Fast Solvers for Discrete Diffusion Models: Theory and Applications of High-Order Algorithms

Yinuo Ren[1,*] **Haoxuan Chen**[1,*,†] **Yuchen Zhu**[2,*] **Wei Guo**[2,*]
Yongxin Chen[2] **Grant M. Rotskoff**[1] **Molei Tao**[2] **Lexing Ying**[1]
[1]Stanford University [2]Georgia Institute of Technology
{yinuoren, haoxuanc, rotskoff, lexing}@stanford.edu
{yzhu738, wei.guo, yongchen, mtao}@gatech.edu

Abstract

Discrete diffusion models have emerged as a powerful generative modeling framework for discrete data with successful applications spanning from text generation to image synthesis. However, their deployment faces challenges due to the high dimensionality of the state space, necessitating the development of efficient inference algorithms. Current inference approaches mainly fall into two categories: exact simulation and approximate methods such as τ-leaping. While exact methods suffer from unpredictable inference time and redundant function evaluations, τ-leaping is limited by its first-order accuracy. In this work, we advance the latter category by tailoring the first extension of high-order numerical inference schemes to discrete diffusion models, enabling larger step sizes while reducing error. We rigorously analyze the proposed schemes and establish the second-order accuracy of the θ-Trapezoidal method in KL divergence. Empirical evaluations on GSM8K-level math-reasoning, GPT-2-level text, and ImageNet-level image generation tasks demonstrate that our method achieves superior sample quality compared to existing approaches under equivalent computational constraints, with consistent performance gains across models ranging from 200M to 8B. Our code is available at `https://github.com/yuchen-zhu-zyc/DiscreteFastSolver`.

1 Introduction

Diffusion and flow-based models on discrete spaces [1–10] have emerged as a cornerstone of modern generative modeling for categorical data, offering unique advantages in domains where continuity assumptions fail. Unlike their continuous counterparts, discrete diffusion models inherently accommodate data with discrete structures, *e.g.*, language tokens, molecular sequences, tokenized images, and graphs, enabling principled generation and inference in combinatorially complex spaces. These models have exerted a large impact on numerous applications, from the design of molecules [11], proteins [12], and DNA sequences [13, 14] under biophysical constraints, to the generation of high-fidelity text [15] and images [16] via autoregressive or masked transitions, *etc.*. Beyond standalone tasks, discrete diffusion models also synergize with methodologies, ranging from tensor networks [17] to guidance mechanisms [18–20].

Discrete diffusion models, despite their broad applicability, face a critical bottleneck: *inference inefficiency*. Current inference methods include: (1) exact simulation methods [21], which ensure unbiased sampling from the pre-trained model but suffer from unpredictable inference time and redundant score evaluations, resulting in poor scaling w.r.t. dimensionality; and (2) approximate

[*]Equal contribution
[†] Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

methods such as τ-leaping [22], which offer simple and parallelizable implementation but, due to their first-order accuracy, requires small step sizes to control discretization error, forcing a stringent trade-off between speed and sample quality.

To address these limitations in possibly computationally constrained environments, we develop high-order numerical schemes tailored for discrete diffusion model inference. Drawing inspirations from acceleration techniques developed for ordinary differential equations (ODEs) [23], stochastic differential equations (SDEs) [24, 25], chemical reaction simulations [26], and most recently continuous diffusion [27–29], our work represents the *first successful adaptation of high-order numerical schemes to the discrete diffusion domain*. Through careful design, these high-order schemes provide unprecedented efficient and versatile solutions for discrete diffusion model inference.

Our Contributions. The main contributions of this paper are summarized as follows:

- We introduce the *first high-order numerical solvers* for discrete diffusion model inference, namely the θ-Runge-Kutta-2 (θ-RK-2) method and the θ-Trapezoidal method;
- We rigorously establish the theoretical properties of both methods, proving *second-order convergence* of the θ-Trapezoidal method and *conditional second-order convergence* of the θ-RK-2 method;
- We empirically validate our theoretical results and demonstrate the *superior performance* of the θ-Trapezoidal method through comprehensive evaluations on large-scale text and image generation benchmarks.

1.1 Related Works

Here we briefly review related works and defer a more detailed discussion to App. A.

Discrete Diffusion Models. Since their introduction, discrete diffusion models have undergone significant refinements, including the development of score-entropy loss [30] and flow-matching formulation [31, 32]. These models generally fall into two categories based on their noise distribution: uniform [30, 20] and masked (absorbing state) [33–35, 21], each offering unique advantages in modeling discrete distributions. Recent theoretical advances have emerged through studies [36–38].

High-Order Scheme for Continuous Diffusion Models. The development of high-order numerical schemes for solving ODEs and SDEs represents decades of research, as comprehensively reviewed in [23, 39, 40]. These schemes have recently been adapted to accelerate continuous diffusion model inference, encompassing approaches such as the exponential integrators [41–43], Adams-Bashforth methods [29, 44, 45], Taylor methods [27, 46] and (stochastic) Runge-Kutta methods [47, 28, 48–51].

High-Order Scheme for Chemical Reaction Systems. Regarding approximate methods for simulating compound Poisson processes and chemical reaction systems with state-dependent intensities, efforts have been made on the τ-leaping method [52], and its extensions [53, 54, 26, 55, 56]. For a quick review of the problem setting and these methods, one may refer to [57, 58]. The adaptation of these methods to discrete diffusion models presents unique challenges due to the presence of both time and state-inhomogeneous intensities in the underlying Poisson processes.

2 Preliminaries

In this subsection, we review several basic concepts and previous error analysis results of discrete diffusion models.

2.1 Discrete Diffusion Models

In discrete diffusion models, one considers a continuous-time Markov chain (CTMC) $(\boldsymbol{x}_t)_{0\leq t\leq T}$ on a finite space $\mathbb{X}$ as the *forward process*. We represent the distribution of $\boldsymbol{x}_t$ by a vector $\boldsymbol{p}_t \in \Delta^{|\mathbb{X}|}$, where $\Delta^{|\mathbb{X}|}$ denotes the probability simplex in $\mathbb{R}^{|\mathbb{X}|}$. Given a target distribution $\boldsymbol{p}_0$, the CTMC

satisfies the following equation:

$$\frac{\mathrm{d}\boldsymbol{p}_t}{\mathrm{d}t} = \boldsymbol{Q}_t \boldsymbol{p}_t, \quad \text{where } \boldsymbol{Q}_t = (Q_t(y,x))_{x,y\in\mathbb{X}} \tag{2.1}$$

is the rate matrix at time t satisfying

$$\text{(i) } Q_t(x,x) = -\sum_{y\neq x} Q_t(y,x),\ \forall x \in \mathbb{X};\ \text{(ii) } Q_t(x,y) \geq 0,\ \forall x \neq y \in \mathbb{X}.$$

Below we will use the notation $\boldsymbol{Q}_t^0 = \boldsymbol{Q}_t - \operatorname{diag} \boldsymbol{Q}_t$. It can be shown that the corresponding backward process is of the same form but with a different rate matrix [59]:

$$\frac{\mathrm{d}\overleftarrow{\boldsymbol{p}}_s}{\mathrm{d}s} = \overline{\boldsymbol{Q}}_s \overleftarrow{\boldsymbol{p}}_s, \quad \text{where } \overline{Q}_s(y,x) = \begin{cases} \frac{\overleftarrow{p}_s(y)}{\overleftarrow{p}_s(x)} \overleftarrow{Q}_s(x,y), & \forall x \neq y \in \mathbb{X}, \\ -\sum_{y'\neq x} \overline{Q}_s(y',x), & \forall x = y \in \mathbb{X}. \end{cases} \tag{2.2}$$

is the rate matrix and $\overleftarrow{*}_s$ denotes $*_{T-s}$. The rate matrix $\boldsymbol{Q}_t$ is often chosen to possess certain sparse structures such that the forward process converges to a simple distribution that is easy to sample from. Popular choices include the uniform and absorbing state cases [30], where the forward process (2.1) converges to the uniform distribution on $\mathbb{X}$ and a Dirac distribution, respectively.

Common training practice is to define the score function (or the score vector) as $\boldsymbol{s}_t(x) = (s_t(x,y))_{y\in\mathbb{X}} := \frac{\boldsymbol{p}_t}{p_t(x)}$ for any $x \in \mathbb{X}$, $t \in [0,T]$ and estimate it by a neural network $\widehat{\boldsymbol{s}}_t^\phi(x)$, where the parameters ϕ are trained by minimizing the score entropy [30, 60] for some weights $\psi_t \geq 0$ as:

$$\min_\phi \int_0^T \psi_t \mathbb{E}_{x_t\sim p_t}\Big[\sum_{y\neq x_t} Q_t(x_t,y)\Big(s_t(x_t,y)\log\tfrac{s_t(x_t,y)}{\widehat{s}_t^\phi(x_t,y)} - s_t(x_t,y) + \widehat{s}_t^\phi(x_t,y)\Big)\Big]\mathrm{d}t. \tag{2.3}$$

Similar to the continuous case, the backward process is approximated by another CTMC $\frac{\mathrm{d}\boldsymbol{q}_s}{\mathrm{d}s} = \widehat{\overline{\boldsymbol{Q}}}_s^\phi \boldsymbol{q}_s$, with $\boldsymbol{q}_0 = \boldsymbol{p}_\infty$ and rate matrix $\widehat{\overline{\boldsymbol{Q}}}_s^\phi$, where $\widehat{\overline{Q}}_s^\phi(y,x) = \overleftarrow{\widehat{s}}_s^\phi(x,y)\overleftarrow{Q}_s(x,y)$ for any $x \neq y \in \mathbb{X}$. The inference is done by first sampling from $\boldsymbol{p}_\infty$ and then evolving the CTMC accordingly. For simplicity, we drop the superscript ϕ hereafter.

2.2 Stochastic Integral Formulation of Discrete Diffusion Models

Discrete diffusion models can also be formulated as stochastic integrals, which is especially useful for their theoretical analysis [38]. In this section, we briefly recapitulate the relevant results and refer to App. B for the mathematical details. Below, we work on the probability space $(\Omega, \mathcal{B}, \mathbb{P})$ and denote the pairwise difference set of the state space $\mathbb{X}$ by $\mathbb{D} := \{x - y : x \neq y \in \mathbb{X}\}$. In this work, we focus on the case where $\mathbb{X} = [S]^d$ with d data dimensions and S sites along each dimension.

We first introduce the Poisson random measure, a key concept in the formulation.

Definition 2.1 (Informal Definition of Poisson Random Measure)**.** *The random measure $N[\lambda](\mathrm{d}t, \mathrm{d}\nu)$ on $\mathbb{R}^+ \times \mathbb{D}$ is called a* Poisson random measure *with* evolving intensity λ *w.r.t. a measure γ on $\mathbb{D}$ if, roughly speaking, the number of jumps of magnitude ν during the infinitesimal time interval $(t, t+\mathrm{d}t]$ is Poisson distributed with mean $\lambda_t(\nu)\gamma(\mathrm{d}\nu)\mathrm{d}t$.*

The forward process (2.1) can thus be represented by the following stochastic integral:

$$x_t = x_0 + \int_0^t \int_{\mathbb{D}} \nu N[\lambda](\mathrm{d}s, \mathrm{d}\nu),$$

where the intensity λ is defined as $\lambda_t(\nu,\omega) = Q_t^0(x_{t^-}(\omega) + \nu, x_{t^-}(\omega))$ if $x_{t^-}(\omega) + \nu \in \mathbb{X}$ and 0 otherwise. Here, the outcome $\omega \in \Omega$ and x_{t^-} denotes the left limit of the càdlàg process x_t at time t with $x_{0^-} = x_0$. We will also omit the variable ω, should it be clear from context. The backward process in discrete diffusion models (2.2) can also be represented similarly as:

$$y_s = y_0 + \int_0^s \int_{\mathbb{D}} \nu N[\mu](\mathrm{d}s, \mathrm{d}\nu), \tag{2.4}$$

where the intensity μ is defined as $\mu_s(\nu,\omega) = \overleftarrow{s}_s(y_{s^-}, y_{s^-}+\nu)\overleftarrow{Q}_s^0(y_{s^-}, y_{s^-}+\nu)$ if $y_{s^-} + \nu \in \mathbb{X}$ and 0 otherwise. During inference, $\widehat{y}_s = \widehat{y}_0 + \int_0^s \int_{\mathbb{D}} \nu N[\widehat{\mu}](\mathrm{d}s, \mathrm{d}\nu)$ is used instead of (2.4), where the estimated intensity $\widehat{\mu}$ is defined by replacing the true score $\boldsymbol{s}_t$ with the neural network estimated score $\widehat{\boldsymbol{s}}_t$ in $\mu_s(\nu,\omega)$. . In the following, we also denote the intensity $\mu_s(\nu,\omega)$ at time s by $\mu_s(\nu, y_{s^-})$ with slight abuse of terminology to emphasize its dependency on ω through $y_{s^-}(\omega)$.

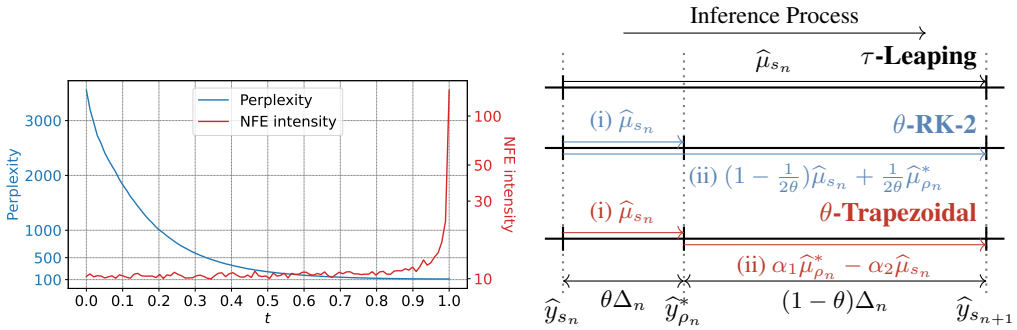

Figure 1: **Left:** Application of the uniformization algorithm to discrete diffusion models for text generation. The x-axis denotes the time of the backward process, and the y-axis denotes the frequency of jumps (NFE). Perplexity convergence occurs before the NFE grows unbounded. **Right:** Comparison between τ-leaping and the proposed second-order schemes (θ-RK-2 and θ-Trapezoidal).

3 Numerical Schemes for Discrete Diffusion Model Inference

Before introducing the proposed numerical schemes, we first review existing numerical schemes for discrete diffusion models, including exact simulation methods and the τ-leaping method, and discuss their merits and limitations.

3.1 Exact Simulation Methods

Unlike in continuous diffusion models, where exact simulation is infeasible, discrete diffusion models permit inference without discretization error. Notable examples of unbiased samplers include uniformization [36] for the uniform state case and the First-Hitting Sampler (FHS) [21] for the absorbing state case. The main idea behind these methods is to first sample the next jump time and then the jump itself. Theoretical analysis [38] reveals that such schemes *lack guarantees with finite computation budget*, since the number of required jumps (and thus the inference time) follows a random distribution with expectation $\Omega(d)$. This computational restriction may be less favorable for high-dimensional applications, such as generative modeling of DNA or protein sequences.

Furthermore, *the absence of discretization error does not necessarily translate to superior sample quality*, given the inherent estimation errors in neural network-based score functions. This limitation is further amplified by the *highly skewed distribution* of jumps, with a concentration occurring during the terminal phase of the backward process, when the neural network-based score function exhibits the highest estimation error. This phenomenon stems from the potential singularity of the target distribution $\boldsymbol{p}_0$, which induces singularities in the score function, making accurate neural network estimation particularly challenging during that phase (*cf.* Assump. 4.4 [38]).

The left figure in Fig. 1 illustrates an application of the uniformization algorithm to discrete diffusion inference for text generation, with detailed experimental parameters presented in Sec. 6.3 and App. D.3. As the process approaches the target distribution ($t \to T$), the number of jumps (in terms of the number of score function evaluations, NFE) grows unbounded, while perplexity improvements become negligible. This skew in computational effort leads to *redundant function evaluations*. Although early stopping is commonly adopted at $T-\delta$ for some small $\delta \ll 1$ to alleviate this inefficiency, this approach introduces challenges in its selection, particularly under computational constraints or when efficiency-accuracy trade-offs are desired. Moreover, the variable jump schedules across batch samples complicate parallelization efforts in exact methods, highlighting the need for more adaptable and efficient algorithmic solutions.

3.2 Approximate Method: τ-Leaping Method

The τ-leaping method [52, 22] is a widely adopted scheme that effectively addresses both dimensionality scaling and inference-time control challenges. This Euler-type scheme approximates the backward process with time-dependent intensity $\widehat{\mu}_t$ via the following updates:

$$\widehat{y}_{t+\Delta} = \widehat{y}_t + \sum_{\nu\in\mathbb{D}} \nu \mathcal{P}\left(\widehat{\mu}_t(\nu)\Delta\right), \tag{3.1}$$

where Δ denotes the time step and $\mathcal{P}(\cdot)$ denotes a Poisson random variable. In general, one may design different discretization schemes for τ-leaping, and the summation in (3.1) is parallelizable, underscoring the method's flexibility and efficiency. We refer to Alg. 3 and App. B.2 for a detailed description of the τ-leaping method for discrete diffusion model inference. Regarding convergence properties as the time discretization becomes increasingly refined, theoretical analyses by [22, 38] have established the error bounds of the τ-leaping method, the results of which are summarized in the following theorem. Further discussion can be found in App. B.2.

Theorem 3.1 (Thm. 4.7 in [38])**.** *Under a certain discretization scheme and technical assumptions, and given an ϵ-accurate score function, the following error bound holds:*

$$D_{\mathrm{KL}}(p_\delta \| \widehat{q}_{T-\delta}) \lesssim \exp(-T) + \epsilon + \kappa T, \tag{3.2}$$

where $\delta \ll 1$ is the early stopping time, κ controls the step size, and T is the time horizon. The notation $\lesssim$ indicates the inequality holds up to a constant factor as $\kappa \to 0$.

The error bound (3.2) decouples three error sources of the τ-leaping scheme: the truncation error $\mathcal{O}(e^{-T})$, the score estimation error ϵ, and the discretization error $\mathcal{O}(\kappa T)$. Similar to the case for the Euler method for ODEs and the Euler-Maruyama scheme for SDEs, the τ-leaping method is a first-order scheme in terms of the discretization error $\mathcal{O}(\kappa T)$.

4 Algorithms: High-Order Inference Schemes

A natural improvement of τ-leaping is to develop high-order schemes for discrete diffusion models. As a foundational example, consider the second-order Runge-Kutta (RK-2) method with two stages [23] for solving the ODE $\mathrm{d}x_t = f_t(x_t)\mathrm{d}t$. This method represents one of the simplest high-order numerical schemes:

$$\widehat{x}^*_{t+\theta\Delta} = \widehat{x}_t + f_t(\widehat{x}_t)\theta\Delta, \quad \widehat{x}_{t+\Delta} = \widehat{x}_t + \left[(1-\tfrac{1}{2\theta})f_t(\widehat{x}_t) + \tfrac{1}{2\theta} f_{t+\theta\Delta}(\widehat{x}^*_{t+\theta\Delta})\right]\Delta. \tag{4.1}$$

This scheme reduces to the exact midpoint method when $\theta = \frac{1}{2}$ and Heun's method when $\theta = 1$. The underlying intuition stems from the observation that for $f \in C^2(\mathbb{R})$, $\left[\left(1-\frac{1}{2\theta}\right) f(0) + \frac{1}{2\theta} f(\theta\Delta)\right]\Delta$ offers a second-order approximation of $\int_0^\Delta f(x)\mathrm{d}x$ in contrast to $f(0)\Delta$, which is only first-order. This approach has been successfully adapted for SDE simulation [24] and continuous diffusion model inference [48, 28, 29, 49, 51]. Notably, these methods enhance sample quality and computational efficiency without requiring additional model training, making the development of high-order schemes for discrete diffusion inference both theoretically appealing and practically viable.

In this section, we propose two high-order solvers for inference in the discrete diffusion model. We will primarily focus on two-stage algorithms aiming for second-order accuracy. Specifically, we will introduce the θ-RK-2 and θ-Trapezoidal methods. Throughout this section, we assume a time discretization scheme $(s_i)_{i\in[0:N]}$ with $0 = s_0 < \cdots < s_N = T - \delta$, where δ is the early stopping time and use the shorthand notations $*_+ = \max\{0, *\}$. For any $s \in (s_n, s_{n+1}]$ and $n \in [0 : N-1]$, we define $\lfloor s \rfloor = s_n$, $\rho_s = (1-\theta)s_n + \theta s_{n+1}$, $\Delta_n = s_{n+1} - s_n$, and *θ-section points* as $\rho_n = (1-\theta)s_n + \theta s_{n+1}$. We choose $\gamma(\mathrm{d}\nu)$ to be the counting measure on $\mathbb{D}$.

4.1 θ-RK-2 Method

We first present the *θ-RK-2 method*, which is simple in design and serves as a natural analog of the second-order RK method for ODEs (4.1) in terms of time and state-dependent Poisson random measures, as a warm-up for the θ-Trapezoidal method. We note that similar methods have been proposed for simulating SDEs driven by Brownian motions or Poisson processes, such as the stochastic [24] and the Poisson [54] RK methods. A summary of this method is given in Alg. 1.

Algorithm 1: θ-RK-2 Method

Input: $\widehat{y}_0 \sim q_0$, $\theta \in (0, 1]$, $(s_n, \rho_n)_{n\in[0:N-1]}$, $\widehat{\mu}$, $\widehat{\mu}^*$.
Output: A sample $\widehat{y}_{s_N} \sim \widehat{q}^{\mathrm{RK}}_{t_N}$.

for $n = 0$ **to** $N-1$ **do**
  $\widehat{y}^*_{\rho_n} \leftarrow \widehat{y}_{s_n} + \sum_{\nu\in\mathbb{D}} \nu\mathcal{P}\left(\widehat{\mu}_{s_n}(\nu)\theta\Delta_n\right)$;
  $\widehat{y}_{s_{n+1}} \leftarrow \widehat{y}_{s_n} +$
    $\sum_{\nu\in\mathbb{D}} \nu\mathcal{P}\left(\mathbf{1}_{\widehat{\mu}_{s_n}>0}\left[\left(1-\frac{1}{2\theta}\right)\widehat{\mu}_{s_n} + \frac{1}{2\theta}\widehat{\mu}^*_{\rho_n}\right]_+(\nu)\Delta_n\right)$;
end

Intuitively, the θ-RK-2 method is a two-stage algorithm that:

(i) Firstly, it runs τ-leaping with step size $\theta\Delta_n$, obtains an *intermediate state* $\widehat{y}^*_{\rho_n}$ at the θ-section point ρ_n, and evaluates the intensity $\widehat{\mu}^*_{\rho_n}$ there;
(ii) Then another step of τ-leaping for a full step Δ_n is run using a weighted sum of the intensities at the current time point s_n and the θ-section point ρ_n.

We emphasize that our method differs from the midpoint method proposed in [52] for simulating chemical reactions, in which the Poisson random variable in the first step is replaced by its expected value. Such modification is in light of the lack of continuity and orderliness of the state space.

4.2 θ-Trapezoidal Method

As to be shown theoretically and empirically, the conceptually simple θ-RK-2 method may have limitations in terms of both accuracy and efficiency. To this end, we propose the following *θ-Trapezoidal method*, which is developed based on existing methods proposed for simulating SDEs [25] and chemical reactions [26]. Below, we introduce two parameters that will be used extensively later:

$$\alpha_1 = \tfrac{1}{2\theta(1-\theta)} \text{ and } \alpha_2 = \tfrac{(1-\theta)^2+\theta^2}{2\theta(1-\theta)}, \text{ with } \alpha_1 - \alpha_2 = 1.$$

The θ-Trapezoidal method is summarized in Alg. 2. Intuitively, it separates each interval $(s_n, s_{n+1}]$ into two sub-intervals $(s_n, \rho_n]$ and $(\rho_n, s_{n+1}]$, on which simulations are detached with different intensities designed in a balanced way.

