriminator ourselves using the DG codebase.

Our discriminators were trained on a single NVIDIA RTX 3090 GPU. For the feature extractor, we used a commonly adopted (Na et al., 2024; Kim et al., 2023) pre-trained classifier from ADM¹⁰ (Dhariwal & Nichol, 2021). The discriminator takes the features extracted by this network as input, and during training, we only updated the parameters of the discriminator network. Detailed training settings are provided in Table 20.

Indeed, while our method requires training an additional discriminator, we would like to emphasize that the training cost is significantly lower compared to that of the score model. As summarized in the table 21, our discriminator is much smaller and faster to train than the pre-trained score network. Compared to fine-tuning a pre-trained

Table 19: FID-NFE with different training epochs of discriminator.

Method	Epoch	$FID{\downarrow}$	$NFE \!\!\downarrow$
EDM	-	3.23	17.00
AC w.o. MH	-	3.40	15.26
+AC	1	2.56	15.80
	2	2.59	15.80
	5	2.64	15.81
	10	2.39	15.81
	20	2.43	15.82
	60	2.38	15.81

¹⁰https://github.com/openai/guided-diffusion

1350 1351

Table 20: Configurations of the discriminator.

1	3	5	2
1	3	5	3
1	3	5	4

1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366

1367 1369 1370 1371

generated samples # Epoch

1372 1373 1374

1375 1376 1377

1378

1379 1380 1381

1382

1399 1400 1401

1402

1403

Η

	_	_	
	3	9	3
l	3	9	4
l	3	9	5
l	3	9	6
l	3	9	7
ı	3	9	8

We acknowledge the use of a Large Language Model (LLM) during the preparation of this manuscript. The LLM was employed solely as a

THE USE OF LARGE LANGEUAGE MODELS (LLMS)

general-purpose writing assistant to improve readability, grammar, and clarity of exposition. It was not involved in the ideation of research questions, the design of experiments, the development of

CIFAR-10 ImageNet 64×64 CelebA-HQ 256×256 ImageNet 256×256 Diffusion Backbone DiT-XL/2 **EDM** DDO ScoreSDE **EDM** ScoreSDE Model Conditional model Feature Extractor ADM ADM ADM ADM ADM ADM Architecture U-Net encoder U-Net encoder U-Net encoder U-Net encoder U-Net encoder U-Net encoder Pre-trained 4 4 Depth 128 128 128 128 Attention Resolutions 32,16,8 32,16,8 32,16,8 32,16,8 32,16,8 32,16,8 (B 32 32 3) (B 32 32 3) (B 32 32 3) (B 256 256 3) (B 32 32 3) Input shape (data) (B 64 64 3) Output shape (feature) (B,8,8,512)(B,8,8,512)(B,8,8,512) (B,8,8,512)(B,8,8,512)(B,8,8,384) Discriminator Model ADM ADM ADM ADM ADM ADM Architecture U-Net encoder U-Net encoder U-Net encoder U-Net encoder U-Net encoder U-Net encoder Depth 128 128 128 128 128 128 Width Attention Resolutions 32,16,8 32,16,8 32,16,8 32,16,8 32,16,8 32,16,8 (B,8,8,512) Input shape (feature) (B,8,8,512)(B,8,8,512)(B,8,8,512)(B,8,8,512)(B,8,8,384)Output shape (logit) (B,1)(B,1)(B,1)(B,1)(B,1)(B,1)Discriminator Training VP VP VP VP VP Time scheduling Cosine VP Time sampling Importance Importance Importance Importance Importance Importance Time weighting $\frac{g^2}{\sigma^2}$ 128 128 128 128 5^{12} Batch size 30,000 50,000 # data samples 50,000 50,000 50,000 50,000

Table 21: Training cost comparison of the score model and discriminator. CIFAR-10

50,000

20

30,000

50

50,000

50

50,000

60

Training	Parameter Size	Training GPU	Training Time
Score (EDM)	55.7M	8×V100 GPUs	~2 days
Discriminator	2.9M	1×RTX 3090	<2 hours

diffusion model, our approach introduces substantially lower com-

50,000

70

50,000

60

putational overhead and does not modify the pre-trained model in any way. Importantly, we enable both acceleration and correction purely through discriminator training only. Table.22 shows the robustness of our time-dependent discriminator. To evaluate the performance of our discriminator at different timesteps, we conducted experiments with $n_{\text{chain}} = 1$ ensuring that each sample is corrected independently. (No acceleration was applied, as our goal was to isolate the effect of the discriminator across timesteps) Using a fixed SNR, we applied the AC-Sampler at various τ values with the same random seed. The results consistently showed improvements in FID across timesteps, suggesting that the discriminator effectively approximates the density ratio $\frac{q_{\tau}}{p_{\theta}^{\theta}}$ at multiple temporal locations.

To evaluate the robustness of our method under an imperfect discriminator, we conducted experiments using partially trained discriminators. We observed that as the discriminator training progressed, the quality of the generated samples consistently improved. Moreover, our method significantly outperformed where all proposals are accepted without a discriminator. These results indicate that even an imperfectly trained discriminator can still yield meaningful performance gains. This supports the theoretical claim in Appendix A.4, where we show that continued discriminator training leads to improvements in KL divergence.

Table 22: FID and NFE comparison of AC with different chain lengths.

Method	τ	FID↓	NFE↓
EDM	-	2.01	35
+AC	3	2.02	48.51
	5	1.97	48.53
	7	1.99	48.54
	10	2.00	48.72
	12	1.92	49.89
	15	2.00	49.03

methods, or the interpretation of results. The scientific contributions of this work, including problem formulation, methodology, theoretical analysis, and empirical evaluation, were conceived and carried out entirely by the authors. The role of the LLM was limited to helping refine the presentation of the text, and it did not contribute substantively to the research process itself.

I GENERATED IMAGES

We provide uncurated sample images at Figure 8, 9, 10, 11, 12 generated by our sampler. These are non-cherry-picked samples generated by applying our algorithm to the baseline models and samplers, used in our experiments.



Figure 8: The uncurated generated images of AC-sampler on unconditional CIFAR-10 with EDM (EDM(Heun) sampler, NFE=26.19, FID=1.97).

Figure 9: The uncurated generated images of AC-sampler on unconditional CIFAR-10 with ScoreSDE (KAR1 sampler, NFE=29.07, FID=23.14).

Figure 10: The uncurated generated images of AC-sampler on conditional ImageNet 64×64 with EDM (EDM(SDE) sampler, NFE=59.30, FID=2.27).



Figure 11: The uncurated generated images of AC-sampler on unconditional CelebA-HQ 256×256 with ScoreSDE (KAR1 sampler, NFE=15.94, FID=15.13).



Figure 12: The uncurated generated images of AC-sampler on unconditional CelebA-HQ 256×256 with ScoreSDE(KAR2 sampler, NFE=20.05, FID=8.45).