Algorithm 2: θ-Trapezoidal Method

Input: $\widehat{y}_0 \sim q_0$, $\theta \in (0, 1]$, $(s_n, \rho_n)_{n\in[0:N-1]}$, $\widehat{\mu}, \widehat{\mu}^*$.
Output: A sample $\widehat{y}_{s_N} \sim \widehat{q}^{\text{trap}}_{t_N}$.

```
for n = 0 to N − 1 do
    ŷ*_{ρ_n} ← ŷ_{s_n} + Σ_{ν∈𝔻} νP(μ̂_{s_n}(ν)θΔ_n);
    ŷ_{s_{n+1}} ← ŷ*_{ρ_n} +
        Σ_{ν∈𝔻} νP((α_1 μ̂*_{ρ_n} − α_2 μ̂_{s_n})_+ (ν)(1 − θ)Δ_n);
end
```

Compared to the θ-RK-2 method, the θ-Trapezoidal method is also two-stage with an identical first step. The second step, however, differs in two major aspects:

(1) The second step starts from the intermediate state $\widehat{y}^*_{\rho_n}$ instead of $\widehat{y}_{s_n}$ and only runs for a fractional step $(1-\theta)\Delta_n$ rather than a full step Δ_n;
(2) The weighted sum is comprised of an altered pair of coefficients $(\alpha_1, -\alpha_2)$, performing an *extrapolation* instead of interpolation with coefficients $(1 - \frac{1}{2\theta}, \frac{1}{2\theta})$ as in the θ-RK-2 method with $\theta \in [\frac{1}{2}, 1]$. This feature will be shown to render the algorithm unconditionally second-order.

Following the common practice in the literature [22], we reject updates with multiple jumps along one dimension in both algorithms, ensuring their well-posedness. A simple analysis shows that rejection only happens with probability $\mathcal{O}(\kappa)$, and we refer to further details in Rmk. C.4. We refer to Props. C.2 and C.3 for the stochastic integral formulations of these two algorithms. We provide a visual comparison between the θ-RK-2 and the θ-Trapezoidal method in the right figure of Fig. 1.

5 Theoretical Analysis

In this section, we provide the theoretical results of the θ-Trapezoidal and θ-RK-2 methods. The goal of this section is to show that under certain conditions, both methods are second-order accurate, improving from the first-order accuracy of the τ-leaping method (*cf.* Thm. 3.1). Our theoretical analysis also reveals that the θ-Trapezoidal method is more robust to the choice of θ than θ-RK-2, to be confirmed by our empirical results in Sec. 6.

5.1 Assumptions

For simplicity, we impose a periodic boundary condition on the state space $\mathbb{X} = [S]^d$, *i.e.*, embed the state space in the d-dimensional torus $\mathbb{T}^d$, to streamline the proofs (*cf.* Rmk. C.4).

Assumption 5.1 (Convergence of Forward Process)**.** *The forward process converges to the stationary distribution exponentially fast,* i.e.*,* $D_{\mathrm{KL}}(p_T \| p_\infty) \lesssim \exp(-T)$.

This assumption ensures rapid convergence of the forward process, controlling error when terminated at a sufficiently large time horizon T, and is automatically satisfied in the masked state case

and the uniform state case, given sufficient connectivity of the graph (*cf.* [38]). The exponential rate aligns with continuous diffusion models (*cf.* [61]).

Assumption 5.2 (Regularity of Intensity)**.** *For the true intensity $\mu_s(\nu, y_{s^-})$ and the estimated intensity $\widehat{\mu}_s(\nu, y_{s^-})$, it holds almost everywhere w.r.t. $\mu_s(\nu, y_{s^-})\gamma(\mathrm{d}\nu)\overleftarrow{p}_{s^-}(\mathrm{d}y_{s^-})$ that: (1) Both intensities belong to $C^2([0, T-\delta])$; (2) Both intensities are upper and lower bounded on $[0, T-\delta]$.*

This assumes two key requirements of the scores: (1) the forward process maintains sufficient smoothness, which is achievable through appropriate time reparametrization; and (2) if and only if a state $y \in \mathbb{X}$ is achievable by the forward process and ν is a permissible jump therefrom, then both its true and estimated intensity are bounded, corresponding to Assumps. 4.3(i), 4.4, and 4.5 [38].

Assumption 5.3 (Estimation Error)**.** *For all grid points and θ-section points, the estimation error of the neural network-based score is small,* i.e.*, for any $s \in \cup_{n\in[0:N-1]}\{s_n, \rho_n\}$, we have*

(1) $\mathbb{E}\left[\int_{\mathbb{D}}\left(\mu_s(\nu)\left(\log\frac{\mu_s(\nu)}{\widehat{\mu}_s(\nu)}-1\right)+\widehat{\mu}_s(\nu)\right)\gamma(\mathrm{d}\nu)\right] \le \epsilon_{\mathrm{I}}$;

(2) $\mathbb{E}\left[\int_{\mathbb{D}}|\mu_s(\nu)-\widehat{\mu}_s(\nu)|\,\gamma(\mathrm{d}\nu)\right] \le \epsilon_{\mathrm{II}}$.

This assumption quantifies the proximity of the estimated intensity $\widehat{\mu}$ to the true intensity μ after sufficient training. Compared with [38], the additional L^∞ part in (2) is required for technical reasons, which is similar to [62, 51]. In practice, such additional assumptions may be realized by adding extra penalty terms to the objective function during training.

5.2 Convergence Guarantees

The following theorem summarizes our theoretical guarantees for the θ-Trapezoidal method:

Theorem 5.4 (Second Order Convergence of θ-Trapezoidal Method)**.** *Suppose $\theta \in (0, 1]$ and $\alpha_1\widehat{\mu}^*_{\rho_s} - \alpha_2\widehat{\mu}_{\lfloor s\rfloor} \ge 0$ for all $s \in [0, T-\delta]$, then the following error bound holds for Alg. 2 under Assumps. 5.1 to 5.3:*

$$D_{\mathrm{KL}}(p_\delta \| \widehat{q}^{\mathrm{trap}}_{T-\delta}) \lesssim \exp(-T) + (\epsilon_{\mathrm{I}} + \epsilon_{\mathrm{II}})T + \kappa^2 T,$$

where δ is the early stopping time, $\kappa = \max_{n\in[0:N-1]}\Delta_n$, i.e.*, the largest stepsize, and $\widehat{q}^{\mathrm{trap}}_{T-\delta}$ is the distribution obtained by Alg. 1 as defined in Prop. C.2.*

The complete proof is presented in App. C.2. The outline is to first bound $D_{\mathrm{KL}}(p_\delta\|\widehat{q}^{\mathrm{trap}}_{T-\delta})$ by the KL divergence between the corresponding path measures, as established in Thm. C.5, and then decompose the integral in the log-likelihood and bound respectively, where the primary technique used is *Dynkin's formula* (Thm. C.10). With a term-by-term comparison with Thm. 3.1, we observe a significant improvement in the discretization error term from $\mathcal{O}(\kappa T)$ to $\mathcal{O}(\kappa^2 T)$. This confirms that the θ-Trapezoidal method achieves second-order accuracy given a sufficient time horizon T and accurate score estimation, with empirical validation presented in Sec. 6.

Theorem 5.5 (Conditional Second-Order Convergence of θ-RK-2 Method)**.** *Suppose $\theta \in (0, \frac{1}{2}]$ and $(1-\frac{1}{2\theta})\widehat{\mu}_{\lfloor s\rfloor} + \frac{1}{2\theta}\widehat{\mu}^*_{\rho_s} \ge 0$ for all $s \in [0, T-\delta]$, then the following error bound holds for Alg. 1 under Assumps. 5.1 to 5.3:*

$$D_{\mathrm{KL}}(p_\delta \| \widehat{q}^{\mathrm{RK}}_{T-\delta}) \lesssim \exp(-T) + (\epsilon_{\mathrm{I}} + \epsilon_{\mathrm{II}})T + \kappa^2 T,$$

where δ is the early stopping time, $\kappa = \max_{n\in[0:N-1]}\Delta_n$, i.e.*, the largest stepsize, and $\widehat{q}^{\mathrm{RK}}_{T-\delta}$ is the distribution obtained by Alg. 2 as defined in Prop. C.3.*

The proof of the theorem above is provided in App. C.3. The restricted range of θ is caused by one specific error term (III.4) (C.9) that permits bounding with *Jensen's inequality* only when $\theta \in (0, \frac{1}{2}]$, similar to its counterpart (II.4) (C.11) in the θ-Trapezoidal method. The limitation arises partially because the weighted sum with coefficients $(1-\frac{1}{2\theta}, \frac{1}{2\theta})$ becomes an *extrapolation* only if $1-\frac{1}{2\theta} < 0$, a feature that naturally holds for all $\theta \in (0, 1]$ in the θ-Trapezoidal method. These theoretical findings are consistent with the empirical observations in Fig. 6 of App. D.3, where the performance of θ-RK-2 method clearly peaks when $\theta \in (0, \frac{1}{2}]$.

Remark 5.6 (Comparison between Trapezoidal and RK-2 Methods)**.** *Trapezoidal methods were originally proposed by [25] as a minimal second-order scheme in the weak sense for simulating SDEs. In simulating chemical reaction contexts, [26] claimed that trapezoidal methods also achieve second-order convergence for* covariance error *apart from the weak error, a property not shared by midpoint (RK-2) methods. Our empirical results partly reflect these findings, while we defer theoretical investigation of covariance error convergence in discrete diffusion models to future work.*

Remark 5.7 (Remark on the Positivity of Extrapolated Intensity)**.** *Due to the nature of extrapolation, both our theorems require an additional assumption on the positivity of the extrapolated intensity, which is classically assumed in [25, 26], and resolving this issue is a long-standing open problem. The best result so far is Prop. 5 [26], claiming clamping the intensity above 0 only causes an error of order $\mathcal{O}(\kappa^p)$, for any large integer p. We empirically demonstrate the validity of this assumption Tab. 6 in practice through the text generation task (Sec. 6.2) and find that positivity occurs for both methods with high probability over 95%, approaching 100% with increasing NFE. We refer to further discussion in Rmk. C.6.*

6 Experiments

Based on the theoretical analysis, we expect the θ-Trapezoidal method to outperform the τ-leaping method and the θ-RK-2 method in terms of sample quality, given the same number of function evaluations. This section empirically validates the anticipated effectiveness of our proposed θ-Trapezoidal method (Alg. 2) through comprehensive evaluations across text and image generation tasks. Our comparative analysis includes established discrete diffusion samplers as baselines, *e.g.*, the Euler method [33], τ-leaping [22], Tweedie τ-leaping [30], First-Hitting Sampler (FHS) [21], Parallel Decoding [63], and Semi-Autoregressive (Semi-AR) sampler [64]. We benchmark on both uniform and masked discrete diffusion models, with experiment details provided in App. D.

6.1 15-State Toy Model

We first evaluate the performance of the θ-Trapezoidal method using a 15-state toy model ($d = 1$, $S = 15$). The target distribution is uniformly generated from Δ^{15}, with rate matrix $\boldsymbol{Q} = \frac{1}{15}\boldsymbol{E} - \boldsymbol{I}$, where $\boldsymbol{E}$ is the all-one and $\boldsymbol{I}$ is the identity matrix. This setup provides analytically available score functions, allowing isolation and quantification of numerical errors introduced by inference algorithms. We apply both the θ-Trapezoidal and the θ-RK-2 method to generate 10^6 samples and estimate the KL divergence between the true ground truth $\boldsymbol{p}_0$ and the generated distribution $\widehat{\boldsymbol{q}}_T$.

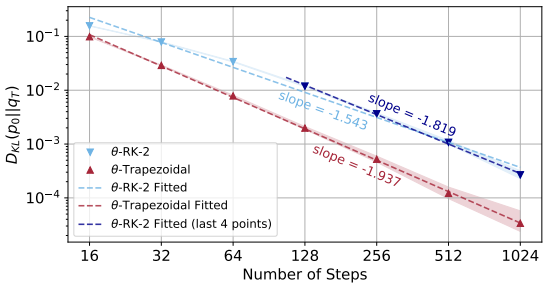

Figure 2: Empirical KL divergence between the true and generated distribution of the toy model vs. number of steps. Data are fitted with linear regression with 95% confidence interval by bootstrapping.

For a fair comparison, we choose $\theta = \frac{1}{2}$ for both methods, and the results are presented in Fig. 2. While both methods exhibit super-linear convergence as the total number of steps grows, the θ-Trapezoidal method outperforms the θ-RK-2 method in terms of both absolute value and convergence rate, while the θ-RK-2 method takes longer to enter the asymptotic regime. Moreover, the fitted line indicates that the θ-Trapezoidal method approximately converges quadratically with respect to the step count, confirming our theoretical results.

6.2 Text Generation

For the text generation task, we employ the pre-trained score function from RADD [33] as our base model for benchmarking inference algorithms. RADD is a masked discrete diffusion model with GPT-2-level text generation capabilities [65] and is trained on the OpenWebText dataset [66] with $d = 1024$ and $S = 50258$. Our comparative analysis maintains consistent computational resources across methods, quantified through the number of score function evaluations (NFE), and evaluates the sample quality produced by FHS, the Euler method, τ-leaping, Tweedie τ-leaping, Semi-AR,

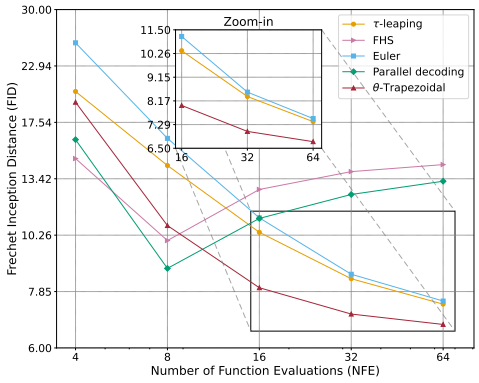

Figure 3: FID of images generated by different sampling algorithms vs. number of function evaluations (NFE). Lower values are better.

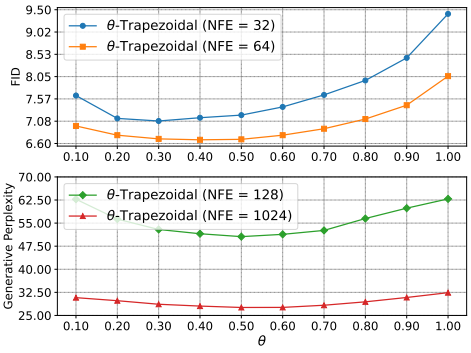

Figure 4: Sampling quality vs. $\theta \in (0, 1]$ in θ-Trapezoid method. **Upper**: Image generation (FID). **Lower**: Text generation (perplexity). Lower is better.

and our proposed θ-Trapezoidal method. We generate text sequences of 1024 tokens and measure their generative perplexity (computed with GPT-2 Large [65]) following the evaluation protocol established in [33].

Tab. 1 presents the results for both low (128) and high (1024) NFEs, with comprehensive results across additional NFE values in Tab. 3. The empirical results demonstrate that the θ-Trapezoidal method consistently produces better samples within a fixed computation budget than existing popular inference algorithms. Notably, it outperforms Euler and Tweedie τ-leaping, two of the best-performing samplers adopted by RADD, by a large margin. It also consistently prevails over FHS, which performs exact simulation at high NFE (1024), supporting again our observations that being free of discretization error does not necessarily imply better sampling quality. These results validate the practical efficiency and accuracy of Alg. 2. Additional evaluation results, including unigram entropy and generative perplexities evaluated under LLaMA 3 [67], are detailed in App. D.2.

Table 1: Generative perplexity (on GPT-2 large) of texts generated by different sampling algorithms. Lower values are better, with the best in **bold**.

Method	NFE $= 128$	NFE $= 1024$
FHS	≤ 122.732	≤ 109.406
Euler	≤ 86.276	≤ 44.686
Tweedie τ-leap.	≤ 85.738	≤ 44.257
τ-leaping	≤ 52.366	≤ 28.797
Semi-AR	≤ 360.793	≤ 147.406
θ-RK-2	≤ 64.317	≤ 36.330
θ-Trapezoidal	$\mathbf{\leq 49.051}$	$\mathbf{\leq 27.553}$

6.3 Image Generation

Our experiments on image generation utilize the pre-trained score function from MaskGIT [63, 68] as the base model, which can be converted into a masked discrete diffusion model by introducing a noise schedule (see App. D.3). MaskGIT employs a masked image transformer architecture trained on ImageNet [69] of 256×256 resolution, where each image amounts to a sequence of 256 discrete image tokens following VQ-GAN tokenization [70] ($d = 256$, $S = 1025$). We evaluate the θ-Trapezoidal method against FHS, the Euler method, τ-leaping, and parallel decoding under equivalent NFE budgets ranging from 4 to 64. Following the setting in [63], we generate 5×10^4 images and compute their Fréchet Inception Distance (FID) against the ImageNet validation split.

Fig. 3 reveals that θ-Trapezoidal method (Alg. 2) consistently achieves lower (and thus better) FID values compared to both the Euler method and τ-leaping across all NFE values. While FHS and parallel decoding show advantages at extremely low NFE (≤ 8), their performance saturates with increased computational resources, making them less favorable compared to our rapidly converging method. Additional results, including generated image samples (Fig. 7), are detailed in App. D.

Remark 6.1 (Algorithm Hyperparameters)**.** *We evaluate the performance of the θ-Trapezoidal method across various θ and NFE values for both text and image generation tasks. As illustrated in Fig. 4, we observe that the θ-Trapezoidal method demonstrates robustness to θ, with a flat landscape near the optimal choice. Our empirical analysis suggests that $\theta \in [0.3, 0.5]$ consistently yields competitive performance across different tasks.*

6.4 Diffusion Large Language Model and Math Reasoning

To verify the effectiveness of the proposed method on a scale, we additionally benchmark its performance on LLaDA-Instruct [64], an 8B masked diffusion LLM with one of the best language modeling performances among discrete diffusion-based LLMs. We examine its performance on GSM8K [71], a math-reasoning dataset consisting of grade-school-level problems. We compare θ-Trapezoidal to the semi-autoregressive (Semi-AR) sampler therein, with both confidence-based (Conf.) and purely random (Rand.) remasking strategies. For each method, we generate a response of 256 tokens in a zero-shot prompting manner, with NFE ranging from 64 to 256, and report the accuracy in Tab. 2. θ-Trapezoidal outperforms the Semi-AR sampler in the low NFE regime, where NFE is strictly smaller than the sequence length. At high NFE regime, θ-Trapezoidal exhibits a similarly competitive performance as other solvers. This observation accords with our claim that high-order samplers perform better with lower NFE budgets, and that these advantages persist even when the model size is large. Further implementation details are available at App. D.4.

Table 2: Response accuracy on GSM8K with different NFEs. The best results are in **bold**.

Accuracy (%)	NFE $= 64$	NFE $= 128$	NFE $= 256$
Semi-AR (Conf.)	33.6	32.0	39.1
Semi-AR (Rand.)	33.8	34.3	**40.3**
θ-Trapezoidal	**35.1**	**38.4**	39.7

7 Conclusion and Future Works

In this work, we introduce the θ-RK-2 and θ-Trapezoidal methods as pioneering high-order numerical schemes tailored for discrete diffusion model inference. Through rigorous analysis based on their stochastic integral formulations, we establish second-order convergence of the θ-Trapezoidal method and that of the θ-RK-2 method under specified conditions. Our analysis indicates that the θ-Trapezoidal method generally provides superior robustness and computational efficiency compared to the θ-RK-2 method. Our empirical evaluations, spanning both a 15-dimensional model with precise score functions and large-scale text and image generation tasks, validate our theoretical findings and demonstrate the superiority performance of our proposed θ-Trapezoidal method over existing samplers in terms of sample quality under equivalent computational constraints. Additionally, we provide a comprehensive analysis of the method's robustness by examining the optimal choice of the parameter θ in our schemes.

Future research directions include comparative analysis of these schemes and development of more sophisticated numerical approaches for discrete diffusion model inference, potentially developing inference methods of higher order [56] or incorporating adaptive step sizes and parallel sampling methodologies. From the perspective of applications, these methods may also show promise for tasks in computational chemistry and biology, particularly in the design of molecules, proteins, and DNA sequences. Moreover, it would also be interesting to explore the usage of higher-order numerical solvers in other problem settings involving the inference of diffusion models, such as sampling via discrete diffusion models [72–75], inference-time scaling of diffusion models [76–91], improving the reasoning ability of diffusion large language models [92, 64, 93–99], etc.

Acknowledgments and Disclosure of Funding

YC is supported by the National Science Foundation under Grants No. ECCS-1942523, DMS-2206576, and CMMI-2450378. GMR is supported by a Google Research Scholar Award. YZ and MT are grateful for partial supports by NSF Grants DMS-1847802, DMS-2513699, DOE Grants NA0004261, SC0026274, and Richard Duke Fellowship. YR, HC and LY acknowledge support of the National Science Foundation under Award No. DMS-2208163.

References

[1] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

[2] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

[3] Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

[4] Griffin Floto, Thorsteinn Jonsson, Mihai Nica, Scott Sanner, and Eric Zhengyu Zhu. Diffusion on the probability simplex. *arXiv preprint arXiv:2309.02530*, 2023.

[5] Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=Lm8T39vLDTE`.

[6] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

[7] Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. *Advances in Neural Information Processing Systems*, 35:34532–34545, 2022.

[8] Pierre H Richemond, Sander Dieleman, and Arnaud Doucet. Categorical sdes with simplex diffusion. *arXiv preprint arXiv:2210.14784*, 2022.

[9] Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=BYWWwSY2G5s`.

[10] Javier E Santos, Zachary R Fox, Nicholas Lubbers, and Yen Ting Lin. Blackout diffusion: generative diffusion models in discrete-state spaces. In *International Conference on Machine Learning*, pages 9034–9059. PMLR, 2023.

[11] Thomas J Kerby and Kevin R Moon. Training-free guidance for discrete diffusion models for molecular generation. *arXiv preprint arXiv:2409.07359*, 2024.

[12] Nathan C Frey, Daniel Berenberg, Karina Zadorozhny, Joseph Kleinhenz, Julien Lafrance-Vanasse, Isidro Hotzel, Yan Wu, Stephen Ra, Richard Bonneau, Kyunghyun Cho, et al. Protein discovery with discrete walk-jump sampling. *arXiv preprint arXiv:2306.12360*, 2023.

[13] Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In *International Conference on Machine Learning*, pages 1276–1301. PMLR, 2023.

[14] Wei Guo, Yuchen Zhu, Molei Tao, and Yongxin Chen. Plug-and-play controllable generation for discrete masked models. *arXiv preprint arXiv:2410.02143*, 2024.

[15] Do Huu Dat, Do Duc Anh, Anh Tuan Luu, and Wray Buntine. Discrete diffusion language model for long text summarization. *arXiv preprint arXiv:2407.10998*, 2024.

[16] Minghui Hu, Yujie Wang, Tat-Jen Cham, Jianfei Yang, and Ponnuthurai N Suganthan. Global context with discrete diffusion in vector quantised modelling for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11502–11511, 2022.

[17] Luke Causer, Grant M Rotskoff, and Juan P Garrahan. Discrete generative diffusion models without stochastic differential equations: a tensor network approach. *arXiv preprint arXiv:2407.11133*, 2024.

[18] Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024.

[19] Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

[20] Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dallatorre, Bernardo P de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*, 2024.

[21] Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.

[22] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

[23] John Charles Butcher. *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*. Wiley-Interscience, 1987.

[24] Kevin Burrage and Pamela Marion Burrage. High strong order explicit runge-kutta methods for stochastic ordinary differential equations. *Applied Numerical Mathematics*, 22(1-3):81–101, 1996.

[25] David F Anderson and Jonathan C Mattingly. A weak trapezoidal method for a class of stochastic differential equations. *Communications in Mathematical Sciences*, 9(1):301–318, 2011.

[26] Yucheng Hu, Tiejun Li, and Bin Min. A weak second order tau-leaping method for chemical kinetic systems. *The Journal of chemical physics*, 135(2), 2011.

[27] Hideyuki Tachibana, Mocho Go, Muneyoshi Inahara, Yotaro Katayama, and Yotaro Watanabe. Quasi-taylor samplers for diffusion generative models based on ideal derivatives. *arXiv preprint arXiv:2112.13339*, 2021.

[28] 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, 2022.

[29] 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*, 2022.

[30] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024.

[31] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

[32] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=GTDKo3Sv9p`.

[33] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

[34] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. Simplified and generalized masked diffusion for discrete data. *arXiv preprint arXiv:2406.04329*, 2024.

[35] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.

[36] Hongrui Chen and Lexing Ying. Convergence analysis of discrete diffusion model: Exact implementation through uniformization. *arXiv preprint arXiv:2402.08095*, 2024.

[37] Zikun Zhang, Zixiang Chen, and Quanquan Gu. Convergence of score-based discrete diffusion models: A discrete-time analysis. *arXiv preprint arXiv:2410.02321*, 2024.

[38] Yinuo Ren, Haoxuan Chen, Grant M Rotskoff, and Lexing Ying. How discrete and continuous diffusion meet: Comprehensive analysis of discrete diffusion models via a stochastic integral framework. *arXiv preprint arXiv:2410.03601*, 2024.

[39] Peter Eris Kloeden and Eckhard Platen. *Numerical solution of stochastic differential equations*. Stochastic Modelling and Applied Probability, Applications of Mathematics, Springer, 1992.

[40] Peter Eris Kloeden, Eckhard Platen, and Henri Schurz. *Numerical solution of SDE through computer experiments*. Springer Science & Business Media, 2012.

[41] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=Loek7hfb46P`.

[42] Qinsheng Zhang, Molei Tao, and Yongxin Chen. gddim: Generalized denoising diffusion implicit models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=1hKE9qjvz-`.

[43] Martin Gonzalez, Nelson Fernandez Pinto, Thuy Tran, Hatem Hajri, Nader Masmoudi, et al. Seeds: Exponential sde solvers for fast high-quality sampling from diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[44] Shuchen Xue, Mingyang Yi, Weijian Luo, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhi-Ming Ma. Sa-solver: Stochastic adams solver for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[45] Qinsheng Zhang, Jiaming Song, and Yongxin Chen. Improved order analysis and design of exponential integrator for diffusion models sampling. *arXiv preprint arXiv:2308.02157*, 2023.

[46] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. *Advances in Neural Information Processing Systems*, 35:30150–30166, 2022.

[47] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

[48] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577, 2022.

[49] Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. *Advances in Neural Information Processing Systems*, 36:55502–55542, 2023.

[50] Gen Li, Yu Huang, Timofey Efimov, Yuting Wei, Yuejie Chi, and Yuxin Chen. Accelerating convergence of score-based diffusion models, provably. *arXiv preprint arXiv:2403.03852*, 2024.

[51] Yuchen Wu, Yuxin Chen, and Yuting Wei. Stochastic Runge-Kutta methods: Provable acceleration of diffusion models. *arXiv preprint arXiv:2410.04760*, 2024.

[52] Daniel T Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of chemical physics*, 115(4):1716–1733, 2001.

[53] Yang Cao, Linda R Petzold, Muruhan Rathinam, and Daniel T Gillespie. The numerical stability of leaping methods for stochastic simulation of chemically reacting systems. *The Journal of chemical physics*, 121(24):12169–12178, 2004.

[54] K Burrage and T Tian. Poisson runge-kutta methods for chemical reaction systems in advances in scientific computing and applications, 2004.

[55] Yucheng Hu and Tiejun Li. Highly accurate tau-leaping methods with random corrections. *The Journal of chemical physics*, 130(12), 2009.

[56] Markus Arns, Peter Buchholz, and Andriy Panchenko. On the numerical analysis of inhomogeneous continuous-time markov chains. *INFORMS Journal on Computing*, 22(3):416–432, 2010.

[57] Desmond J Higham. Modeling and simulating chemical reactions. *SIAM review*, 50(2):347–368, 2008.

[58] Weinan E, Tiejun Li, and Eric Vanden-Eijnden. *Applied stochastic analysis*, volume 199. American Mathematical Soc., 2021.

[59] Frank P Kelly. *Reversibility and stochastic networks*. Cambridge University Press, 2011.

[60] Joe Benton, Yuyang Shi, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. From denoising diffusions to denoising markov models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(2):286–301, 2024.

[61] Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=r5njV3BsuD`.

[62] Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ode is provably fast. *Advances in Neural Information Processing Systems*, 36, 2024.

[63] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[64] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

[65] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[66] Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

[67] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[68] Victor Besnier and Mickael Chen. A pytorch reproduction of masked generative image transformer. *arXiv preprint arXiv:2310.14400*, 2023.

[69] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[70] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[71] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[72] Peter Holderrieth, Michael S Albergo, and Tommi Jaakkola. Leaps: A discrete neural sampler via locally equivariant networks. *arXiv preprint arXiv:2502.10843*, 2025.

[73] Yuchen Zhu, Wei Guo, Jaemoo Choi, Guan-Horng Liu, Yongxin Chen, and Molei Tao. Mdns: Masked diffusion neural sampler via stochastic optimal control. *arXiv preprint arXiv:2508.10684*, 2025.

[74] Zijing Ou, Ruixiang Zhang, and Yingzhen Li. Discrete neural flow samplers with locally equivariant transformer. *arXiv preprint arXiv:2505.17741*, 2025.

[75] Wei Guo, Jaemoo Choi, Yuchen Zhu, Molei Tao, and Yongxin Chen. Proximal diffusion neural sampler. *arXiv preprint arXiv:2510.03824*, 2025.

[76] Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

[77] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Scaling inference time compute for diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2523–2534, 2025.

[78] Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

[79] Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. *arXiv preprint arXiv:2503.02819*, 2025.

[80] Haoxuan Chen, Yinuo Ren, Martin Renqiang Min, Lexing Ying, and Zachary Izzo. Solving inverse problems via diffusion-based priors: An approximation-free ensemble sampling approach. *arXiv preprint arXiv:2506.03979*, 2025.

[81] Yinuo Ren, Wenhao Gao, Lexing Ying, Grant M Rotskoff, and Jiequn Han. Driftlite: Lightweight drift control for inference-time scaling of diffusion models. *arXiv preprint arXiv:2509.21655*, 2025.

[82] Sophia Tang, Yuchen Zhu, Molei Tao, and Pranam Chatterjee. Tr2-d2: Tree search guided trajectory-aware fine-tuning for discrete diffusion. *arXiv preprint arXiv:2509.25171*, 2025.

[83] Meihua Dang, Jiaqi Han, Minkai Xu, Kai Xu, Akash Srivastava, and Stefano Ermon. Inference-time scaling of diffusion language models with particle gibbs sampling. *arXiv preprint arXiv:2507.08390*, 2025.

[84] Vignav Ramesh and Morteza Mardani. Test-time scaling of diffusion models via noise trajectory search. *arXiv preprint arXiv:2506.03164*, 2025.

[85] Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models. *arXiv preprint arXiv:2506.20701*, 2025.

[86] Tianlang Chen, Minkai Xu, Jure Leskovec, and Stefano Ermon. Rfg: Test-time scaling for diffusion large language model reasoning with reward-free guidance. *arXiv preprint arXiv:2509.25604*, 2025.

[87] Xiangcheng Zhang, Haowei Lin, Haotian Ye, James Zou, Jianzhu Ma, Yitao Liang, and Yilun Du. Inference-time scaling of diffusion models through classical search. *arXiv preprint arXiv:2505.23614*, 2025.

[88] Mohsin Hasan, Marta Skreta, Alan Aspuru-Guzik, Yoshua Bengio, and Kirill Neklyudov. Discrete feynman-kac correctors. In *2nd AI for Math Workshop@ ICML 2025*, 2025.

[89] Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.

[90] Cheuk Kit Lee, Paul Jeha, Jes Frellsen, Pietro Lio, Michael Samuel Albergo, and Francisco Vargas. Debiasing guidance for discrete diffusion with sequential monte carlo. *arXiv preprint arXiv:2502.06079*, 2025.

[91] Zijing Ou, Chinmay Pani, and Yingzhen Li. Inference-time scaling of discrete diffusion models via importance weighting and optimal proposal design. *arXiv e-prints*, pages arXiv–2505, 2025.

[92] Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, et al. Diffusion of thought: Chain-of-thought reasoning in diffusion language models. *Advances in Neural Information Processing Systems*, 37:105345–105374, 2024.

[93] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

[94] Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025.

[95] Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*, 2025.

[96] Chengyu Wang, Paria Rashidinejad, DiJia Su, Song Jiang, Sid Wang, Siyan Zhao, Cai Zhou, Shannon Zejiang Shen, Feiyu Chen, Tommi Jaakkola, et al. Spg: Sandwiched policy gradient for masked diffusion language models. *arXiv preprint arXiv:2510.09541*, 2025.

[97] Siyan Zhao, Mengchen Liu, Jing Huang, Miao Liu, Chenyu Wang, Bo Liu, Yuandong Tian, Guan Pang, Sean Bell, Aditya Grover, et al. Inpainting-guided policy optimization for diffusion large language models. *arXiv preprint arXiv:2509.10396*, 2025.

[98] Cai Zhou, Chenxiao Yang, Yi Hu, Chenyu Wang, Chubin Zhang, Muhan Zhang, Lester Mackey, Tommi Jaakkola, Stephen Bates, and Dinghuai Zhang. Coevolutionary continuous discrete diffusion: Make your diffusion language model a latent reasoner. *arXiv preprint arXiv:2510.03206*, 2025.

[99] Yuchen Zhu, Wei Guo, Jaemoo Choi, Petr Molodyk, Bo Yuan, Molei Tao, and Yongxin Chen. Enhancing reasoning for diffusion llms via distribution matching policy optimization. *arXiv preprint arXiv:2510.08233*, 2025.

[100] Ilia Igashov, Arne Schneuing, Marwin Segler, Michael Bronstein, and Bruno Correia. Retrobridge: Modeling retrosynthesis with markov bridges. *arXiv preprint arXiv:2308.16212*, 2023.

[101] Yang Li, Jinpei Guo, Runzhong Wang, and Junchi Yan. From distribution learning in training to gradient search in testing for combinatorial optimization. *Advances in Neural Information Processing Systems*, 36, 2024.

[102] Zhiqing Sun and Yiming Yang. Difusco: Graph-based diffusion solvers for combinatorial optimization. *Advances in Neural Information Processing Systems*, 36:3706–3731, 2023.

[103] Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Bac Nguyen, Stefano Ermon, and Yuki Mitsufuji. G2d2: Gradient-guided discrete diffusion for image inverse problem solving. *arXiv preprint arXiv:2410.14710*, 2024.

[104] Wenda Chu, Yang Song, and Yisong Yue. Split gibbs discrete diffusion posterior sampling. *arXiv preprint arXiv:2503.01161*, 2025.

[105] Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex X Lu, Nicolo Fusi, Ava P Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *BioRxiv*, pages 2023–09, 2023.

[106] Patrick Emami, Aidan Perreault, Jeffrey Law, David Biagioni, and Peter St John. Plug & play directed evolution of proteins with gradient-based discrete mcmc. *Machine Learning: Science and Technology*, 4(2):025014, 2023.

[107] Dmitry Penzar, Daria Nogina, Elizaveta Noskova, Arsenii Zinkevich, Georgy Meshcheryakov, Andrey Lando, Abdul Muntakim Rafi, Carl De Boer, and Ivan V Kulakovskiy. Legnet: a best-in-class deep learning model for short dna regulatory regions. *Bioinformatics*, 39(8):btad457, 2023.

[108] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[109] John J Yang, Jason Yim, Regina Barzilay, and Tommi Jaakkola. Fast non-autoregressive inverse folding with discrete diffusion. *arXiv preprint arXiv:2312.02447*, 2023.

[110] Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design. *arXiv preprint arXiv:2402.05841*, 2024.

[111] Kai Yi, Bingxin Zhou, Yiqing Shen, Pietro Liò, and Yuguang Wang. Graph denoising diffusion for inverse protein folding. *Advances in Neural Information Processing Systems*, 36, 2024.

[112] Yiheng Zhu, Jialu Wu, Qiuyi Li, Jiahuan Yan, Mingze Yin, Wei Wu, Mingyang Li, Jieping Ye, Zheng Wang, and Jian Wu. Bridge-if: Learning inverse protein folding with markov bridges. *arXiv preprint arXiv:2411.02120*, 2024.

[113] Patrick Esser, Robin Rombach, Andreas Blattmann, and Bjorn Ommer. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis. *Advances in neural information processing systems*, 34:3518–3532, 2021.

[114] Jose Lezama, Tim Salimans, Lu Jiang, Huiwen Chang, Jonathan Ho, and Irfan Essa. Discrete predictor-corrector diffusion models for image synthesis. In *The Eleventh International Conference on Learning Representations*, 2022.

[115] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022.

[116] Ari Seff, Wenda Zhou, Farhan Damani, Abigail Doyle, and Ryan P Adams. Discrete object generation with reversible inductive construction. *Advances in neural information processing systems*, 32, 2019.

[117] Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*, pages 4474–4484. PMLR, 2020.

[118] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020.

[119] Yiming Qin, Clement Vignac, and Pascal Frossard. Sparse training of discrete diffusion models for graph generation. *arXiv preprint arXiv:2311.02142*, 2023.

[120] Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

[121] Kilian Konstantin Haefeli, Karolis Martinkus, Nathanaël Perraudin, and Roger Wattenhofer. Diffusion models for graphs benefit from discrete state spaces. *arXiv preprint arXiv:2210.01549*, 2022.

[122] Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. Defog: Discrete flow matching for graph generation. *arXiv preprint arXiv:2410.04263*, 2024.

[123] Jun Hyeong Kim, Seonghwan Kim, Seokhyun Moon, Hyeongwoo Kim, Jeheon Woo, and Woo Youn Kim. Discrete diffusion schrödinger bridge matching for graph transformation. *arXiv preprint arXiv:2410.01500*, 2024.

[124] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Layoutdm: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10167–10176, 2023.

[125] Junyi Zhang, Jiaqi Guo, Shizhao Sun, Jian-Guang Lou, and Dongmei Zhang. Layoutdiffusion: Improving graphic layout generation by discrete diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7226–7236, 2023.

[126] Seunggeun Chi, Hyung-gun Chi, Hengbo Ma, Nakul Agarwal, Faizan Siddiqui, Karthik Ramani, and Kwonjoon Lee. M2d2m: Multi-motion generation from text with discrete diffusion models. *arXiv preprint arXiv:2407.14502*, 2024.

[127] Yunhong Lou, Linchao Zhu, Yaxiong Wang, Xiaohan Wang, and Yi Yang. Diversemotion: Towards diverse human motion generation via discrete diffusion. *arXiv preprint arXiv:2309.01372*, 2023.

[128] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1720–1733, 2023.

[129] Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P Breckon, and Chris G Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. In *European Conference on Computer Vision*, pages 170–188. Springer, 2022.

[130] Zhicong Tang, Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Improved vector quantized diffusion models. *arXiv preprint arXiv:2205.16007*, 2022.

[131] Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. Discrete contrastive diffusion for cross-modal music and image generation. *arXiv preprint arXiv:2206.07771*, 2022.

[132] Zhichao Wu, Qiulin Li, Sixing Liu, and Qun Yang. Dctts: Discrete diffusion model with contrastive learning for text-to-speech generation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11336–11340. IEEE, 2024.

[133] Jun Han, Zixiang Chen, Yongqian Li, Yiwen Kou, Eran Halperin, Robert E Tillman, and Quanquan Gu. Guided discrete diffusion for electronic health record generation. *arXiv preprint arXiv:2404.12314*, 2024.

[134] Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. Tabdiff: a multi-modal diffusion model for tabular data generation. *arXiv preprint arXiv:2410.20626*, 2024.

[135] Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. *arXiv preprint arXiv:2211.15029*, 2022.

[136] Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation. *arXiv preprint arXiv:2112.06749*, 2021.

[137] Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. Ar-diffusion: Auto-regressive diffusion model for text generation. *Advances in Neural Information Processing Systems*, 36:39957–39974, 2023.

[138] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq-v2: Bridging discrete and continuous text spaces for accelerated seq2seq diffusion models. *arXiv preprint arXiv:2310.05793*, 2023.

[139] Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023.

[140] Kun Zhou, Yifan Li, Wayne Xin Zhao, and Ji-Rong Wen. Diffusion-nat: Self-prompting discrete diffusion for non-autoregressive text generation. *arXiv preprint arXiv:2305.04044*, 2023.

[141] Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. *arXiv preprint arXiv:2410.21357*, 2024.

[142] Yinuo Ren, Grant M Rotskoff, and Lexing Ying. A unified approach to analysis and design of denoising markov models. *arXiv preprint arXiv:2504.01938*, 2025.

[143] Yong-Hyun Park, Chieh-Hsin Lai, Satoshi Hayakawa, Yuhta Takida, and Yuki Mitsufuji. Jump your steps: Optimizing sampling schedule of discrete diffusion models. *arXiv preprint arXiv:2410.07761*, 2024.

[144] Yixiu Zhao, Jiaxin Shi, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models. *arXiv preprint arXiv:2407.21243*, 2024.

[145] Zixiang Chen, Huizhuo Yuan, Yongqian Li, Yiwen Kou, Junkai Zhang, and Quanquan Gu. Fast sampling via discrete non-markov diffusion models with predetermined transition time. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[146] Lingxiao Zhao, Xueying Ding, Lijun Yu, and Leman Akoglu. Improving and unifying discrete&continuous-time discrete denoising diffusion. *arXiv preprint arXiv:2402.03701*, 2024.

[147] Machel Reid, Vincent Josua Hellendoorn, and Graham Neubig. Diffuser: Diffusion via edit-based reconstruction. In *The Eleventh International Conference on Learning Representations*, 2023.

[148] Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Distillation of discrete diffusion through dimensional correlations. *arXiv preprint arXiv:2410.08709*, 2024.

[149] Ludwig Winkler, Lorenz Richter, and Manfred Opper. Bridging discrete and continuous state spaces: Exploring the ehrenfest process in time-continuous diffusion models. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=8GYclcxQXB`.

[150] Harshit Varma, Dheeraj Nagaraj, and Karthikeyan Shanmugam. Glauber generative model: Discrete diffusion models via binary classification. *arXiv preprint arXiv:2405.17035*, 2024.

[151] Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36, 2024.

[152] Severi Rissanen, Markus Heinonen, and Arno Solin. Improving discrete diffusion models via structured preferential generation. *arXiv preprint arXiv:2405.17889*, 2024.

[153] Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gómez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising. *arXiv preprint arXiv:2410.06264*, 2024.

[154] Oscar Davis, Samuel Kessler, Mircea Petrache, Avishek Joey Bose, et al. Fisher flow matching for generative modeling over discrete data. *arXiv preprint arXiv:2405.14664*, 2024.

[155] GN Mil'shtejn. Approximate integration of stochastic differential equations. *Theory of Probability & Its Applications*, 19(3):557–562, 1975.

[156] Assyr Abdulle and Stephane Cirilli. S-rock: Chebyshev methods for stiff stochastic differential equations. *SIAM Journal on Scientific Computing*, 30(2):997–1014, 2008.

[157] Evelyn Buckwar and Renate Winkler. Multistep methods for sdes and their application to problems with small noise. *SIAM journal on numerical analysis*, 44(2):779–803, 2006.

[158] Denis Talay and Luciano Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic analysis and applications*, 8(4):483–509, 1990.

[159] Kevin Burrage and Pamela M Burrage. Order conditions of stochastic runge–kutta methods by b-series. *SIAM Journal on Numerical Analysis*, 38(5):1626–1646, 2000.

[160] Kevin Burrage and Tianhai Tian. Predictor-corrector methods of runge–kutta type for stochastic differential equations. *SIAM Journal on Numerical Analysis*, 40(4):1516–1537, 2002.

[161] Andreas Rössler. Runge-kutta methods for the numerical solution of stochastic differential equations. *Shaker-Verlag, Aachen*, 2003.

[162] Andreas Rößler. Runge–kutta methods for the strong approximation of solutions of stochastic differential equations. *SIAM Journal on Numerical Analysis*, 48(3):922–952, 2010.

[163] James M Foster, Goncalo Dos Reis, and Calum Strange. High order splitting methods for sdes satisfying a commutativity condition. *SIAM Journal on Numerical Analysis*, 62(1):500–532, 2024.

[164] Lei Li, Jianfeng Lu, Jonathan Mattingly, and Lihan Wang. Numerical methods for stochastic differential equations based on gaussian mixtures. *Communications in Mathematical Sciences*, 19(6):1549–1577, 2021.

[165] Ruoqi Shen and Yin Tat Lee. The randomized midpoint method for log-concave sampling. *Advances in Neural Information Processing Systems*, 32, 2019.

[166] Nima Anari, Sinho Chewi, and Thuy-Duong Vuong. Fast parallel sampling under isoperimetry. *arXiv preprint arXiv:2401.09016*, 2024.

[167] Lu Yu and Arnak Dalalyana. Parallelized midpoint randomization for langevin monte carlo. *arXiv preprint arXiv:2402.14434*, 2024.

[168] Changyou Chen, Nan Ding, and Lawrence Carin. On the convergence of stochastic gradient mcmc algorithms with high-order integrators. *Advances in neural information processing systems*, 28, 2015.

[169] Alain Durmus, Umut Simsekli, Eric Moulines, Roland Badeau, and Gaël Richard. Stochastic gradient richardson-romberg markov chain monte carlo. *Advances in neural information processing systems*, 29, 2016.

[170] Xuechen Li, Yi Wu, Lester Mackey, and Murat A Erdogdu. Stochastic runge-kutta accelerates langevin monte carlo and beyond. *Advances in neural information processing systems*, 32, 2019.

[171] Sotirios Sabanis and Ying Zhang. Higher order langevin monte carlo algorithm. *Electron. J. Statist*, 13(2):3805–3850, 2019.

[172] Wenlong Mou, Yi-An Ma, Martin J Wainwright, Peter L Bartlett, and Michael I Jordan. High-order langevin diffusion yields an accelerated mcmc algorithm. *Journal of Machine Learning Research*, 22(42):1–41, 2021.

[173] Pierre Monmarché. High-dimensional mcmc with a standard splitting scheme for the underdamped langevin diffusion. *Electronic Journal of Statistics*, 15(2):4117–4166, 2021.

[174] James Foster, Terry Lyons, and Harald Oberhauser. The shifted ode method for underdamped langevin mcmc. *arXiv preprint arXiv:2101.03446*, 2021.

[175] Kevin Burrage, PM Burrage, and Tianhai Tian. Numerical methods for strong solutions of stochastic differential equations: an overview. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 460(2041):373–402, 2004.

[176] Grigori N Milstein and Michael V Tretyakov. *Stochastic numerics for mathematical physics*, volume 39. Springer, 2004.

[177] Alfred B Bortz, Malvin H Kalos, and Joel L Lebowitz. A new algorithm for monte carlo simulation of ising spin systems. *Journal of Computational physics*, 17(1):10–18, 1975.

[178] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434, 1976.

[179] Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361, 1977.

[180] Yang Cao, Dan Gillespie, and Linda Petzold. Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. *Journal of Computational Physics*, 206(2):395–411, 2005.

[181] Yang Cao, Daniel T Gillespie, and Linda R Petzold. The slow-scale stochastic simulation algorithm. *The Journal of chemical physics*, 122(1), 2005.

[182] Weinan E, Di Liu, Eric Vanden-Eijnden, et al. Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates. *The Journal of chemical physics*, 123(19), 2005.

[183] Weinan E, Di Liu, and Eric Vanden-Eijnden. Nested stochastic simulation algorithms for chemical kinetic systems with multiple time scales. *Journal of computational physics*, 221 (1):158–180, 2007.

[184] Michael A Gibson and Jehoshua Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The journal of physical chemistry A*, 104(9): 1876–1889, 2000.

[185] David F Anderson. A modified next reaction method for simulating chemical systems with time dependent propensities and delays. *The Journal of chemical physics*, 127(21), 2007.

[186] Casper HL Beentjes and Ruth E Baker. Uniformization techniques for stochastic simulation of chemical reaction networks. *The Journal of Chemical Physics*, 150(15), 2019.

[187] Muruhan Rathinam, Linda R Petzold, Yang Cao, and Daniel T Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics*, 119(24):12784–12794, 2003.

[188] Daniel T Gillespie and Linda R Petzold. Improved leap-size selection for accelerated stochastic simulation. *The journal of chemical physics*, 119(16):8229–8234, 2003.

[189] Kevin Burrage, Tianhai Tian, and Pamela Burrage. A multi-scaled approach for simulating chemical reaction systems. *Progress in biophysics and molecular biology*, 85(2-3):217–234, 2004.

[190] Yang Cao, Daniel T Gillespie, and Linda R Petzold. Avoiding negative populations in explicit poisson tau-leaping. *The Journal of chemical physics*, 123(5), 2005.

[191] Anne Auger, Philippe Chatelain, and Petros Koumoutsakos. R-leaping: Accelerating the stochastic simulation algorithm by reaction leaps. *The Journal of chemical physics*, 125(8), 2006.

[192] Yang Cao, Daniel T Gillespie, and Linda R Petzold. Adaptive explicit-implicit tau-leaping method with automatic tau selection. *The Journal of chemical physics*, 126(22), 2007.

[193] Basil Bayati, Philippe Chatelain, and Petros Koumoutsakos. D-leaping: Accelerating stochastic simulation algorithms for reactions with delays. *Journal of Computational Physics*, 228 (16):5908–5916, 2009.

[194] Yang Cao and Linda Petzold. Slow-scale tau-leaping method. *Computer methods in applied mechanics and engineering*, 197(43-44):3472–3479, 2008.

[195] Zhouyi Xu and Xiaodong Cai. Unbiased τ-leap methods for stochastic simulation of chemically reacting systems. *The Journal of chemical physics*, 128(15), 2008.

[196] Mary Sehl, Alexander V Alekseyenko, and Kenneth L Lange. Accurate stochastic simulation via the step anticipation τ-leaping (sal) algorithm. *Journal of Computational Biology*, 16(9): 1195–1208, 2009.

[197] Krishna A Iyengar, Leonard A Harris, and Paulette Clancy. Accurate implementation of leaping in space: The spatial partitioned-leaping algorithm. *The Journal of chemical physics*, 132(9), 2010.

[198] David F Anderson and Desmond J Higham. Multilevel monte carlo for continuous time markov chains, with applications in biochemical kinetics. *Multiscale Modeling & Simulation*, 10(1):146–179, 2012.

[199] Alvaro Moraes, Raúl Tempone, and Pedro Vilanova. Hybrid chernoff tau-leap. *Multiscale Modeling & Simulation*, 12(2):581–615, 2014.

[200] Jill Padgett and Silvana Ilie. An adaptive tau-leaping method for stochastic simulations of reaction-diffusion systems. *AIP Advances*, 6(3), 2016.

[201] Jana Lipková, Georgios Arampatzis, Philippe Chatelain, Bjoern Menze, and Petros Koumoutsakos. S-leaping: an adaptive, accelerated stochastic simulation algorithm, bridging τ-leaping and r-leaping. *Bulletin of mathematical biology*, 81(8):3074–3096, 2019.

[202] Muruhan Rathinam, Linda R Petzold, Yang Cao, and Daniel T Gillespie. Consistency and stability of tau-leaping schemes for chemical reaction systems. *Multiscale Modeling & Simulation*, 4(3):867–895, 2005.

[203] Tiejun Li. Analysis of explicit tau-leaping schemes for simulating chemically reacting systems. *Multiscale Modeling & Simulation*, 6(2):417–436, 2007.

[204] Yucheng Hu, Tiejun Li, and Bin Min. The weak convergence analysis of tau-leaping methods: revisited. *Communications in Mathematical Sciences*, 9(4):965–996, 2011.

[205] David F Anderson, Desmond J Higham, and Yu Sun. Complexity of multilevel monte carlo tau-leaping. *SIAM Journal on Numerical Analysis*, 52(6):3106–3127, 2014.

[206] Chuchu Chen and Di Liu. Error analysis for d-leaping scheme of chemical reaction system with delay. *Multiscale Modeling & Simulation*, 15(4):1797–1829, 2017.

[207] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[208] Linfeng Zhang, Weinan E, and Lei Wang. Monge-ampère flow for generative modeling. *arXiv preprint arXiv:1809.10188*, 2018.

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

[210] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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

[212] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34: 1415–1428, 2021.

[213] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[214] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[215] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[216] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[217] Haoxuan Chen, Yinuo Ren, Lexing Ying, and Grant M Rotskoff. Accelerating diffusion models with parallel sampling: Inference at sub-linear time complexity. *arXiv preprint arXiv:2405.15986*, 2024.

[218] Zhenyu Zhou, Defang Chen, Can Wang, and Chun Chen. Fast ode-based sampling for diffusion models in around 5 steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7777–7786, 2024.

[219] 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, 2023.

[220] Hanzhong Guo, Cheng Lu, Fan Bao, Tianyu Pang, Shuicheng Yan, Chao Du, and Chongxuan Li. Gaussian mixture solvers for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[221] Jonathan Heek, Emiel Hoogeboom, and Tim Salimans. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024.

[222] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[223] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

[224] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.

[225] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

[226] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023.

[227] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[228] Vinh Tong, Trung-Dung Hoang, Anji Liu, Guy Van den Broeck, and Mathias Niepert. Learning to discretize denoising diffusion odes. *arXiv preprint arXiv:2405.15506*, 2024.

[229] Eric Frankel, Sitan Chen, Jerry Li, Pang Wei Koh, Lillian J Ratliff, and Sewoong Oh. S4s: Solving for a diffusion model solver. *arXiv preprint arXiv:2502.17423*, 2025.

[230] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. *arXiv preprint arXiv:2112.07068*, 2021.

[231] Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *International Conference on Machine Learning*, pages 42390–42402. PMLR, 2023.

[232] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15762–15772, 2024.

[233] Valentin De Bortoli, Alexandre Galashov, Arthur Gretton, and Arnaud Doucet. Accelerated diffusion models via speculative sampling. *arXiv preprint arXiv:2501.05370*, 2025.

[234] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021.

[235] 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, 2024.

[236] Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[237] Hyungjin Chung, Jeongsol Kim, Sehui Kim, and Jong Chul Ye. Parallel diffusion models of operator and image for blind inverse problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6059–6069, 2023.

[238] Zhiwei Tang, Jiasheng Tang, Hao Luo, Fan Wang, and Tsung-Hui Chang. Accelerating parallel sampling of diffusion models. In *Forty-first International Conference on Machine Learning*, 2024.

[239] Jiezhang Cao, Yue Shi, Kai Zhang, Yulun Zhang, Radu Timofte, and Luc Van Gool. Deep equilibrium diffusion restoration with parallel sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2824–2834, 2024.

[240] Nikil Roashan Selvam, Amil Merchant, and Stefano Ermon. Self-refining diffusion samplers: Enabling parallelization via parareal iterations. *arXiv preprint arXiv:2412.08292*, 2024.

[241] Saravanan Kandasamy and Dheeraj Nagaraj. The poisson midpoint method for langevin dynamics: Provably efficient discretization for diffusion models. *arXiv preprint arXiv:2405.17068*, 2024.

[242] Shivam Gupta, Linda Cai, and Sitan Chen. Faster diffusion-based sampling with randomized midpoints: Sequential and parallel. *arXiv preprint arXiv:2406.00924*, 2024.

[243] Philip Protter. Point process differentials with evolving intensities. In *Nonlinear stochastic problems*, pages 467–472. Springer, 1983.

[244] Bernt Øksendal and Agnes Sulem. *Applied Stochastic Control of Jump Diffusions*. Springer, 2019.

A Further Discussion on Related Works

In this section, we provide a more detailed literature review of both continuous and discrete diffusion models, as well as several studies on the numerical methods for SDEs and chemical reaction systems, which are highly related to our work.

Discrete Diffusion Models: Methodology, Theory, and Applications. Discrete diffusion and flow-based models [1–4, 6–10, 22] have recently been proposed as generalizations of continuous diffusion models to model discrete distributions.

Such models have been widely used in various areas of science and engineering, including but not limited to modeling retrosynthesis [100], combinatorial optimization [101, 102], solving inverse problems [103, 104] and sampling high-dimensional discrete distributions [90, 72], designing molecules, proteins, and DNA sequences [105, 13, 106, 12, 107–109, 31, 110, 11, 111, 112], image synthesis [113–115], text summarization [15], as well as the generation of graph [116–123], layout [124, 125], motion [126, 127], sound [22, 128], image [16, 129–131], speech [132], electronic health record [133], tabular data [134] and text [135–140, 34, 35, 141, 14]. Inspired by the huge success achieved by discrete diffusion models in practice, researchers have also conducted some studies on the theoretical properties of these models, such as [36–38, 142].

An extensive amount of work has also explored the possibility of making discrete diffusion models more effective from many aspects, such as optimizing the sampling schedule [143], adding correctors [144], developing fast samplers [145], designing correctors based on information learnt by the model [144], simplifying the loss function for training [146], adding editing-based refinements [147], synergizing these models with other techniques and methodologies like distillation [148], Ehrenfest processes [149], Glauber dynamics [150], tensor networks [17], enhanced guidance mechanisms [151, 18–20], structured preferential generation [152], the plan-and-denoise framework [153] and alternative metrics, *e.g.*, the Fisher information metric [154]. However, to the best of our knowledge, existing work on accelerating the inference of discrete diffusion models is relatively sparse compared to the ones we listed above, which makes it a direction worth exploring and serves as one of the main motivations behind this work.

Numerical Methods for SDEs and Chemical Reaction Systems. Below, we review advanced numerical methods proposed for simulating SDEs and chemical reaction systems, which are the main techniques adopted in our work. For the simulation of SDEs driven by Brownian motions, many studies have been performed to design more accurate numerical schemes, which have been widely applied to tackle problems in computational physics, optimization, and Monte Carlo sampling. Examples of such work include the Milstein method [155], explicit methods [156], multistep methods [157], extrapolation-type methods [158, 25], stochastic Runge Kutta methods [24, 159–162], splitting methods [163], methods based on gaussian mixtures [164], randomized midpoint method [165], parallel sampling methods [166, 167] as well as high-order methods for stochastic gradient Markov Chain Monte Carlo [168, 169], underdamped and overdamped Langevin Monte Carlo [170–174]. For a more comprehensive list of related numerical methods, one may refer to [39, 175, 176, 40, 58].

Regarding the simulation of chemical reaction systems, numerical methods can be categorized into two classes. The first class consists of exact simulation methods, which are similar to the Kinetic Monte Carlo (KMC) method [177] developed for simulating spin dynamics and crystal growth in condensed matter physics. Examples of such methods include the Gillespie algorithm (or the Stochastic Simulation Algorithm, a.k.a. SSA) [178, 179] and its variants for multiscale modeling [180–183], the next reaction method and its variants [184, 185], uniformization-based methods [186], etc. The second class of methods are approximate simulation methods, including but not limited to the τ-leaping method [52] and its variants [187, 188, 53, 54, 189–195, 55, 196, 56, 197, 26, 198–201]. For a subset of the methods listed above, numerical analysis has also been performed in many works [202–206] to justify their validity.

Continuous Diffusion Models: Methodology, Theory, and Acceleration. Continuous diffusion and probability flow-based models [207–216] have also been the most popular methods in generative modeling, with a wide range of applications in science and engineering. For a list of related work on the theoretical studies and applications of these models, one may refer to the literature review

conducted in [217, 38]. Here we will only review studies on accelerating the inference of continuous diffusion models, which motivates our work.

An incomplete list of accelerating methods includes approximate mean direction solver [218], restart sampling [219], gaussian mixture solvers [220], self-consistency [221–224], knowledge distillation [225–229], combination with underdamped Langevin dynamics [230], operator learning [231] and more recently ideas from accelerating large language models (LLMs) like caching [232] and speculative decoding [233]. Among all the proposed accelerating methods, one major class of methods are developed based on techniques from numerical analysis like adaptive step sizes [234], exponential integrators [41–43], predictor-corrector solver [235], Adams-Bashforth methods [29, 44, 45], Taylor methods [27, 46], Picard iteration and parallel sampling [236–240, 217], (stochastic) Runge-Kutta methods [47, 28, 48–51] and randomized midpoint method [241, 242]. In contrast, there have been fewer studies on the acceleration of discrete diffusion models via techniques from numerical analysis, which inspires the study undertaken in this paper.

B Mathematical Background

In this section, we provide the mathematical background for the stochastic integral formulation of discrete diffusion models, the error analysis of the τ-leaping method, and useful lemmas for the theoretical analysis of high-order schemes for discrete diffusion models.

B.1 Stochastic Integral Formulation of Discrete Diffusion Models

Throughout this section, we will assume that $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space, $\mathbb{X}$ is a finite-state space, and denote the pairwise difference set of the state space by $\mathbb{D} := \{x - y : x \neq y \in \mathbb{X}\}$. We also assume that the pairwise difference set $\mathbb{X}$ is equipped with a metric $\|\cdot\|$, a finite measure γ, and a σ-algebra $\mathcal{B}$.

As a warm-up, we introduce the definition of the Poisson random measure for a time-homogeneous counting process.

Definition B.1 (Poisson Random Measure [38, Definition A.1])**.** *The random measure $N(\mathrm{d}t, \mathrm{d}\nu)$ on $\mathbb{R}^+ \times \mathbb{D}$ is called a* Poisson random measure *w.r.t. measure γ if it is a random counting measure satisfying the following properties:*

(i) For any $B \in \mathcal{B}$ and $0 \leq s < t$,

$$N((s,t] \times B) \sim \mathcal{P}\left(\gamma(B)(t-s)\right);$$

(ii) For any $t \geq 0$ and pairwise disjoint sets $\{B_i\}_{i\in[n]} \subset \mathcal{B}$,

$$\{N_t(B_i) := N((0,t] \times B_i)\}_{i\in[n]}$$

are independent stochastic processes.

Then we define the Poisson random measure with evolving intensities. The term "evolving" refers to that the intensity is both time and state-dependent.

Definition B.2 (Poisson Random Measure with Evolving Intensity [38, Definition A.3])**.** *Suppose $\lambda_t(y)$ is a non-negative predictable process on $\mathbb{R}^+ \times \mathbb{D} \times \Omega$ satisfying that for any $0 \leq T < \overline{T}$, $\int_0^T \lambda_t(\nu)\mathrm{d}t < \infty$, a.s..*

The random measure $N[\lambda](\mathrm{d}t, \mathrm{d}\nu)$ on $\mathbb{R}^+ \times \mathbb{D}$ is called a Poisson random measure *with* evolving intensity *$\lambda_t(\nu)$ w.r.t. measure γ if it is a random counting measure satisfying the following properties:*

(i) For any $B \in \mathcal{B}$ and $0 \leq s < t$,

$$N[\lambda]((s,t] \times B) \sim \mathcal{P}\left(\int_s^t \int_B \lambda_\tau(\nu)\gamma(\mathrm{d}\nu)\mathrm{d}\tau\right);$$

(ii) For any $t \geq 0$ and pairwise disjoint sets $\{B_i\}_{i\in[n]} \subset \mathcal{B}$,

$$\{N_t[\lambda](B_i) := N[\lambda]((0,t] \times B_i)\}_{i\in[n]}$$

are independent stochastic processes.

Remark B.3 (Construction of Poisson Random Measure with Evolving Intensity)**.** *As discussed in Thm. A.4 in [38] and originally proposed by [243], the Poisson random measure with evolving intensity can be constructed in the following way.*

One first augments the $(\mathbb{X}, \mathcal{B}, \nu)$ *measure space to a product space* $(\mathbb{D} \times \mathbb{R}, \mathcal{B} \times \mathcal{B}(\mathbb{R}), \gamma \times m)$*, where* m *is the Lebesgue measure on* $\mathbb{R}$*, and* $\mathcal{B}(\mathbb{R})$ *is the Borel* σ*-algebra on* $\mathbb{R}$*. The Poisson random measure with evolving intensity* $\lambda_t(\nu)$ *can be defined in the augmented measure space as*

$$N[\lambda]((s,t] \times B) := \int_s^t \int_B \int_{\mathbb{R}} \mathbf{1}_{0 \le \xi \le \lambda_\tau(\nu)} N(\mathrm{d}\tau, \mathrm{d}\nu, \mathrm{d}\xi), \tag{B.1}$$

where $N(\mathrm{d}\tau, \mathrm{d}\nu, \mathrm{d}\xi)$ *is the Poisson random measure on* $\mathbb{R}^+ \times \mathbb{D} \times \mathbb{R}$ *w.r.t. measure* $\nu(\mathrm{d}y)\mathrm{d}\xi$*.*

The following theorem provides the change of measure theorem for Poisson random measure with evolving intensity, which is crucial for the theoretical analysis of numerical schemes for discrete diffusion models.

Theorem B.4 (Change of Measure for Poisson Random Measure with Evolving Density [38, Thm. 3.3])**.** *Let* $N[\lambda](\mathrm{d}t, \mathrm{d}\nu)$ *be a Poisson random measure with evolving intensity* $\lambda_t(\nu)$*, and* $h_t(\nu)$ *a positive predictable process on* $\mathbb{R}^+ \times \mathbb{D} \times \Omega$*. Suppose the following exponential process is a local* $\mathcal{F}_t$*-martingale:*

$$Z_t[h] := \exp\left(\int_0^t \int_{\mathbb{D}} \log h_t(\nu) N[\lambda](\mathrm{d}t \times \mathrm{d}\nu) - \int_0^t \int_{\mathbb{D}} (h_t(\nu) - 1)\lambda_t(\nu)\gamma(\mathrm{d}\nu) \right), \tag{B.2}$$

and $\mathbb{Q}$ *is another probability measure on* $(\Omega, \mathcal{F})$ *such that* $\mathbb{Q} \ll \mathbb{P}$ *with Radon-Nikodym derivative* $\mathrm{d}\mathbb{Q}/\mathrm{d}\mathbb{P}|_{\mathcal{F}_t} = Z_t[h]$*.*

Then the Poisson random measure $N[\lambda](\mathrm{d}t, \mathrm{d}\nu)$ *under the measure* $\mathbb{Q}$ *is a Poisson random measure with evolving intensity* $\lambda_t(\nu)h_t(\nu)$*.*

B.2 Error Analysis of τ-leaping

The τ-leaping method was originally proposed by [52] and adopted for the inference of discrete diffusion models by [22]. A summary of the algorithm is given in Alg. 3. In this subsection, we provide a sketch for the error analysis of the τ-leaping method when applied to discrete diffusion models, which will be compared with that of high-order schemes later on.

Algorithm 3: τ-Leaping Method for Discrete Diffusion Model Inference

Input: $\widehat{y}_0 \sim q_0$, $\theta \in [0,1]$, time discretization $(s_n, \rho_n)_{n \in [0:N-1]}$, $\widehat{\mu}$, $\widehat{\mu}^*$ as defined in Prop. C.2.
Output: A sample $\widehat{y}_{s_N} \sim \widehat{q}^{\mathrm{RK}}_{t_N}$.

```
for n = 0 to N − 1 do
    ŷ_{s_{n+1}} ← ŷ_{s_n} + Σ_{ν∈D} νP(μ̂_{s_n}(ν)Δ_n);
end
```

Proof of Thm. 3.1. As we are considering the case where $\mathbb{X} = [S]^d$, *i.e.* the state space is a d-dimensional grid with S states along each dimension, we have $\log|\mathbb{X}| = d\log S$. Then we consider a simple time-homogeneous transition matrix $\boldsymbol{Q}_t \equiv \boldsymbol{Q}$ that allows jumps between neighboring states with equal probability. Specifically, we have

$$Q(y,x) = \begin{cases} 1, & \|x-y\|_1 = 1, \\ -2d, & x = y, \end{cases}$$

which can be verified to satisfy Assumption 4.3(i) in [38] with $C = 1$ and $\underline{D} = \overline{D} = 2d$. Assumption 4.3(ii) is also satisfied, as shown in Example B.10 of [38].

Then we may apply Thm. 4.7 in [38] by using the required time discretization scheme according to the properties of the target distribution and plugging in the corresponding values of $C, \underline{D}, \overline{D}$. The result follows by scaling the transition matrix $\boldsymbol{Q}$ by $\frac{1}{d}$, equivalent to scaling the time by d. □

C Proofs

In this section, we provide the missing proofs in the main text. We will first provide the proofs of the stochastic integral formulations of high-order schemes for discrete diffusion models in App. C.1. Then we will provide the proofs of the main results for the θ-Trapezoidal method in App. C.2 and the θ-RK-2 method in App. C.3. We remark that the proof for the θ-Trapezoidal method requires more techniques and is more involved, to which the proof for the θ-RK-2 method is analogous. In App. C.4, we provide the detailed lemmas and computations omitted in the proofs of Thms. 5.4 and 5.5.

C.1 Stochastic Integral Formulations of High-Order Schemes

In order to rigorously analyze the θ-RK-2 method, we need the following definition:

Definition C.1 (Intermediate Process)**.** *We define the intermediate process $\widehat{y}^*_s$ piecewisely on $(s_n, s_{n+1}]$ as follows:*

$$\widehat{y}^*_s = \widehat{y}_{s_n} + \int_{s_n}^{s} \int_{\mathbb{D}} \nu N\left[\widehat{\mu}_{s_n}\right](\mathrm{d}s, \mathrm{d}\nu), \tag{C.1}$$

where the intensity $\widehat{\mu}_{s_n}$ is given by $\widehat{\mu}_{s_n}(\nu, \widehat{y}_{s_n}) = \overleftarrow{\widehat{s}}_{s_n}(\widehat{y}_{s_n}, \widehat{y}_{s_n}+\nu)\overleftarrow{\widehat{Q}}^0_{s_n}(\widehat{y}_{s_n}, \widehat{y}_{s_n}+\nu)$, i.e.*, $\widehat{y}^*_s$ is the process obtained by performing τ-leaping from time s_n to s with intensity $\widehat{\mu}$.*

The following proposition provides the stochastic integral formulation of this method.

Proposition C.2 (Stochastic Integral Formulation of θ-RK-2 Method)**.** *The θ-RK-2 method (Alg. 1) is equivalent to solving the following stochastic integral:*

$$\widehat{y}^{\mathrm{RK}}_s = \widehat{y}^{\mathrm{RK}}_0 + \int_0^s \int_{\mathbb{D}} \nu N\left[\widehat{\mu}^{\mathrm{RK}}\right](\mathrm{d}s, \mathrm{d}\nu), \tag{C.2}$$

in which the intensity $\widehat{\mu}^{\mathrm{RK}}$ is defined as a weighted sum

$$\widehat{\mu}^{\mathrm{RK}}_s(\nu) = (1-\tfrac{1}{2\theta})\widehat{\mu}_{\lfloor s\rfloor}(\nu, \widehat{y}^{\mathrm{RK}}_{\lfloor s\rfloor}) + \tfrac{1}{2\theta}\widehat{\mu}^*_{\rho_s}(\nu, \widehat{y}^*_{\rho_s}), \tag{C.3}$$

and the intermediate intensity $\widehat{\mu}^*$ is defined piecewisely as

$$\widehat{\mu}^*_s(\nu, \widehat{y}^*_s) = \overleftarrow{\widehat{s}}_s(\widehat{y}^*_s, \widehat{y}^*_s+\nu)\overleftarrow{\widehat{Q}}^0_s(\widehat{y}^*_s, \widehat{y}^*_s+\nu), \tag{C.4}$$

with the intermediate process $\widehat{y}^*_s$ defined in (C.1) for the corresponding interval. We will call $\widehat{y}^{\mathrm{RK}}_s$ the interpolating process *of the θ-RK-2 method and denote the distribution of $\widehat{y}^{\mathrm{RK}}_s$ by $\widehat{q}^{\mathrm{RK}}_s$.*

The following proposition establishes the stochastic integral formulation of the θ-Trapezoidal method, whose proof can be found in App. C.1.

Proposition C.3 (Stochastic Integral Formulation of θ-Trapezoidal Method)**.** *The θ-Trapezoidal method (Alg. 2) is equivalent to solving the following stochastic integral:*

$$\widehat{y}^{\mathrm{trap}}_s = \widehat{y}^{\mathrm{trap}}_0 + \int_0^s \int_{\mathbb{D}} N[\widehat{\mu}^{\mathrm{trap}}](\mathrm{d}s, \mathrm{d}\nu) \tag{C.5}$$

where the intensity $\widehat{\mu}^{\mathrm{trap}}$ is defined piecewisely as

$$\widehat{\mu}^{\mathrm{trap}}_s(\nu) = \mathbf{1}_{s<\rho_s}\widehat{\mu}_{\lfloor s\rfloor}(\nu, \widehat{y}^{\mathrm{trap}}_{\lfloor s\rfloor}) + \mathbf{1}_{s\geq\rho_s}\left(\alpha_1\widehat{\mu}^*_{\rho_s}(\nu, \widehat{y}^*_{\rho_s}) - \alpha_2\widehat{\mu}_{\lfloor s\rfloor}(\nu, \widehat{y}^{\mathrm{trap}}_{\lfloor s\rfloor})\right)_+. \tag{C.6}$$

Above, $\mathbf{1}_{(\cdot)}$ denotes the indicator function and the intermediate process $\widehat{y}^*_s$ is defined in (C.1) for the corresponding interval. We will call the process $\widehat{y}^{\mathrm{trap}}_s$ the interpolating process *of the θ-Trapezoidal method and denote the distribution of $\widehat{y}^{\mathrm{trap}}_s$ by $\widehat{q}^{\mathrm{trap}}_s$.*

Proof of Props. C.2 and C.3. Without loss of generality, we give the proof on the interval $(s_n, s_{n+1}]$ for $n \in [0 : N-1]$, and the generalization to the whole interval $[0, T]$ is straightforward.

Notice that once we condition on the filtration $\mathcal{F}_{s_n}$ and construct the intermediate process $\widehat{y}^*_s$ as specified in (C.1) along the interval $(s_n, s_{n+1}]$, the intermediate intensity $\widehat{\mu}^*$ and the piecewise intensity $\widehat{\mu}_{\lfloor s\rfloor}$ do not evolve with time s or the interpolating processes $\widehat{y}^{\mathrm{RK}}_s$ (or $\widehat{y}^{\mathrm{trap}}_s$, respectively) since it

only depends on the state, the intensity at the beginning of the interval s_n and other randomness that is independent of the interpolating process.

Therefore, the stochastic integral on this interval can be rewritten as for the θ-RK-2 scheme that

$$\begin{aligned}\widehat{y}^{\mathrm{RK}}_{s_{n+1}} &= \widehat{y}^{\mathrm{RK}}_{s_n} + \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \nu N[\widehat{\mu}^{\mathrm{trap}}](\mathrm{d}s, \mathrm{d}\nu) \\ &= \widehat{y}^{\mathrm{RK}}_{s_n} + \int_{\mathbb{D}} \nu N[\widehat{\mu}^{\mathrm{RK}}]((s_n, s_{n+1}], \mathrm{d}\nu) \\ &= \widehat{y}^{\mathrm{RK}}_{s_n} + \int_{\mathbb{D}} \nu \mathcal{P}(\widehat{\mu}^{\mathrm{RK}}_{s_n}(\nu)(s_{n+1} - s_n))\gamma(\mathrm{d}\nu),\end{aligned}$$

and for the θ-Trapezoidal scheme that

$$\begin{aligned}\widehat{y}^{\mathrm{trap}}_{s_{n+1}} &= \widehat{y}^{\mathrm{trap}}_{s_n} + \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \nu N[\widehat{\mu}^{\mathrm{trap}}](\mathrm{d}s, \mathrm{d}\nu) \\ &= \widehat{y}^{\mathrm{trap}}_{s_n} + \int_{\mathbb{D}} \nu N[\widehat{\mu}^{\mathrm{trap}}]((s_n, s_{n+1}], \mathrm{d}\nu) \\ &= \widehat{y}^{\mathrm{trap}}_{s_n} + \int_{\mathbb{D}} \nu \mathcal{P}(\widehat{\mu}^{\mathrm{trap}}_{s_n}(\nu)(s_{n+1} - s_n))\gamma(\mathrm{d}\nu),\end{aligned}$$

and the statement follows by taking $\gamma(\mathrm{d}\nu)$ as the counting measure. □

Remark C.4 (Remark on Rejection Sampling and Periodicity Assumption)**.** *The rejection sampling procedure in both algorithms (Algs. 1 and 2) guarantees well-posedness in the rare scenarios where a large drawn value of Poisson random variables or multiple simultaneous jumps in one coordinate would result in an update out of the state space $\mathbb{X} = [S]^d$. To enforce this, we simply allow at most one jump per update across the summation, for example, in the update*

$$\widehat{y}^*_{\rho_n} \leftarrow \widehat{y}_{s_n} + \sum_{\nu \in \mathbb{D}} \nu \mathcal{P}\left(\widehat{\mu}_{s_n}(\nu)\theta\Delta_n\right),$$

as the standard practice in the literature [22, 38]. The indicator function $\mathbf{1}_{\widehat{\mu}_{s_n} > 0}$ in Alg. 1 is also used to ensure that only valid jumps from the current state $\widehat{y}_{s_n}$ are considered, while in Alg. 2, this is implicitly guaranteed by taking the positive part of $\alpha_1\widehat{\mu}^*_{\rho_n} - \alpha_2\widehat{\mu}_{s_n}$, which implies the positivity of $\alpha_1\widehat{\mu}^*_{\rho_n}$ and thus the validity of the jumps $\widehat{y}^*_{\rho_n}$. We point out that the single-jump rule is only a convenient sufficient condition, one should notice that this condition is not necessary for the well-posedness of our algorithms, since our setting of the state space $\mathbb{X}$ carries both orderliness and algebraic structure, and thus one could in principle admit multiple simultaneous jumps without ambiguity.

Over the full inference process, the total probability of rejection is at most $\mathcal{O}(\kappa)$. Below, we give a brief justification and we refer to Proposition A.14 in [38] for a complete proof of this claim. During the update aforementioned, the probability of at least two jumps occurring is bounded by

$$\begin{aligned}&\mathbb{P}\left(\sum_{\nu \in \mathbb{D}} \mathcal{P}\left(\widehat{\mu}_{s_n}(\nu)\theta\Delta_n\right) > 1\right) = 1 - \mathbb{P}\left(\mathcal{P}\left(\sum_{\nu \in \mathbb{D}} \widehat{\mu}_{s_n}(\nu)\theta\Delta_n\right) \leq 1\right) \\ =&1 - \exp\left(-\sum_{\nu \in \mathbb{D}} \widehat{\mu}_{s_n}(\nu)\theta\Delta_n\right)\left(1 + \sum_{\nu \in \mathbb{D}} \widehat{\mu}_{s_n}(\nu)\theta\Delta_n\right) \\ \lesssim&\left(\sum_{\nu \in \mathbb{D}} \widehat{\mu}_{s_n}(\nu)\theta\Delta_n\right)^2 \lesssim \Delta_n^2.\end{aligned}$$

Summing $\mathcal{O}(\Delta_n^2)$ over N steps gives $\sum_{n=0}^{N-1} \Delta_n^2 \lesssim \kappa T$, and an identical argument applies to the second update in each iteration. Hence, the overall rejection rate is at most $\mathcal{O}(\kappa)$.

When we impose periodic boundary conditions, $\mathbb{X} = [S]^d$ is equipped with a convenient algebraic structure: addition and scalar multiplication are globally well-defined. In that case, Algs. 1 and 2 match exactly the stochastic integral formulations in Props. C.2 and C.3. This alignment removes the need for per-step rejection, streamlines the application of the change-of-measure argument, and greatly simplifies the convergence proofs of Thms. 5.4 and 5.5. Even without periodicity, those theorems hold with probability at least $1 - \mathcal{O}(\kappa)$, as shown above.

C.2 Convergence Analysis of the θ-Trapezoidal Method

Theorem C.5. *Let $\overleftarrow{p}_{0:T-\delta}$ and $\widehat{q}^{\mathrm{trap}}_{0:T-\delta}$ be the path measures of the backward process with the stochastic integral formulation (2.4) and the interpolating process (C.5) of the θ-Trapezoidal method (Alg. 2), then it holds that*

$$\begin{aligned} &D_{\mathrm{KL}}(\overleftarrow{p}_{T-\delta}\|\widehat{q}^{\mathrm{trap}}_{T-\delta}) \le D_{\mathrm{KL}}(\overleftarrow{p}_{0:T-\delta}\|\widehat{q}^{\mathrm{trap}}_{0:T-\delta}) \\ \le& D_{\mathrm{KL}}(\overleftarrow{p}_0\|\widehat{q}_0) + \mathbb{E}\left[\int_0^{T-\delta}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\widehat{\mu}^{\mathrm{trap}}_s(\nu)} - \mu_s(\nu) + \widehat{\mu}^{\mathrm{trap}}_s(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right], \end{aligned} \tag{C.7}$$

where the intensity $\widehat{\mu}^{\mathrm{trap}}$ is defined in (C.5), and the expectation is taken w.r.t. both paths generated by the backward process (2.4) and the randomness of the Poisson random measure used in the first step of each iteration of the algorithm, i.e.*, the construction of the intermediate process (C.1), which is assumed to be independent of that of the backward process.*

Proof. First, we will handle the randomness introduced by the Poisson random measure in the first step of each iteration of the θ-Trapezoidal method. For the ease of presentation, we encode the aforementioned randomness as a random variable ζ and suppose it is still supported on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ while being independent of the backward process. Then for each realization of ζ, the intermediate process $\widehat{y}^*_s$ is constructed as in (C.1) and the corresponding intensity $\widehat{\mu}^*_s$ is defined in (C.4).

Given the stochastic integral formulation of the backward process (2.4) and the interpolating process of the θ-Trapezoidal method (C.5), we have by Thm. B.4 that this particular realization of the path measure $\widehat{q}^{\mathrm{trap}}_{0:T-\delta}$ can be obtained by changing the path measure $\overleftarrow{p}_{0:T-\delta}$ with the Radon-Nikodym derivative

$$Z_t\left[\frac{\widehat{\mu}^{\mathrm{trap}}}{\mu}\right] = \exp\left(-\int_0^t\int_{\mathbb{D}}\log\frac{\mu_s(\nu)}{\widehat{\mu}^{\mathrm{trap}}_s(\nu)}N[\mu](\mathrm{d}s,\mathrm{d}\nu) + \int_0^t\int_{\mathbb{D}}\left(\mu_s(\nu) - \widehat{\mu}^{\mathrm{trap}}_s(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right),$$

i.e.,

$$\begin{aligned} D_{\mathrm{KL}}(\overleftarrow{p}_{0:T-\delta}\|\widehat{q}^{\mathrm{trap}}_{0:T-\delta}|\zeta) &= \mathbb{E}\left[\log Z^{-1}_{T-\delta}\left[\frac{\widehat{\mu}^{\mathrm{trap}}}{\mu}\right]\right] \\ =&\mathbb{E}\left[\int_0^{T-\delta}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\widehat{\mu}^{\mathrm{trap}}_s(\nu)} - \mu_s(\nu) + \widehat{\mu}^{\mathrm{trap}}_s(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right]. \end{aligned}$$

Then it is easy to see by the data processing inequality and the chain rule of KL divergence that

$$\begin{aligned} &D_{\mathrm{KL}}(\overleftarrow{p}_{T-\delta}\|\widehat{q}^{\mathrm{trap}}_{T-\delta}) \le D_{\mathrm{KL}}(\overleftarrow{p}_{0:T-\delta}\|\widehat{q}^{\mathrm{trap}}_{0:T-\delta}) \le \mathbb{E}\left[D_{\mathrm{KL}}(\overleftarrow{p}_{T-\delta}\|\widehat{q}^{\mathrm{trap}}_{T-\delta}|\zeta)\right] \\ =&D_{\mathrm{KL}}(\overleftarrow{p}_0\|\widehat{q}_0) + \mathbb{E}\left[\int_0^{T-\delta}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\widehat{\mu}^{\mathrm{trap}}_s(\nu)} - \mu_s(\nu) + \widehat{\mu}^{\mathrm{trap}}_s(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right], \end{aligned}$$

and the proof is complete. □

In the following, we will provide the outline of the proof of Thm. 5.4, where we leave the proof of several lemmas and detailed calculations to App. C.4 for the clarity of presentation.

Proof of Thm. 5.4. Throughout this proof, including the subsequent lemmas and propositions that will be detailed in App. C.4, we will assume that $(y_s)_{s\in[0,T]}$ is a process generated by the path measure $\overleftarrow{p}_{0:T}$ of the backward process with the stochastic integral formulation (2.4) and set it as the underlying paths of the expectation in (C.7) as required by Thm. C.5. Especially, $y_s \sim \overleftarrow{p}_s$ holds for any $s \in [0, T]$. For simplicity, we will assume that the process y_s is left-continuous at each grid point s_i for $i \in [0 : N]$, which happens with probability one.

We first consider the interval $(s_n, s_{n+1}]$ for $n \in [0 : N-1]$, and thus we have $\lfloor s \rfloor = s_n$ and $\rho_s = \rho_n$. Within this interval, we will denote its intermediate process as appeared in (C.1) as y^*_s, and the corresponding intermediate intensity as appeared in (C.4) as $\widehat{\mu}^*_s$. In the following discussion, we will assume implicitly that the processes are conditioned on the filtration $\mathcal{F}_{s_n}$.

By the definition of the intensity $\widehat{\mu}^{\mathrm{trap}}(\nu)$ as specified in (C.6)

$$\widehat{\mu}_s^{\mathrm{trap}} = \mathbf{1}_{s<\rho_s}\widehat{\mu}_{\lfloor s\rfloor} + \mathbf{1}_{s\geq\rho_s}\left(\alpha_1\widehat{\mu}^*_{\rho_s} - \alpha_2\widehat{\mu}_{\lfloor s\rfloor}\right)_+,$$

we can rewrite the corresponding part of the integral in (C.7) as

$$\begin{aligned}
&\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\widehat{\mu}_s^{\mathrm{trap}}(\nu)} - \mu_s(\nu) + \widehat{\mu}_s^{\mathrm{trap}}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\\
&= \left(\int_{s_n}^{\rho_n} + \int_{\rho_n}^{s_{n+1}}\right)\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\widehat{\mu}_s^{\mathrm{trap}}(\nu)} - \mu_s(\nu) + \widehat{\mu}_s^{\mathrm{trap}}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\\
&= \underbrace{\int_{s_n}^{\rho_n}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\widehat{\mu}_{s_n}(\nu)} - \mu_s(\nu) + \widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s}_{\mathrm{(I)}}\\
&+ \underbrace{\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\alpha_1\widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)} - \mu_s(\nu) + \alpha_1\widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s}_{\mathrm{(II)}},
\end{aligned}$$

where the assumption that $\alpha_1\widehat{\mu}^*_{\rho_s} - \alpha_2\widehat{\mu}_{\lfloor s\rfloor} \geq 0$ for all $s \in [0, T-\delta]$ is applied here for the second term (II) above.

Decomposition of the Integral. Next, we decompose the integral (I) and (II) into several terms, the magnitudes of which or combinations of which are to be bounded.

(i) The first term is decomposed as

$$\mathrm{(I)} = \mathrm{(I.1)} + \mathrm{(I.2)} + \mathrm{(I.3)} + \mathrm{(I.4)},$$

where each term is defined as

$$\begin{aligned}
\mathrm{(I.1)} &= \int_{s_n}^{\rho_n}\int_{\mathbb{D}}\left(\mu_{s_n}(\nu)\log\frac{\mu_{s_n}(\nu)}{\widehat{\mu}_{s_n}(\nu)} - \mu_{s_n}(\nu) + \widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s,\\
\mathrm{(I.2)} &= \int_{s_n}^{\rho_n}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\mu_s(\nu) - \mu_s(\nu) - \mu_{s_n}(\nu)\log\mu_{s_n}(\nu) + \mu_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s,\\
\mathrm{(I.3)} &= \int_{s_n}^{\rho_n}\int_{\mathbb{D}}\left(\mu_s(\nu) - \mu_{s_n}(\nu)\right)\left(\log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right) - \log\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s,\\
\mathrm{(I.4)} &= \int_{s_n}^{\rho_n}\int_{\mathbb{D}}\mu_{s_n}(\nu)\log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\\
&\quad - \int_{s_n}^{\rho_n}\int_{\mathbb{D}}\mu_s(\nu)\log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s.
\end{aligned}$$

(ii) The second term is decomposed as

$$\mathrm{(II)} = \mathrm{(II.1)} + \mathrm{(II.2)} + \mathrm{(II.3)} + \mathrm{(II.4)} + \mathrm{(II.5)} + \mathrm{(II.6)},$$

where each term is defined as

$$\begin{aligned}
\mathrm{(II.1)} &= \alpha_1\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\mu_{\rho_n}(\nu)\log\frac{\mu_{\rho_n}(\nu)}{\widehat{\mu}_{\rho_n}(\nu)} - \mu_{\rho_n}(\nu) + \widehat{\mu}_{\rho_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\\
&\quad - \alpha_2\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\mu_{s_n}(\nu)\log\frac{\mu_{s_n}(\nu)}{\widehat{\mu}_{s_n}(\nu)} - \mu_{s_n}(\nu) + \widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s,\\
\mathrm{(II.2)} &= \int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\mu_s(\nu) - \mu_s(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\\
&\quad - \int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\alpha_1(\mu_{\rho_n}(\nu)\log\mu_{\rho_n}(\nu) - \mu_{\rho_n}(\nu)) - \alpha_2(\mu_{s_n}(\nu)\log\mu_{s_n}(\nu) - \mu_{s_n}(\nu))\right)\gamma(\mathrm{d}\nu)\mathrm{d}s,
\end{aligned}$$

$$
\begin{aligned}
\text{(II.3)} &= \int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \alpha_1 \left(\widehat{\mu}^*_{\rho_n}(\nu) - \widehat{\mu}_{\rho_n}(\nu)\right) \gamma(\mathrm{d}\nu)\mathrm{d}s, \\
\text{(II.4)} &= \int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\alpha_1 \mu_{\rho_n}(\nu) \log \widehat{\mu}_{\rho_n}(\nu) - \alpha_2 \mu_{s_n}(\nu) \log \widehat{\mu}_{s_n}(\nu)\right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\
&\quad - \int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\alpha_1 \mu_{\rho_n}(\nu) - \alpha_2 \mu_{s_n}(\nu)\right) \log \left(\alpha_1 \widehat{\mu}_{\rho_n}(\nu) - \alpha_2 \widehat{\mu}_{s_n}(\nu)\right) \gamma(\mathrm{d}\nu)\mathrm{d}s, \\
\text{(II.5)} &= \int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\alpha_1 \mu_{\rho_n}(\nu) - \alpha_2 \mu_{s_n}(\nu)\right) \log \left(\alpha_1 \widehat{\mu}_{\rho_n}(\nu) - \alpha_2 \widehat{\mu}_{s_n}(\nu)\right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\
&\quad - \int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\alpha_1 \mu_{\rho_n}(\nu) - \alpha_2 \mu_{s_n}(\nu)\right) \log \left(\alpha_1 \widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2 \widehat{\mu}_{s_n}(\nu)\right) \gamma(\mathrm{d}\nu)\mathrm{d}s, \\
\text{(II.6)} &= \int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\alpha_1 \mu_{\rho_n}(\nu) - \alpha_2 \mu_{s_n}(\nu)\right) \log \left(\alpha_1 \widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2 \widehat{\mu}_{s_n}(\nu)\right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\
&\quad - \int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \mu_s(\nu) \log \left(\alpha_1 \widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2 \widehat{\mu}_{s_n}(\nu)\right) \gamma(\mathrm{d}\nu)\mathrm{d}s.
\end{aligned}
$$

Bounding the Error Terms. Then we briefly summarize the intuitions and related techniques used in the bounds of the terms above, and the detailed calculations and proofs of the lemmas and propositions are deferred to App. C.4.

(i) *Error due to estimation error associated with the intensity:* The terms (I.1) and (II.1) are bounded by the assumption on the estimation error of the intensity $\widehat{\mu}_s$ (Assump. 5.3), as

$$
\mathbb{E}\left[(\text{I.1}) + (\text{II.1})\right] \leq \theta \Delta_n \epsilon_{\text{I}} + \alpha_1(1-\theta)\Delta_n \epsilon_{\text{I}} = \theta \Delta_n \epsilon_{\text{I}} + \tfrac{1}{2\theta} \Delta_n \epsilon_{\text{I}} \lesssim \Delta_n \epsilon_{\text{I}},
$$

for any $\theta \in (0, 1]$.

The term (II.4) is bounded by Prop. C.9, as

$$
\mathbb{E}\left[(\text{II.4})\right] \lesssim \Delta_n \epsilon_{\text{II}},
$$

where Jensen's inequality is applied here based on the convexity of the loss.

(ii) *Error related to the smoothness of intensity:* By Cor. C.13, the terms (I.2) and (II.2) are bounded by

$$
\mathbb{E}\left[(\text{I.2}) + (\text{II.2})\right] \leq \Delta_n^3.
$$

By Cor. C.14, the terms (I.4) and (II.6) are bounded by

$$
\mathbb{E}\left[(\text{I.4}) + (\text{II.6})\right] \leq \Delta_n^3.
$$

Intuitively, the bounds on these terms closely relate to the properties of the jump process and quantify the smoothness assumption on the intensity μ_s (Assump. 5.2), especially when the intensity does not vary significantly within the interval $(s_n, s_{n+1}]$. The main technique used for bounding these terms is Dynkin's Formula (Thm. C.10). The third-order accuracy here directly follows from the intuition provided in Sec. 4 based on numerical quadrature.

(iii) *Error involving the intermediate process:* The terms (II.3) and (II.5) are bounded by Prop. C.18 and Cor. C.19 respectively as follows

$$
\mathbb{E}\left[(\text{II.3})\right] \lesssim \Delta_n^3 + \Delta_n^2 \epsilon_{\text{II}}, \quad \text{and} \quad \mathbb{E}\left[(\text{II.5})\right] \lesssim \Delta_n^3 + \Delta_n^2 \epsilon_{\text{II}},
$$

The term (I.3) is bounded by Prop. C.20 as below

$$
\mathbb{E}\left[(\text{I.3})\right] \lesssim \Delta_n^3.
$$

The three terms above all involve the intermediate process y_s^* and the corresponding intermediate density $\widehat{\mu}_s^*$.

In conclusion, by summing up all these terms, we have

$$\begin{aligned}&\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\widehat{\mu}_s^{\mathrm{trap}}(\nu)}-\mu_s(\nu)+\widehat{\mu}_s^{\mathrm{trap}}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\\ \lesssim&\Delta_n(\epsilon_{\mathrm{I}}+\epsilon_{\mathrm{II}})+\Delta_n^3+\Delta_n^2\epsilon_{\mathrm{II}}\lesssim\Delta_n(\epsilon_{\mathrm{I}}+\epsilon_{\mathrm{II}})+\Delta_n^3.\end{aligned}$$

Therefore, the overall error is bounded by first applying Thm. C.5 and then the upper bound derived above to each interval $(s_n, s_{n+1}]$, which yields

$$\begin{aligned}&D_{\mathrm{KL}}(\overleftarrow{p}_{T-\delta}\|\widehat{q}_{T-\delta}^{\mathrm{trap}})\\ \leq&D_{\mathrm{KL}}(\overleftarrow{p}_0\|\widehat{q}_0)+\mathbb{E}\left[\int_0^{T-\delta}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\widehat{\mu}_s^{\mathrm{trap}}(\nu)}-\mu_s(\nu)+\widehat{\mu}_s^{\mathrm{trap}}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right]\\ \lesssim&D_{\mathrm{KL}}(\overleftarrow{p}_0\|\widehat{q}_0)+\sum_{n=0}^{N-1}\left(\Delta_n(\epsilon_{\mathrm{I}}+\epsilon_{\mathrm{II}})+\Delta_n^3\right)\\ \lesssim&\exp(-T)+T(\epsilon_{\mathrm{I}}+\epsilon_{\mathrm{II}})+\kappa^2T,\end{aligned}$$

as desired. □

Remark C.6 (Discussion on the Positivity Assumption)**.** *In the following, we will take the positivity assumption in Thm. 5.4 as an example, and the case of the θ-RK-2 method is similar. In the statement of Thm. 5.4, we have assumed that*

$$\alpha_1\widehat{\mu}_{\rho_s}^*(\nu)-\alpha_2\widehat{\mu}_{\lfloor s\rfloor}(\nu)\geq 0$$

in (C.6) for all $s\in[0,T-\delta]$, which allows us to replace $\left(\alpha_1\widehat{\mu}_{\rho_s}^*(\nu)-\alpha_2\widehat{\mu}_{\lfloor s\rfloor}(\nu)\right)_+$ by the difference itself. [25] showed that this approximation is at most of $\mathcal{O}(\Delta_n^3)$ within the corresponding interval, and [26] further proved that for any order $p\geq 1$, there exists a sufficiently small step size Δ such that this approximation is at least p-th order, i.e.*, of order $\mathcal{O}(\Delta^p)$ for that step.*

We give a brief justification of this assumption here. We consider the expectation of the difference itself, which is given by

$$\begin{aligned}&\mathbb{E}\left[\alpha_1\widehat{\mu}_{\rho_s}^*(\nu)-\alpha_2\widehat{\mu}_{\lfloor s\rfloor}(\nu)\right]=\mathbb{E}\left[\widehat{\mu}_{\lfloor s\rfloor}(\nu)+\alpha_1\left(\widehat{\mu}_{\rho_s}^*(\nu)-\widehat{\mu}_{\rho_s}(\nu)\right)+\alpha_1\left(\widehat{\mu}_{\rho_s}(\nu)-\widehat{\mu}_{\lfloor s\rfloor}(\nu)\right)\right]\\ \gtrsim&1-\alpha_1(\kappa\epsilon_{\mathrm{II}}+\kappa)=1-\mathcal{O}(\kappa),\end{aligned}$$

where we used $\mathbb{E}\left[|\widehat{\mu}_{\rho_s}^*(\nu)-\widehat{\mu}_{\rho_s}(\nu)|\right]\lesssim\kappa\epsilon_{\mathrm{II}}$, as established in (C.17) *and $\mathbb{E}\left[|\widehat{\mu}_{\rho_s}(\nu)-\widehat{\mu}_{\lfloor s\rfloor}(\nu)|\right]\lesssim\kappa$, as shown in* (C.18)*. Therefore, as long as the step sizes Δ_n are sufficiently small, the positivity assumption is valid in the sense that the expectation of the difference is at least $1-\mathcal{O}(\kappa)$.*

C.3 Convergence Analysis of the θ-RK-2 Method

Here we may again apply the data processing inequality and the chain rule of KL divergence to upper bound the error associated with the θ-RK-2 method. A statement of the upper bound is provided in Thm. C.7 below, whose proof is omitted here since it is similar to that of Thm. C.5 above.

Theorem C.7. *Let $\overleftarrow{p}_{0:T-\delta}$ and $\widehat{q}_{0:T-\delta}^{\mathrm{RK}}$ be the path measures of the backward process with the stochastic integral formulation (2.4) and the interpolating process (C.2) of the θ-RK-2 method (Alg. 1), then it holds that*

$$\begin{aligned}&D_{\mathrm{KL}}(\overleftarrow{p}_{T-\delta}\|\widehat{q}_{T-\delta}^{\mathrm{RK}})\leq D_{\mathrm{KL}}(\overleftarrow{p}_{0:T-\delta}\|\widehat{q}_{0:T-\delta}^{\mathrm{RK}})\\ \leq&D_{\mathrm{KL}}(\overleftarrow{p}_0\|\widehat{q}_0)+\mathbb{E}\left[\int_0^{T-\delta}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\widehat{\mu}_s^{\mathrm{RK}}(\nu)}-\mu_s(\nu)+\widehat{\mu}_s^{\mathrm{RK}}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right],\end{aligned}\tag{C.8}$$

where the intensity $\widehat{\mu}^{\mathrm{RK}}$ is defined in (C.2), and the expectation is taken w.r.t. both paths generated by the backward process (2.4) and the randomness of the Poisson random measure used in the first step of each iteration of the algorithm, i.e.*, the construction of the intermediate process (C.1), which is assumed to be independent of that of the backward process.*

Following the same flow as in the proof of Thm. 5.4, we will first provide an outline of the proof of Thm. 5.5, and defer the proof of several key lemmas and detailed calculations to App. C.4 for the clarity of presentation. We will also comment on the differences that may lead to the less desirable numerical properties of the θ-RK-2 method.

Proof of Thm. 5.5. In the following proof sketch, we will be using the same notation as in the proof of Thm. 5.4, and we will assume that the process y_s is left-continuous at each grid point s_i for $i \in [0:N]$. We also start by taking a closer look at the integral within each interval $(s_n, s_{n+1}]$ for $n \in [0:N-1]$, and denote the intermediate process as appeared in (C.1) as y^*_s and the corresponding intermediate intensity as appeared in (C.4) as $\widehat{\mu}^*_s$.

As defined in (C.3), the intensity $\widehat{\mu}^{\mathrm{RK}}(\nu)$ is given by

$$\widehat{\mu}^{\mathrm{RK}}_s(\nu) = \left(1 - \tfrac{1}{2\theta}\right) \widehat{\mu}_{\lfloor s \rfloor}(\nu) + \tfrac{1}{2\theta} \widehat{\mu}^*_{\rho_s}(\nu),$$

which helps us rewrite the corresponding part of the integral in (C.8) as

$$\begin{aligned}
&\int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\mu_s(\nu) \log \frac{\mu_s(\nu)}{\widehat{\mu}^{\mathrm{RK}}_s(\nu)} - \mu_s(\nu) + \widehat{\mu}^{\mathrm{RK}}_s(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\
&= \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \\
&\underbrace{\left(\mu_s(\nu) \log \frac{\mu_s(\nu)}{(1 - \frac{1}{2\theta})\widehat{\mu}_{s_n}(\nu) + \frac{1}{2\theta}\widehat{\mu}^*_{\rho_n}(\nu)} - \mu_s(\nu) + \left(1 - \tfrac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta}\widehat{\mu}^*_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s}_{\text{(III)}}.
\end{aligned}$$

Above we again use the positivity assumption that $(1 - \frac{1}{2\theta})\widehat{\mu}_{\lfloor s \rfloor} + \frac{1}{2\theta}\widehat{\mu}^*_{\rho_s} \geq 0$ for the term (III) above, just as what we have done in the proof and discussion of Thm. 5.4 above.

Decomposition of the Integral. Then we perform a similar decomposition of the integral as in the proof of Thm. 5.4 as follows:

$$\text{(III)} = \text{(III.1)} + \text{(III.2)} + \text{(III.3)} + \text{(III.4)} + \text{(III.5)} + \text{(III.6)},$$

where each term is defined as

$$\begin{aligned}
\text{(III.1)} &= \left(1 - \tfrac{1}{2\theta}\right) \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\mu_{s_n}(\nu) \log \left(\frac{\mu_{s_n}(\nu)}{\widehat{\mu}_{s_n}(\nu)} \right) - \mu_{s_n}(\nu) + \widehat{\mu}_{s_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\
&\quad + \tfrac{1}{2\theta} \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\mu_{\rho_n}(\nu) \log \left(\frac{\mu_{\rho_n}(\nu)}{\widehat{\mu}_{\rho_n}(\nu)} \right) - \mu_{\rho_n}(\nu) + \widehat{\mu}_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s, \\
\text{(III.2)} &= \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\mu_s(\nu) \log \mu_s(\nu) - \mu_s(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\
&\quad - \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \tfrac{1}{2\theta}\right) \left(\mu_{s_n}(\nu) \log \mu_{s_n}(\nu) - \mu_{s_n} \right) + \tfrac{1}{2\theta} \left(\mu_{\rho_n}(\nu) \log \mu_{\rho_n}(\nu) - \mu_{\rho_n}(\nu) \right) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s, \\
\text{(III.3)} &= \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \tfrac{1}{2\theta} \left(\widehat{\mu}^*_{\rho_n}(\nu) - \widehat{\mu}_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s, \\
\text{(III.4)} &= \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \tfrac{1}{2\theta}\right) \mu_{s_n}(\nu) \log \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta} \mu_{\rho_n}(\nu) \log \widehat{\mu}_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\
&\quad - \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \tfrac{1}{2\theta}\right) \mu_{s_n}(\nu) + \tfrac{1}{2\theta} \mu_{\rho_n}(\nu) \right) \log \left(\left(1 - \tfrac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta} \widehat{\mu}_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s, \\
\text{(III.5)} &= \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \tfrac{1}{2\theta}\right) \mu_{s_n}(\nu) + \tfrac{1}{2\theta} \mu_{\rho_n}(\nu) \right) \log \left(\left(1 - \tfrac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta} \widehat{\mu}_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\
&\quad - \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \tfrac{1}{2\theta}\right) \mu_{s_n}(\nu) + \tfrac{1}{2\theta} \mu_{\rho_n}(\nu) \right) \log \left(\left(1 - \tfrac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta} \widehat{\mu}^*_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s, \\
\text{(III.6)} &= \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \tfrac{1}{2\theta}\right) \mu_{s_n}(\nu) + \tfrac{1}{2\theta} \mu_{\rho_n}(\nu) \right) \log \left(\left(1 - \tfrac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta} \widehat{\mu}^*_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s
\end{aligned}$$

$$- \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \mu_s(\nu) \log \left(\left(1 - \tfrac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta} \widehat{\mu}^*_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s.$$

Bounding the Error Terms. Then we briefly summarize the intuitions and related techniques used in the bound of the terms above,. Detailed calculations and proofs of the lemmas and propositions used here are deferred to App. C.4.

(i) *Error due to the intensity estimation:* The terms in $(\mathrm{III}.1)$ are bounded by the assumption on the estimation error of the intensity $\widehat{\mu}_s$ (Assump. 5.3) as follows

$$\mathbb{E}\left[(\mathrm{III}.1)\right] \leq \left(1 - \tfrac{1}{2\theta}\right) \Delta_n \epsilon_{\mathrm{I}} + \tfrac{1}{2\theta} \Delta_n \epsilon_{\mathrm{I}} = \Delta_n \epsilon_{\mathrm{I}},$$

for any $\theta \in (0, 1]$.

(ii) *Error related to the smoothness of intensity:* By Cors. C.16 and C.17, the terms $(\mathrm{III}.2)$ and $(\mathrm{III}.6)$ are bounded by

$$\mathbb{E}\left[(\mathrm{III}.2)\right] \leq \Delta_n^3, \quad \text{and} \quad \mathbb{E}\left[(\mathrm{III}.6)\right] \leq \Delta_n^3,$$

respectively.

(iii) *Error involving the intermediate process:* The term $(\mathrm{III}.3)$ and $(\mathrm{III}.5)$ are bounded in almost the same way as that of Prop. C.18 and Cor. C.19. By simply altering the integral upper limits, we obtain that

$$\mathbb{E}\left[(\mathrm{III}.3)\right] \lesssim \Delta_n^3 + \Delta_n^2 \epsilon_{\mathrm{II}}, \quad \mathbb{E}\left[(\mathrm{III}.5)\right] \lesssim \Delta_n^3 + \Delta_n^2 \epsilon_{\mathrm{II}}.$$

The only term that cannot be directly bounded based on results in App. C.4 is $(\mathrm{III}.4)$, which is given by

$$\begin{aligned}\mathbb{E}\left[(\mathrm{III}.4)\right] = \mathbb{E}\Bigg[& \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \tfrac{1}{2\theta}\right) \mu_{s_n}(\nu) \log \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta} \mu_{\rho_n}(\nu) \log \widehat{\mu}_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\ & - \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \tfrac{1}{2\theta}\right) \mu_{s_n}(\nu) + \tfrac{1}{2\theta} \mu_{\rho_n}(\nu) \right) \log \left(\left(1 - \tfrac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta} \widehat{\mu}_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \Bigg]\end{aligned} \tag{C.9}$$

Recall that in the proof of its counterpart (Prop. C.9), we utilized the convexity of the loss function and the extrapolation nature of the second step in the θ-Trapezoidal method (C.6) to bound the error term. However, the same technique cannot be directly applied to the θ-RK-2 method for any $\theta \in [0, 1]$, as the intensity $\widehat{\mu}_s^{\mathrm{RK}}$ is an interpolation of the intensity $\widehat{\mu}_s$ when $\theta \in (\frac{1}{2}, 1]$. Therefore, below we will first focus on the case when $\theta \in (0, \frac{1}{2}]$.

To be specific, by the assumption on the estimation error (Assump. 5.3), we can reduce (C.9) to

$$\begin{aligned}\mathbb{E}\Bigg[& \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \tfrac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) \log \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta} \mu_{\rho_n}(\nu) \log \widehat{\mu}_{\rho_n}(\nu) \right) \\ & - \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \tfrac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta} \widehat{\mu}_{\rho_n}(\nu) \right) \log \left(\left(1 - \tfrac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta} \widehat{\mu}_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \Bigg],\end{aligned} \tag{C.10}$$

which can then be upper bounded based on Jensen's inequality and the convexity of the loss function for $\theta \in (0, \frac{1}{2}]$.

Summing up the bounds of the terms above, we have

$$\begin{aligned}& \int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\mu_s(\nu) \log \frac{\mu_s(\nu)}{\widehat{\mu}_s^{\mathrm{RK}}(\nu)} - \mu_s(\nu) + \widehat{\mu}_s^{\mathrm{RK}}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\ \lesssim & \Delta_n(\epsilon_{\mathrm{I}} + \epsilon_{\mathrm{II}}) + \Delta_n^3 + \Delta_n^2 \epsilon_{\mathrm{II}} \lesssim \Delta_n(\epsilon_{\mathrm{I}} + \epsilon_{\mathrm{II}}) + \Delta_n^3,\end{aligned}$$

Consequently, the overall error of the θ-RK-2 method is bounded by

$$
\begin{aligned}
&D_{\mathrm{KL}}(\overleftarrow{p}_{T-\delta}\|\widehat{q}^{\mathrm{RK}}_{T-\delta}) \\
&\leq D_{\mathrm{KL}}(\overleftarrow{p}_0\|\widehat{q}_0) + \mathbb{E}\left[\int_0^{T-\delta}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\widehat{\mu}^{\mathrm{RK}}_s(\nu)} - \mu_s(\nu) + \widehat{\mu}^{\mathrm{RK}}_s(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right] \\
&\lesssim D_{\mathrm{KL}}(\overleftarrow{p}_0\|\widehat{q}_0) + \sum_{n=0}^{N-1}\left(\Delta_n(\epsilon_{\mathrm{I}}+\epsilon_{\mathrm{II}}) + \Delta_n^3\right) \\
&\lesssim \exp(-T) + T(\epsilon_{\mathrm{I}}+\epsilon_{\mathrm{II}}) + \kappa^2 T,
\end{aligned}
$$

which suggests that the θ-RK-2 is also of second order when $\theta \in (0, \frac{1}{2}]$. For the other case when $\theta \in (\frac{1}{2}, 1]$, we will provide a brief discussion in the remark below. □

Remark C.8 (Discussions on the case when $\theta \in (\frac{1}{2}, 1]$)**.** *For $\theta \in (\frac{1}{2}, 1]$, the term (C.10) is positive and thus not necessarily bounded. One may wonder if, despite being positive, this term is still of at least second order. However, the answer seems negative. By applying the Dynkin's formula (Thm. C.10 and Cor. C.11) to $\mu_s \log \widehat{\mu}_s$ in the term* (III.4)*, we have that the first integral in (C.9) can be expanded as follows*

$$
\begin{aligned}
&\mathbb{E}\left[\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\left(1-\tfrac{1}{2\theta}\right)\mu_{s_n}(\nu)\log\widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta}\mu_{\rho_n}(\nu)\log\widehat{\mu}_{\rho_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right] \\
=&\tfrac{1}{2\theta}\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\mu_{s_n}(\nu)\log\widehat{\mu}_{s_n}(\nu) + \theta\Delta_n\mathcal{L}\left(\mu_{s_n}(\nu)\log\widehat{\mu}_{s_n}(\nu)\right)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s \\
&+\left(1-\tfrac{1}{2\theta}\right)\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}}\mu_{s_n}(\nu)\log\widehat{\mu}_{s_n}(\nu)\gamma(\mathrm{d}\nu)\mathrm{d}s + \mathcal{O}(\Delta_n^2) \\
=&\Delta_n\int_{\mathbb{D}}\mu_{s_n}(\nu)\log\widehat{\mu}_{s_n}(\nu)\gamma(\mathrm{d}\nu) + \frac{1}{2}\Delta_n^2\int_{\mathbb{D}}\mathcal{L}\left(\mu_{s_n}(\nu)\log\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu) + \mathcal{O}(\Delta_n^3).
\end{aligned}
$$

Similarly, applying Dynkin's formula to the following function

$$
G_s(\nu, y_{s^-}) = \left(\tfrac{1}{2\theta}\mu_s(\nu, y_{s^-}) + \left(1-\tfrac{1}{2\theta}\right)\mu_{s_n}(\nu, y_{s^-})\right)\log\left(\tfrac{1}{2\theta}\widehat{\mu}_s(\nu, y_{s^-}) + \left(1-\tfrac{1}{2\theta}\right)\widehat{\mu}_{s_n}(\nu, y_{s^-})\right),
$$

with $G_0(\nu, y_{s_n}) = \mu_{s_n}(\nu, y_{s_n})\log\widehat{\mu}_{s_n}(\nu, y_{s_n})$ allows us to expand the second integral in (C.9) as below

$$
\begin{aligned}
&\mathbb{E}\left[\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\tfrac{1}{2\theta}\mu_{\rho_n}(\nu) + \left(1-\tfrac{1}{2\theta}\right)\mu_{s_n}(\nu)\right)\log\left(\tfrac{1}{2\theta}\widehat{\mu}_{\rho_n}(\nu) + \left(1-\tfrac{1}{2\theta}\right)\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right] \\
=&\Delta_n\int_{\mathbb{D}}G_{s_n}(y_{s_n})\gamma(\mathrm{d}\nu) + \theta\Delta_n^2\int_{\mathbb{D}}\mathcal{L}G_{s_n}(y_{s_n})\gamma(\mathrm{d}\nu) + \mathcal{O}(\Delta_n^3),
\end{aligned}
$$

where

$$
\begin{aligned}
&\mathcal{L}G_{s_n}(\nu, y_{s_n}) \\
=&\tfrac{1}{2\theta}\partial_s\mu_{s_n}(\nu, y_{s_n})\log\widehat{\mu}_{s_n}(\nu, y_{s_n}) + \tfrac{1}{2\theta}\mu_{s_n}(\nu, y_{s_n})\tfrac{1}{2\theta}\frac{\partial_s\widehat{\mu}_{s_n}(\nu, y_{s_n})}{\widehat{\mu}_{s_n}(\nu, y_{s_n})} \\
+&\tfrac{1}{2\theta}\int_{\mathbb{D}}\mu_{s_n}(\nu, y_{s_n}+\nu')\log\left(\tfrac{1}{2\theta}\widehat{\mu}_s(\nu, y_{s_n}+\nu') + \left(1-\tfrac{1}{2\theta}\right)\widehat{\mu}_{s_n}(\nu, y_{s_n}+\nu')\right)\gamma(\mathrm{d}\nu') \\
-&\tfrac{1}{2\theta}\int_{\mathbb{D}}\mu_{s_n}(\nu, y_{s_n})\log\widehat{\mu}_{s_n}(\nu, y_{s_n})\gamma(\mathrm{d}\nu') \\
+&\left(1-\tfrac{1}{2\theta}\right)\mu_{s_n}(\nu, y_{s_n})\tfrac{1}{2\theta}\frac{\partial_s\widehat{\mu}_{s_n}(\nu, y_{s_n})}{\widehat{\mu}_{s_n}(\nu, y_{s_n})} \\
+&\left(1-\tfrac{1}{2\theta}\right)\int_{\mathbb{D}}\mu_{s_n}(\nu, y_{s_n}+\nu')\log\left(\tfrac{1}{2\theta}\widehat{\mu}_s(\nu, y_{s_n}+\nu') + \left(1-\tfrac{1}{2\theta}\right)\widehat{\mu}_{s_n}(\nu, y_{s_n}+\nu')\right)\gamma(\mathrm{d}\nu') \\
-&\left(1-\tfrac{1}{2\theta}\right)\int_{\mathbb{D}}\mu_{s_n}(\nu, y_{s_n})\log\widehat{\mu}_{s_n}(\nu, y_{s_n})\gamma(\mathrm{d}\nu') \\
=&\tfrac{1}{2\theta}\partial_s\mu_{s_n}(\nu, y_{s_n})\log\widehat{\mu}_{s_n}(\nu, y_{s_n}) + \tfrac{1}{2\theta}\mu_{s_n}(\nu, y_{s_n})\frac{\partial_s\widehat{\mu}_{s_n}(\nu, y_{s_n})}{\widehat{\mu}_{s_n}(\nu, y_{s_n})} \\
+&\tfrac{1}{2\theta}\int_{\mathbb{D}}\mu_{s_n}(\nu, y_{s_n}+\nu')\log\widehat{\mu}_s(\nu, y_{s_n}+\nu')\gamma(\mathrm{d}\nu') \\
+&\left(1-\tfrac{1}{2\theta}\right)\int_{\mathbb{D}}\mu_{s_n}(\nu, y_{s_n}+\nu')\log\widehat{\mu}_s(\nu, y_{s_n}+\nu')\gamma(\mathrm{d}\nu') \\
-&\tfrac{1}{2\theta}\int_{\mathbb{D}}\mu_{s_n}(\nu, y_{s_n})\log\widehat{\mu}_{s_n}(\nu, y_{s_n})\gamma(\mathrm{d}\nu') - \left(1-\tfrac{1}{2\theta}\right)\int_{\mathbb{D}}\mu_{s_n}(\nu, y_{s_n})\log\widehat{\mu}_{s_n}(\nu, y_{s_n})\gamma(\mathrm{d}\nu').
\end{aligned}
$$

This further implies that

$$
\begin{aligned}
\theta\mathcal{L}G_{s_n}(y_{s_n}) = &\frac{1}{2}\mathcal{L}\left(\mu_{s_n}(\nu)\log\widehat{\mu}_{s_n}(\nu)\right) \\
&+\tfrac{1}{2\theta}\int_{\mathbb{D}}\left(\mu_{s_n}(\nu, y_{s_n}+\nu')\log\widehat{\mu}_s(\nu, y_{s_n}+\nu') - \mu_{s_n}(\nu, y_{s_n})\log\widehat{\mu}_{s_n}(\nu, y_{s_n})\right)\gamma(\mathrm{d}\nu').
\end{aligned}
$$

Comparing the first and second order terms in the two expansions of the two integrals in (C.9) above then implies that the term (III.4) *is of at most second order.*

C.4 Lemmas and Propositions

In this section, we provide the detailed proofs of the lemmas and propositions omitted in the proof of Thms. 5.4 and 5.5.

Error due to the Intensity Estimation. Apart from the terms (I.1) and (II.1) in the proof of Thm. 5.4 and the term (III.1) in the proof of Thm. 5.5, we also need to bound the error terms (II.4) in terms of the intensity estimation error, which is given by the following proposition. Notably, the following bound also utilizes the convexity of the loss function and the extrapolation nature of the second step in the θ-Trapezoidal method (C.6).

Proposition C.9. *For the interval $(s_n, s_{n+1}]$ for $n \in [0 : N-1]$, we have the following error bound:*

$$
\begin{aligned}
\mathbb{E}\left[(\mathrm{II}.4)\right] =&\mathbb{E}\Bigg[\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\alpha_1\mu_{\rho_n}(\nu)\log\widehat{\mu}_{\rho_n}(\nu) - \alpha_2\mu_{s_n}(\nu)\log\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s \\
&-\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\alpha_1\mu_{\rho_n}(\nu) - \alpha_2\mu_{s_n}(\nu)\right)\log\left(\alpha_1\widehat{\mu}_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\Bigg] \\
\lesssim&\Delta_n\epsilon_{\mathrm{II}}.
\end{aligned}
\tag{C.11}
$$

Proof. We first define and bound three error terms (II.4.1), (II.4.2), and (II.4.3) with score estimation error (Assump. 5.3) as follows:

$$\begin{aligned}\mathbb{E}\left[|(\text{II.4.1})|\right] =& \mathbb{E}\left[\left|\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}} \alpha_1\left(\mu_{\rho_n}(\nu)\log\widehat{\mu}_{\rho_n}(\nu) - \widehat{\mu}_{\rho_n}(\nu)\log\widehat{\mu}_{\rho_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right|\right] \\ \leq& \alpha_1\mathbb{E}\left[\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}} \left|\mu_{\rho_n}(\nu) - \widehat{\mu}_{\rho_n}(\nu)\right|\left|\log\widehat{\mu}_{\rho_n}(\nu)\right|\gamma(\mathrm{d}\nu)\mathrm{d}s\right] \\ \lesssim& \mathbb{E}\left[\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}} \left|\mu_{\rho_n}(\nu) - \widehat{\mu}_{\rho_n}(\nu)\right|\gamma(\mathrm{d}\nu)\mathrm{d}s\right] \lesssim \Delta_n\epsilon_{\text{II}},\end{aligned}$$

Similarly, we also have

$$\mathbb{E}\left[|(\text{II.4.2})|\right] = \mathbb{E}\left[\left|\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}} \alpha_2\left(\mu_{s_n}(\nu)\log\widehat{\mu}_{s_n}(\nu) - \widehat{\mu}_{s_n}(\nu)\log\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right|\right] \lesssim \Delta_n\epsilon_{\text{II}},$$

and

$$\begin{aligned}\mathbb{E}\left[|(\text{II.4.3})|\right] =& \mathbb{E}\Bigg[\Bigg|\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}} (\alpha_1\mu_{\rho_n}(\nu) - \alpha_2\mu_{s_n}(\nu))\log\left(\alpha_1\widehat{\mu}_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s \\ &- \int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}} (\alpha_1\widehat{\mu}_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu))\log\left(\alpha_1\widehat{\mu}_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\Bigg|\Bigg] \\ \lesssim& \Delta_n\epsilon_{\text{II}}.\end{aligned}$$

The remaining term $(\text{II.4.4}) = (\text{II.4}) - (\text{II.4.1}) - (\text{II.4.2}) - (\text{II.4.3})$ is then given by

$$\begin{aligned}(\text{II.4.4}) =& \int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}} \left(\alpha_1\widehat{\mu}_{\rho_n}(\nu)\log\widehat{\mu}_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\log\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s \\ &- \int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}} (\alpha_1\widehat{\mu}_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu))\log\left(\alpha_1\widehat{\mu}_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s \leq 0,\end{aligned}$$

where the last inequality follows from Jensen's inequality, *i.e.,*

$$\alpha_1 x\log x - \alpha_2 y\log y \leq (\alpha_1 x - \alpha_2 y)\log(\alpha_1 x - \alpha_2 y),$$

for $\alpha_1, \alpha_2 \geq 0$ and $\alpha_1 - \alpha_2 = 1$. Therefore, by summing up the terms above, we have

$$\mathbb{E}\left[(\text{II.4})\right] \leq \mathbb{E}\left[(\text{II.4.1}) + (\text{II.4.2}) + (\text{II.4.3}) + (\text{II.4.4})\right] \lesssim \Delta_n\epsilon_{\text{II}},$$

and the proof is complete. □

Error Related to the Smoothness of Intensity. Below we first present the Dynkin's formula, which is the most essential tool for the proof of the error related to the smoothness of the intensity.

Theorem C.10 (Dynkin's Formula)**.** *Let $(y_t)_{t\in[0,\tau]}$ be the following process:*

$$y_t = y_0 + \int_0^t\int_{\mathbb{D}} \nu N[\mu](\mathrm{d}s, \mathrm{d}\nu),$$

where $N[\mu](\mathrm{d}s, \mathrm{d}\nu)$ is a Poisson random measure with intensity μ of the form $\mu_s(\nu, y_{s^-})$. For any $f \in C^1([0,\tau]\times\mathbb{X})$, we define the generator of the process $(y_t)_{t\in[0,\tau]}$ as below

$$\mathcal{L}f_t(y) = \lim_{\tau\to 0^+}\left[\frac{f_{t+\tau}(y_{t+\tau}) - f_t(y_t)}{\tau}\middle| y_t = y\right] = \partial_t f_t(y) + \int_{\mathbb{D}} \left(f_t(y+\nu) - f_t(y)\right)\mu_t(\nu, y)\gamma(\mathrm{d}\nu). \tag{C.12}$$

Then we have that

$$\mathbb{E}\left[f_t(y_t)\right] = f_0(y_0) + \mathbb{E}\left[\int_0^t \mathcal{L}f_s(y_s)\mathrm{d}s\right].$$

Proof. The definition and the form of the generator $\mathcal{L}$, as well as the Dynkin's formula are all well-known in the literature of jump processes. We refer readers to detailed discussions on these topics in [244].

Here we take $X(t) = (t, y_t)$, $z = (\nu, \xi)$, $\alpha(t, X(t)) = 0$, $\sigma(t, X(t)) = 0$, $\gamma(t, X(t^-), z) = \nu \mathbf{1}_{0 \le \xi \le \mu_t(\nu, y_{t^-})}$ in the statement of Thm. 1.19 in [244] and replace the compensated Poisson random measure $\widetilde{N}(dt, dz)$ with the Poisson random measure $N(\mathrm{d}s, \mathrm{d}\nu, \mathrm{d}\xi)$ defined as Rmk. B.3. Then we are allowed to use the ordinary Poisson random measure instead of the compensated one since we are working with a finite measure $\gamma(\mathrm{d}\nu)$.

From Thm. 1.22 in [244], we have that

$$\begin{aligned}\mathcal{L}f_t(y) &= \partial_t f_t(y) + \int_{\mathbb{D}} \int_{\mathbb{R}} \left(f_t(y + \nu \mathbf{1}_{0 \le \xi \le \mu_t(\nu, y)}) - f_t(y) \right) \gamma(\mathrm{d}\nu) \mathrm{d}\xi \\ &= \partial_t f_t(y) + \int_{\mathbb{D}} \left(f_t(y + \nu) - f_t(y) \right) \mu_t(\nu, y) \gamma(\mathrm{d}\nu),\end{aligned}$$

and the proof is complete. □

In many cases below, we will need the following first-order expansion of the expectation of the function $f_t(y_t)$ by assuming the second-order smoothness of the function f.

Corollary C.11. *Suppose that the process $(y_t)_{t \in [0,\tau]}$ and the generator $\mathcal{L}$ are defined as in Thm. C.10. If we further assume that $f \in C^2([0, \tau] \times \mathbb{X})$, then it holds that*

$$\mathbb{E}\left[f_t(y_t) \right] = f_0(y_0) + t\mathcal{L}f_0(y_0) + \mathcal{O}(t^2).$$

Proof. We expand the function $f_s(y_s)$ from $t = 0$ as follows

$$\begin{aligned}\mathbb{E}\left[f_t(y_t) \right] =& f_0(y_0) + \mathbb{E}\left[\int_0^t \mathcal{L} f_s(y_s) \mathrm{d}s \right] \\ =& f_0(y_0) + \mathbb{E}\left[\int_0^t \mathcal{L}\left(f_0(y_0) + \int_0^s \mathcal{L} f_\sigma(y_\sigma) \mathrm{d}\sigma \right) \mathrm{d}s \right] \\ =& f_0(y_0) + \mathcal{L}f_0(y_0) t + \mathbb{E}\left[\int_0^t \int_0^s \mathcal{L}^2 f_\sigma(y_\sigma) \mathrm{d}\sigma \mathrm{d}s \right],\end{aligned}$$

where $\mathcal{L}^2$ is the second-order generator of the process $(y_t)_{t \in [0,\tau]}$ defined as follows

$$\begin{aligned}\mathcal{L}^2 f_\sigma(y) &= \mathcal{L}\left(\partial_\sigma f_\sigma(y) + \int_{\mathbb{D}} \left(f_\sigma(y + \nu) - f_\sigma(y) \right) \mu_\sigma(\nu) \gamma(\mathrm{d}\nu) \right) \\ &= \partial_\sigma^2 f_\sigma(y) + 2 \int_{\mathbb{D}} \left(\partial_\sigma f_\sigma(y + \nu) - \partial_\sigma f_\sigma(y) \right) \mu_\sigma(\nu) \gamma(\mathrm{d}\nu) \\ &+ \int_{\mathbb{D}} \left(f_\sigma(y + \nu) - f_\sigma(y) \right) \partial_\sigma \mu_\sigma(\nu) \gamma(\mathrm{d}\nu) \\ &+ \int_{\mathbb{D}} \int_{\mathbb{D}} \left(f_\sigma(y + \nu + \nu') - f_\sigma(y + \nu') - f_\sigma(y + \nu) + f_\sigma(y) \right) \mu_\sigma(\nu) \mu_\sigma(\nu') \gamma(\mathrm{d}\nu) \gamma(\mathrm{d}\nu'),\end{aligned}$$

which is bounded uniformly by a constant based on the assumption on the smoothness of the function f up to the second order and the boundedness of the measure $\gamma(\mathrm{d}\nu)$. Therefore, the second-order term above is of magnitude $\mathcal{O}(t^2)$, and the proof is complete. □

The following lemma provides a general recipe for bounding a combination of errors, which resembles standard analysis performed for numerical quadratures. In fact, the following lemma can be easily proved by Taylor expansion when the process $(y_t)_{t \in [0,\tau]}$ is constant, *i.e.*, $y_t \equiv y$. Cor. C.11 offers an analogous approach to perform the expansion when the process $(y_t)_{t \in [0,\tau]}$ is not constant.

Lemma C.12. *For any function $f \in C^2([0,\tau]\times\mathbb{X})$ and the true backward process $(y_t)_{t\in[0,\tau]}$ defined in (2.4), it holds that*

$$\left|\mathbb{E}\left[\int_0^{\theta\tau} f_0(y_0)\mathrm{d}s + \int_{\theta\tau}^{\tau}\left(\alpha_1 f_{\theta\tau}(y_{\theta\tau}) - \alpha_2 f_0(y_0)\right)\mathrm{d}s - \int_0^{\tau} f_s(y_s)\mathrm{d}s\right]\right| \lesssim \tau^3.$$

Proof. Let $\mathcal{L}$ be the generator defined in Thm. C.10. By applying the Dynkin's formula (Thm. C.10 and Cor. C.11) to the function $f_t(y_t)$ and plugging in the expression of the generator $\mathcal{L}$, we have that

$$\begin{aligned}
&\mathbb{E}\left[\int_0^{\theta\tau} f_0(y_0)\mathrm{d}s - \alpha_2\int_{\theta\tau}^{\tau} f_0(y_0)\mathrm{d}s + \alpha_1\int_{\theta\tau}^{\tau} f_{\theta\tau}(y_{\theta\tau})\mathrm{d}s - \int_0^{\tau} f_s(y_s)\mathrm{d}s\right]\\
=&\theta\tau f_0(y_0) - \alpha_2(1-\theta)\tau f_0(y_0) + \alpha_1(1-\theta)\tau\left(f_0(y_0) + \theta\tau\mathcal{L}f_0(y_0)\right)\\
&-\int_0^{\tau}\left(f_0(y_0) + s\mathcal{L}f_0(y_0)\right)\mathrm{d}s + \mathcal{O}(\tau^3)\\
=&\left(\theta - \alpha_2(1-\theta) + \alpha_1(1-\theta) - 1\right)\tau f_0(y_0) + \alpha_1(1-\theta)\theta\tau^2\mathcal{L}f_0(y_0) - \frac{\tau^2}{2}\mathcal{L}f_0(y_0) + \mathcal{O}(\tau^3),
\end{aligned}$$

which is of the order $\mathcal{O}(\tau^3)$ by noticing that

$$\begin{aligned}
\theta - \alpha_2(1-\theta) + \alpha_1(1-\theta) - 1 &= \left(\tfrac{1}{2\theta(1-\theta)} - \tfrac{\theta^2+(1-\theta)^2}{2\theta(1-\theta)}\right)(1-\theta) - (1-\theta) = 0\\
\alpha_1(1-\theta)\theta - \tfrac{1}{2} &= \tfrac{1}{2\theta(1-\theta)}(1-\theta)\theta - \tfrac{1}{2} = 0,
\end{aligned}$$

and the proof is complete. □

We remark that in Thm. C.10, Cor. C.11, and Lem. C.12, the smoothness of the function f implies that its derivatives up to the relevant order are bounded by constants independent of the time step τ. This condition is verified in the subsequent proofs.

Then we are ready to bound some of the error terms in the proof of Thm. 5.4 with Lem. C.12.

Corollary C.13. *For the interval $(s_n, s_{n+1}]$ for $n \in [0:N-1]$, we have the following error bound:*

$$\begin{aligned}
&\left|\mathbb{E}\left[(\mathrm{I}.2) + (\mathrm{II}.2)\right]\right|\\
=&\Bigg|\mathbb{E}\Bigg[\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\mu_s(\nu) - \mu_s(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\\
&-\int_{s_n}^{\rho_n}\int_{\mathbb{D}}\left(\mu_{s_n}(\nu)\log\mu_{s_n}(\nu) + \mu_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\\
&-\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\alpha_1(\mu_{\rho_n}(\nu)\log\mu_{\rho_n}(\nu) - \mu_{\rho_n}(\nu)) - \alpha_2(\mu_{s_n}(\nu)\log\mu_{s_n}(\nu) - \mu_{s_n}(\nu))\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\Bigg]\Bigg|\\
\lesssim&\Delta_n^3.
\end{aligned}$$

Proof. The bound is obtained by applying Lem. C.12 with f being the function

$$f_s(y_s) = \int_{\mathbb{D}}\mu_s(\nu)\log\mu_s(\nu)\gamma(\mathrm{d}\nu),$$

Strictly speaking, $f_s(y_s)$ is actually in the form of $f_s(y_{s^-})$, but the argument can be easily extended to this case by assuming time continuity of the function f. □

Corollary C.14. *For the interval $(s_n, s_{n+1}]$ for $n \in [0:N-1]$, we have the following error bound:*

$$\begin{aligned}
&\left|\mathbb{E}\left[(\mathrm{I}.4) + (\mathrm{II}.6)\right]\right|\\
=&\Bigg|\mathbb{E}\Bigg[\int_{s_n}^{\rho_n}\int_{\mathbb{D}}\mu_{s_n}(\nu)\log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\\
&+\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}}\left(\alpha_1\mu_{\rho_n}(\nu) - \alpha_2\mu_{s_n}(\nu)\right)\log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\\
&-\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}}\mu_s(\nu)\log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\Bigg]\Bigg| \lesssim \Delta_n^3.
\end{aligned}$$

Proof. Note that the intermediate process y_s^* defined in (C.1) is driven by a Poisson random measure that is independent of the Poisson random measure driving the process y_s within the interval $(s_n, s_{n+1}]$. Therefore, the error bound is obtained by

(1) Taking the expectation w.r.t. the intermediate process y_s^* and thus the intermediate intensity $\widehat{\mu}_s^*$, and

(2) Then applying Lem. C.12 with f being the following function

$$f_s(y_s) = \int_{\mathbb{D}} \mu_s(\nu)\mathbb{E}\left[\log\left(\alpha_1\widehat{\mu}_{\rho_n}^*(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\right]\gamma(\mathrm{d}\nu).$$

The result follows directly. □

Now we turn to the error term (III.6) in Thm. 5.5, for which we need the following variant of Lem. C.12.

Lemma C.15. *For any function $f \in C^2([0,\tau]\times\mathbb{X})$ and the true backward process $(y_t)_{t\in[0,\tau]}$ defined in (2.4), it holds that*

$$\left|\mathbb{E}\left[\int_0^\tau \left(\left(1-\tfrac{1}{2\theta}\right) f_0(y_0) + \tfrac{1}{2\theta} f_{\theta\tau}(y_{\theta\tau})\right)\mathrm{d}s - \int_0^\tau f_s(y_s)\mathrm{d}s\right]\right| \lesssim \tau^3.$$

Proof. The proof is similar to that of Lem. C.12. Specifically, we let $\mathcal{L}$ be the generator defined in Thm. C.10, apply the Dynkin's formula (Thm. C.10 and Cor. C.11) to the function $f_t(y_t)$ and plug in the expression of the generator $\mathcal{L}$, which yields

$$\begin{aligned}
&\mathbb{E}\left[\int_0^\tau \left(\left(1-\tfrac{1}{2\theta}\right) f_0(y_0) + \tfrac{1}{2\theta} f_{\theta\tau}(y_{\theta\tau})\right)\mathrm{d}s - \int_0^\tau f_s(y_s)\mathrm{d}s\right]\\
=&\left(1-\tfrac{1}{2\theta}\right)\tau f_0(y_0) + \tfrac{1}{2\theta}\int_0^\tau \left(f_0(y_0) + \theta\tau\mathcal{L}f_0(y_0)\right)\mathrm{d}s - \int_0^\tau \left(f_0(y_0) + s\mathcal{L}f_0(y_0)\right)\mathrm{d}s + \mathcal{O}(\tau^3)\\
=&\mathcal{O}(\tau^3),
\end{aligned}$$

as desired. □

Corollary C.16. *For the interval $(s_n, s_{n+1}]$ for $n \in [0:N-1]$, we have the following error bound:*

$$\begin{aligned}
&|\mathbb{E}\left[(\text{III.2})\right]|\\
=&\left|\mathbb{E}\Bigg[\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}} \left(\mu_s(\nu)\log\mu_s(\nu) - \mu_s(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right.\\
&\left.-\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}} \left(\left(1-\tfrac{1}{2\theta}\right)\left(\mu_{s_n}(\nu)\log\mu_{s_n}(\nu) - \mu_{s_n}\right) + \tfrac{1}{2\theta}\left(\mu_{\rho_n}(\nu)\log\mu_{\rho_n}(\nu) - \mu_{\rho_n}(\nu)\right)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\Bigg]\right|\\
\lesssim&\ \Delta_n^3.
\end{aligned}$$

Proof. By applying Lem. C.15 with f being the function

$$f_s(y_s) = \int_{\mathbb{D}} \mu_s(\nu)\log\mu_s(\nu)\gamma(\mathrm{d}\nu),$$

we have that the result follows directly. □

Corollary C.17. *For any $n \in [0:N-1]$ and the corresponding interval $(s_n, s_{n+1}]$, we have the following error bound:*

$$\begin{aligned}
&|\mathbb{E}\left[(\text{III.6})\right]|\\
=&\left|\mathbb{E}\Bigg[\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}} \left(\left(1-\tfrac{1}{2\theta}\right)\mu_{s_n}(\nu) + \tfrac{1}{2\theta}\mu_{\rho_n}(\nu)\right)\log\left(\left(1-\tfrac{1}{2\theta}\right)\widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta}\widehat{\mu}_{\rho_n}^*(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right.\\
&\left.-\int_{s_n}^{s_{n+1}}\int_{\mathbb{D}} \mu_s(\nu)\log\left(\left(1-\tfrac{1}{2\theta}\right)\widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta}\widehat{\mu}_{\rho_n}^*(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\Bigg]\right| \lesssim \Delta_n^3.
\end{aligned}$$

Proof. Following the arguments in the proof of Cor. C.14, the error bound is obtained by first taking the expectation w.r.t. the intermediate process y_s^* and thus the intermediate intensity $\widehat{\mu}_s^*$, and then applying Lem. C.15 with f being the function

$$f_s(y_s) = \int_{\mathbb{D}} \mu_s(\nu) \mathbb{E}\left[\log\left(\left(1 - \tfrac{1}{2\theta}\right)\widehat{\mu}_{s_n}(\nu) + \tfrac{1}{2\theta}\widehat{\mu}^*_{\rho_n}(\nu)\right)\right]\gamma(\mathrm{d}\nu),$$

as desired. □

Error involving the Intermediate Process.

Proposition C.18. *For the interval $(s_n, s_{n+1}]$ with $n \in [0 : N-1]$, we have the following error bound:*

$$\mathbb{E}\left[(\mathrm{II}.3)\right] = \mathbb{E}\left[\int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\widehat{\mu}^*_{\rho_n}(\nu) - \widehat{\mu}_{\rho_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right] \lesssim \Delta_n^3 + \Delta_n^2 \epsilon_{\mathrm{II}}.$$

Proof. First, we rewrite the error term (II.3) as

$$\begin{aligned}\mathbb{E}\left[(\mathrm{II}.3)\right] &= \mathbb{E}\left[\int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\widehat{\mu}^*_{\rho_n}(\nu) - \widehat{\mu}_{\rho_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right] \\ &\lesssim \int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\mathbb{E}\left[\widehat{\mu}^*_{\rho_n}(\nu)\right] - \mathbb{E}\left[\widehat{\mu}_{\rho_n}(\nu)\right]\right)\gamma(\mathrm{d}\nu)\mathrm{d}s.\end{aligned} \tag{C.13}$$

Then we expand the integrand by applying the Dynkin's formula (Thm. C.10 and Cor. C.11) to the function $\widehat{\mu}_s(\nu)$ w.r.t. the intermediate process $(y_s^*)_{s\in[s_n,\rho_n]}$ and the process $(y_s)_{s\in[s_n,\rho_n]}$ respectively as follows

$$\begin{aligned}&\mathbb{E}\left[\widehat{\mu}^*_{\rho_n}(\nu)\right] - \mathbb{E}\left[\widehat{\mu}_{\rho_n}(\nu)\right] \\ =&\mathbb{E}\left[\widehat{\mu}_{s_n}(\nu) + \mathcal{L}^*\widehat{\mu}_{s_n}(\nu)\Delta_n + \mathcal{O}(\Delta_n^2)\right] - \mathbb{E}\left[\widehat{\mu}_{s_n}(\nu) + \mathcal{L}\widehat{\mu}_{s_n}(\nu)\Delta_n + \mathcal{O}(\Delta_n^2)\right] \\ =&\mathbb{E}\left[(\mathcal{L}^* - \mathcal{L})\widehat{\mu}_{s_n}(\nu)\Delta_n\right] + \mathcal{O}(\Delta_n^2),\end{aligned}$$

where the generators $\mathcal{L}^*$ and $\mathcal{L}$ are defined as in (C.12) w.r.t. the processes $(y_s^*)_{s\in[s_n,\rho_n]}$ and $(y_s)_{s\in[s_n,\rho_n]}$, respectively, *i.e.*, for any function $f \in C^1([s_n, \rho_n] \times \mathbb{X})$, we have

$$\begin{aligned}\mathcal{L}^* f_s(y) &= \partial_s f_s(y) + \int_{\mathbb{D}} \left(f_s(y+\nu) - f_s(y)\right)\widehat{\mu}_{s_n}(\nu)\gamma(\mathrm{d}\nu), \\ \mathcal{L} f_s(y) &= \partial_s f_s(y) + \int_{\mathbb{D}} \left(f_s(y+\nu) - f_s(y)\right)\mu_s(\nu)\gamma(\mathrm{d}\nu).\end{aligned} \tag{C.14}$$

Therefore, for the term $\mathbb{E}\left[|(\mathcal{L}^* - \mathcal{L})\widehat{\mu}_{s_n}(\nu)|\right]$ evaluated at $s = s_n$, we have

$$\begin{aligned}\mathbb{E}\left[|(\mathcal{L}^* - \mathcal{L})\widehat{\mu}_{s_n}(\nu)|\right] &= \mathbb{E}\left[\left|\int_{\mathbb{D}} \left(\widehat{\mu}_{s_n}(y+\nu) - \widehat{\mu}_{s_n}(y)\right)\left(\widehat{\mu}_{s_n}(\nu) - \mu_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\right|\right] \\ &\lesssim \mathbb{E}\left[\int_{\mathbb{D}} \left|\widehat{\mu}_{s_n}(\nu) - \mu_{s_n}(\nu)\right|\gamma(\mathrm{d}\nu)\right] \lesssim \epsilon_{\mathrm{II}},\end{aligned} \tag{C.15}$$

where we used the assumption on the estimation error (Assump. 5.3) in the last inequality. Then we can further reduce (C.13) to

$$\int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\mathbb{E}\left[\widehat{\mu}^*_{\rho_n}(\nu)\right] - \mathbb{E}\left[\widehat{\mu}_{\rho_n}(\nu)\right]\right)\gamma(\mathrm{d}\nu)\mathrm{d}s \lesssim \int_{\rho_n}^{s_{n+1}} \left(\epsilon_{\mathrm{II}}\Delta_n + \mathcal{O}(\Delta_n^2)\right)\mathrm{d}s \lesssim \epsilon_{\mathrm{II}}\Delta_n^2 + \Delta_n^3,$$

and the proof is complete. □

Corollary C.19. *For the interval $(s_n, s_{n+1}]$ for $n \in [0 : N-1]$, we have the following error bound:*

$$\begin{aligned}\mathbb{E}\left[(\mathrm{II}.5)\right] =&\mathbb{E}\Big[\int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} (\alpha_1\mu_{\rho_n}(\nu) - \alpha_2\mu_{s_n}(\nu)) \log\left(\alpha_1\widehat{\mu}_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s \\ &- \int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} (\alpha_1\mu_{\rho_n}(\nu) - \alpha_2\mu_{s_n}(\nu)) \log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\Big] \\ \lesssim&\Delta_n^3 + \Delta_n^2\epsilon_{\mathrm{II}}.\end{aligned}$$

Proof. Since the two integrands in (II.5) only differ by replacing $\widehat{\mu}^*_{\rho_n}(\nu)$ with $\widehat{\mu}_{\rho_n}(\nu)$, we have the following upper bound by using the assumption on the boundedness of the intensities (Assump. 5.2 (II))

$$\begin{aligned}
&\mathbb{E}\left[(\mathrm{II}.5)\right]\\
\lesssim&\mathbb{E}\left[\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}}\left|\alpha_1\mu_{\rho_n}(\nu)-\alpha_2\mu_{s_n}(\nu)\right|\frac{1}{\alpha_1\widehat{\mu}_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)}\alpha_1\left|\widehat{\mu}_{\rho_n}(\nu)-\widehat{\mu}^*_{\rho_n}(\nu)\right|\gamma(\mathrm{d}\nu)\mathrm{d}s\right]\\
\lesssim&\mathbb{E}\left[\int_{\rho_n}^{s_{n+1}}\int_{\mathbb{D}}\left|\widehat{\mu}_{\rho_n}(\nu)-\widehat{\mu}^*_{\rho_n}(\nu)\right|\gamma(\mathrm{d}\nu)\mathrm{d}s\right]\\
\lesssim&\Delta_n\mathbb{E}\left[\int_{\mathbb{D}}\left|\widehat{\mu}_{\rho_n}(\nu)-\widehat{\mu}^*_{\rho_n}(\nu)\right|\right]\gamma(\mathrm{d}\nu)
\end{aligned}\tag{C.16}$$

Applying the same arguments as in Prop. C.18, which uses the generators $\mathcal{L}$ and $\mathcal{L}^*$ defined in (C.14), we can bound the RHS above as follows

$$\begin{aligned}
&\mathbb{E}\left[\left|\widehat{\mu}^*_{\rho_n}(\nu)-\widehat{\mu}_{\rho_n}(\nu)\right|\right]\\
=&\mathbb{E}\left[\left|\left(\widehat{\mu}_{s_n}(\nu)+\mathcal{L}^*\widehat{\mu}_{s_n}(\nu)\Delta_n+\mathcal{O}(\Delta_n^2)\right)-\left(\widehat{\mu}_{s_n}(\nu)+\mathcal{L}\widehat{\mu}_{s_n}(\nu)\Delta_n+\mathcal{O}(\Delta_n^2)\right)\right|\right]\\
\lesssim&\Delta_n\mathbb{E}\left[\left|(\mathcal{L}^*-\mathcal{L})\widehat{\mu}_{s_n}(\nu)\right|\right]+\mathcal{O}(\Delta_n^2)\lesssim\Delta_n\epsilon_{\mathrm{II}}+\mathcal{O}(\Delta_n^2)
\end{aligned}\tag{C.17}$$

where the last inequality follows from (C.15). Substituting (C.17) into (C.16) then yields the desired upper bound. □

Proposition C.20. *For the interval $(s_n, s_{n+1}]$ with $n \in [0 : N-1]$, we have the following error bound:*

$$\begin{aligned}
\mathbb{E}\left[(\mathrm{I}.3)\right]=&\mathbb{E}\left[\int_{s_n}^{\rho_n}\int_{\mathbb{D}}\left(\mu_s(\nu)-\mu_{s_n}(\nu)\right)\left(\log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)\right)-\log\widehat{\mu}_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right]\\
\lesssim&\Delta_n^3.
\end{aligned}$$

Proof. First, we observe by Dynkin's formula (Thm. C.10) that

$$\mathbb{E}\left[\left|\mu_s(\nu)-\mu_{s_n}(\nu)\right|\right]=\mathbb{E}\left[\left|\int_{s_n}^{s}\mathcal{L}\mu_{s_n}\mathrm{d}s+\mathcal{O}(\Delta_n^2)\right|\right]\lesssim\Delta_n,$$

and also

$$\mathbb{E}\left[\left|\widehat{\mu}_s(\nu)-\widehat{\mu}_{s_n}(\nu)\right|\right]=\mathbb{E}\left[\left|\int_{s_n}^{s}\mathcal{L}^*\widehat{\mu}_{s_n}\mathrm{d}s+\mathcal{O}(\Delta_n^2)\right|\right]\lesssim\Delta_n.\tag{C.18}$$

Secondly, applying the given assumption (Assump. 5.2 (II)) on the boundedness of the intensities yields

$$\begin{aligned}
&\mathbb{E}\left[\left|\log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)\right)-\log\widehat{\mu}_{s_n}(\nu)\right|\right]\\
\lesssim&\frac{1}{\widehat{\mu}_{s_n}(\nu)}\mathbb{E}\left[\left|\alpha_1\widehat{\mu}^*_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)-\widehat{\mu}_{s_n}(\nu)\right|\right]\\
\lesssim&\mathbb{E}\left[\left|\alpha_1\widehat{\mu}^*_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)-\widehat{\mu}_{s_n}(\nu)\right|\right]\\
\lesssim&\mathbb{E}\left[\alpha_1\left|\widehat{\mu}^*_{\rho_n}(\nu)-\widehat{\mu}_{s_n}(\nu)\right|\right]\\
\lesssim&\mathbb{E}\left[\left|\widehat{\mu}^*_{\rho_n}(\nu)-\widehat{\mu}_{\rho_n}(\nu)\right|\right]+\mathbb{E}\left[\left|\widehat{\mu}_{\rho_n}(\nu)-\widehat{\mu}_{s_n}(\nu)\right|\right]\\
\lesssim&\Delta_n+\Delta_n\epsilon_{\mathrm{II}}+\mathcal{O}(\Delta_n^2)\lesssim\Delta_n
\end{aligned}\tag{C.19}$$

where the last inequality follows from (C.17) proved above. Therefore, we may further deduce that

$$\begin{aligned}
&\mathbb{E}\left[(\mathrm{I}.3)\right]\\
\leq&\int_{s_n}^{\rho_n}\int_{\mathbb{D}}\mathbb{E}\left[\left|\mu_s(\nu)-\mu_{s_n}(\nu)\right|\right]\\
&\qquad\mathbb{E}\left[\left|\log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)\right)-\log\left(\alpha_1\widehat{\mu}_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)\right)\right|\right]\gamma(\mathrm{d}\nu)\mathrm{d}s\\
\lesssim&\Delta_n^3,
\end{aligned}$$

where the first inequality is due to the independency of y_s and y^*_s for $s \in [s_n, \rho_n]$, and the proof is complete. □

D Details of Numerical Experiments

In Apps. D.1 to D.4, we present additional numerical results for the 15-dimensional toy model, text generation, image generation, and diffusion LLM, respectively.

D.1 15-Dimensional Toy Model

We first derive the closed-form formula of the marginal distributions $\boldsymbol{p}_t$ in this model. Recall that the state space $\mathbb{X} = \{1, 2, ..., d\}$ with $d = 15$, and the initial distribution is $\boldsymbol{p}_0 \in \Delta^d$. The rate matrix at any time is $\boldsymbol{Q} = \frac{1}{d}\boldsymbol{E} - \boldsymbol{I}$. By solving (2.1), we see that

$$\boldsymbol{p}_t = \mathrm{e}^{t\boldsymbol{Q}}\boldsymbol{p}_0 = \left(\frac{1-\mathrm{e}^{-t}}{d}\boldsymbol{E} + \mathrm{e}^{-t}\boldsymbol{I}\right)\boldsymbol{p}_0,$$

and therefore $\boldsymbol{p}_t$ converges to the uniform distribution $\boldsymbol{p}_\infty = \frac{1}{d}\mathbf{1}$ as $t \to \infty$. The formula of $\boldsymbol{p}_t$ directly yields the scores $\boldsymbol{s}_t(x) = \frac{\boldsymbol{p}_t}{p_t(x)}$.

During inference, we initialize at the uniform distribution $\boldsymbol{q}_0 = \boldsymbol{p}_\infty$ and run from time 0 to $T = 12$. The truncation error of this choice of time horizon is of the magnitude of 10^{-12} reflected by $D_{\mathrm{KL}}(\boldsymbol{p}_T \| \boldsymbol{p}_\infty)$, and therefore negligible. The discrete time points form an arithmetic sequence.

We generate 10^6 samples for each algorithm and use `np.bincount` to obtain the empirical distribution $\widehat{\boldsymbol{q}}_T$ as the output distribution. Finally, the KL divergence is computed by

$$D_{\mathrm{KL}}(\boldsymbol{p}_0 \| \widehat{\boldsymbol{q}}_T) = \sum_{i=1}^{d} p_0(i) \log \frac{p_0(i)}{\widehat{q}_T(i)}.$$

We also perform bootstrapping for 1000 times to obtain the 95% confidence interval of the KL divergence, the results are shown by the shaded area in Fig. 2. The fitted lines are obtained by standard linear regression on the log-log scale with the slopes marked beside each line in Fig. 2.

D.2 Text Generation

For text generation, we use the small version of RADD [33] checkpoint[1] trained with λ-DCE loss. We choose an early stopping time $\delta = 10^{-3}$ for a stable numerical simulation. Since RADD is a masked discrete diffusion model, we can freely choose the noise schedule $\sigma(t)$ used in the inference process. We consider the following log-linear noise schedule used in the model training,

$$\sigma(t) = \frac{1-\epsilon}{1-(1-\epsilon)t}, \quad \bar{\sigma}(t) = \int_0^t \sigma(s)\mathrm{d}s = -\log(1-(1-\epsilon)t) \tag{D.1}$$

where we choose $\epsilon = 10^{-3}$.

The score function $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ used for computing the transition rate matrix can be computed from the RADD score model $\boldsymbol{p}_\theta$ using the following formula from [33],

$$\boldsymbol{s}_t^\theta(\boldsymbol{x}_t) = \frac{\mathrm{e}^{-\bar{\sigma}(t)}}{1-\mathrm{e}^{-\bar{\sigma}(t)}}\boldsymbol{p}_\theta(\boldsymbol{x}_t), \tag{D.2}$$

where the model $\boldsymbol{p}_\theta$ is trained to approximate the conditional distribution of the masked positions given all unmasked positions. More specifically, let d be the length of the sequence and $\{1, 2, ..., S\}$ be the vocabulary set (not including the mask token). Then given a partially masked sequence $\boldsymbol{x} = (x^1, ..., x^d)$, the model $\boldsymbol{p}_\theta(\boldsymbol{x})$ outputs a $d \times S$ matrix whose (ℓ, s) element approximates $\mathbb{P}_{\boldsymbol{X}\sim\boldsymbol{p}_{\mathrm{data}}}(x^\ell = s | \boldsymbol{X}^{\mathrm{UM}} = \boldsymbol{x}^{\mathrm{UM}})$ when x^ℓ is mask, and is $\mathbf{1}_{X^\ell, s}$ if otherwise. Here, $\boldsymbol{x}^{\mathrm{UM}}$ represents the unmasked portion of the sequence $\boldsymbol{x}$.

We adopt a uniform discretization of the time interval $(\delta, 1]$. For θ-RK-2 and θ-Trapezoidal, we pick $\theta = \frac{1}{2}$. We compare our proposed θ-RK-2 and θ-Trapezoidal with the Euler method, Tweedie τ-leaping, τ-leaping, and we present full results across all NFEs ranging from 16 to 1024 in Tabs. 4 and 5. For each method, we generate 1024 samples and compute the average perplexities and unigram entropy on both GPT-2 large[2] and LLaMA 3.[3] All the experiments are run on a single NVIDIA A100 GPU.

Table 3: Generative perplexity of texts generated by different sampling algorithms on GPT-2 large. Lower values are better, with the best in **bold**.

Method	NFE $= 16$	NFE $= 32$	NFE $= 64$	NFE $= 128$	NFE $= 256$	NFE $= 512$	NFE $= 1024$
FHS	≤ 307.425	≤ 186.594	≤ 141.625	≤ 122.732	≤ 113.310	≤ 113.026	≤ 109.406
Euler	≤ 277.962	≤ 160.586	≤ 111.597	≤ 86.276	≤ 68.092	≤ 55.622	≤ 44.686
Tweedie τ-leaping	≤ 277.133	≤ 160.248	≤ 110.848	≤ 85.738	≤ 70.102	≤ 55.194	≤ 44.257
τ-leaping	≤ 126.835	≤ 96.321	≤ 69.226	≤ 52.366	≤ 41.694	≤ 33.789	≤ 28.797
Semi-AR	≤ 2857.469	≤ 1543.302	≤ 741.184	≤ 360.793	≤ 222.303	≤ 164.162	≤ 147.406
θ-RK-2	≤ 127.363	≤ 109.351	≤ 86.102	≤ 64.317	≤ 49.816	≤ 40.375	≤ 33.971
θ-Trapezoidal	$\mathbf{\leq 123.585}$	$\mathbf{\leq 89.912}$	$\mathbf{\leq 66.549}$	$\mathbf{\leq 49.051}$	$\mathbf{\leq 39.959}$	$\mathbf{\leq 32.456}$	$\mathbf{\leq 27.553}$

Table 4: Generative perplexity of texts generated by different sampling algorithms on LLaMA 3. Lower values are better, with the best in **bold**.

Method	NFE $= 16$	NFE $= 32$	NFE $= 64$	NFE $= 128$	NFE $= 256$	NFE $= 512$	NFE $= 1024$
FHS	≤ 342.498	≤ 210.742	≤ 155.258	≤ 132.135	≤ 127.526	≤ 123.013	≤ 120.791
Euler	≤ 318.413	≤ 175.555	≤ 125.955	≤ 91.051	≤ 75.245	≤ 59.971	≤ 49.406
Tweedie τ-leaping	≤ 316.744	≤ 172.941	≤ 121.248	≤ 94.253	≤ 75.403	≤ 59.943	≤ 49.239
τ-leaping	≤ 152.867	≤ 117.930	≤ 86.980	≤ 68.090	$\mathbf{\leq 53.664}$	≤ 44.676	≤ 38.293
Semi-AR	≤ 2696.883	≤ 1684.973	≤ 829.391	≤ 410.177	≤ 251.963	≤ 166.927	≤ 162.093
θ-RK-2	≤ 150.439	≤ 132.090	≤ 107.066	≤ 80.742	≤ 63.277	≤ 52.563	≤ 44.687
θ-Trapezoidal	$\mathbf{\leq 146.027}$	$\mathbf{\leq 113.260}$	$\mathbf{\leq 83.456}$	$\mathbf{\leq 66.071}$	≤ 54.307	$\mathbf{\leq 44.293}$	$\mathbf{\leq 35.524}$

From the results in Tabs. 3 to 5, we observe that θ-Trapezoidal almost outperforms all other approaches and generates samplers with better perplexities across almost all NFEs. We also noticed that both the Euler method and Tweedie τ-leaping perform similarly, and are beaten by a large margin by θ-RK-2 and τ-leaping.

In Tab. 6, we present the percentage of positive extrapolated intensities for different algorithms across NFE values. This partially validates the assumption in our theoretical analysis (Thms. 5.4 and 5.5) that the intensity remains positive throughout the sampling process.

D.3 Image Generation

For the image generation, we use the checkpoint of MaskGIT [63, 68] reproduced in Pytorch[4]. Recall that the MaskGIT is a masked image model which, given a partially masked sequence, outputs the conditional distributions of the masked positions given the unmasked portion, just like the model $\boldsymbol{p}_\theta(\cdot)$ in the aforementioned masked text model, RADD. Therefore, by similarly introducing a time noise schedule $\sigma(t)$ (for which we adopt the same log-linear schedule (D.1) in our experiment), we obtain a masked discrete diffusion model akin to the RADD. The score function can be computed accordingly using the model output as in (D.2).

We choose an early stopping time $\delta = 10^{-3}$, and adopt a uniform discretization of the time interval $(\delta, 1]$ for θ-RK-2, θ-Trapezoidal, τ-leaping and the Euler method. For parallel decoding, we use a linear randomization strategy in the re-masking step and an $\arccos$ masking scheduler, as recommended in [63]. For each method, we generate 50k samples in a class-conditioned way and compute its FID against the validation split of ImageNet. We use classifier-free guidance to enhance generation quality and set the guidance strength to $w = 3$.

We present the full results for NFE ranging from 4 to 64 in Fig. 5. All the experiments are run on 1 NVIDIA A100. Notably, θ-Trapezoidal with $\theta = \frac{1}{3}$ is the best-performing method except for extremely low NFE budgets. While θ-Trapezoidal with $\theta = \frac{1}{2}$ in general demonstrates a less competitive performance, it converges to the same generation quality as $\theta = \frac{1}{3}$ in the high NFE regime. We also noticed that when using extrapolation with $\theta = \frac{1}{3}$, θ-RK-2 beats τ-leaping for NFE larger than 8, which again accords with our theoretical prediction of its competitive performance in $\theta \in (0, \frac{1}{2}]$ regime.

[1] `https://huggingface.co/JingyangOu/radd-lambda-dce`

[2] `https://huggingface.co/openai-community/gpt2-large`

[3] `https://huggingface.co/meta-llama/Meta-Llama-3-8B`

[4] `https://github.com/valeoai/Maskgit-pytorch`

Table 5: Unigram entropy of texts generated by different sampling algorithms on LLaMA 3.

Method	NFE $= 16$	NFE $= 32$	NFE $= 64$	NFE $= 128$	NFE $= 256$	NFE $= 512$	NFE $= 1024$
FHS	7.843	7.793	7.748	7.712	7.714	7.716	7.717
Euler	7.785	7.677	7.594	7.446	7.343	7.158	6.962
Tweedie τ-leaping	7.786	7.675	7.564	7.453	7.345	7.151	6.970
τ-leaping	7.048	7.122	7.016	6.890	6.706	6.537	6.407
Semi-AR	8.019	8.056	7.994	7.908	7.836	7.771	7.810
θ-RK-2	6.772	7.017	7.085	7.010	6.831	6.682	6.548
θ-Trapezoidal	7.126	7.163	7.033	6.919	6.740	6.532	6.412

Table 6: Percentage of positive extrapolated intensities for different algorithms across NFE values.

Method	NFE $= 32$	NFE $= 64$	NFE $= 128$	NFE $= 256$	NFE $= 512$	NFE $= 1024$
θ-RK-2	97.21 ± 3.1	98.31 ± 2.0	98.01 ± 1.3	99.27 ± 0.9	99.44 ± 0.7	99.52 ± 0.6
θ-Trapezoidal	95.67 ± 4.8	97.06 ± 3.6	98.22 ± 2.4	98.87 ± 1.6	99.24 ± 1.1	99.43 ± 0.9

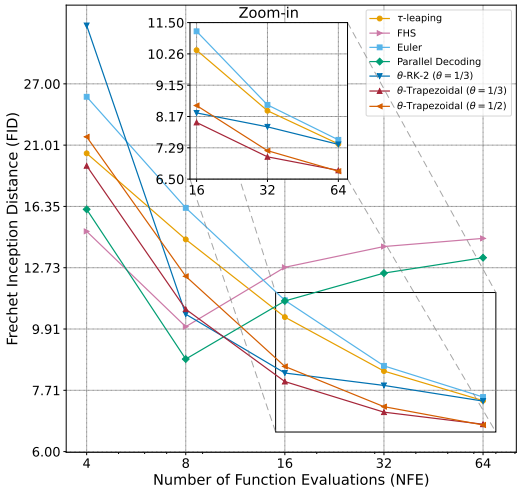

Figure 5: FID of images generated by sampling algorithms vs. number of function evaluations (NFE) with different parameter choices. Lower values are better.

To investigate the robustness of θ-RK-2 with respect to the choice of θ, we also benchmark its performance across multiple choices at NFE 32 and 64, and we present the results in Fig. 6. We observe that the performance of θ-RK-2 has a flat landscape around the optimal θ choices, which fall in the range $[0.15, 0.4]$. In general, as shown by the curve, the method performs better when extrapolation is used to compute the transition rate matrix, confirming the correctness of our theory (Thm. 5.5) and our discussions. Similar to the behavior of θ-Trapezoidal method in Fig. 4, the performance of θ-RK-2 has a flat landscape around the optimal θ choices, which typically falls in the range $[0.3, 0.5]$. In general, as shown by the curve, both methods exhibit better performances when extrapolations are deployed, which, once again, certifies the validity of our theoretical results.

Finally, we visualize some images generated with θ-Trapezoidal on 6 different classes in Fig. 7. θ-Trapezoidal consistently generates high-fidelity images that are visually similar to the ground truth ones and well aligned with the concept.

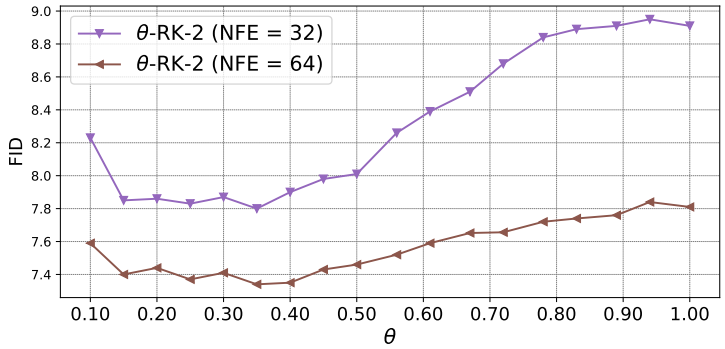

Figure 6: Sampling quality vs. $\theta \in (0, 1]$ in θ-RK-2 algorithm. Sampling quality is quantified through FID.

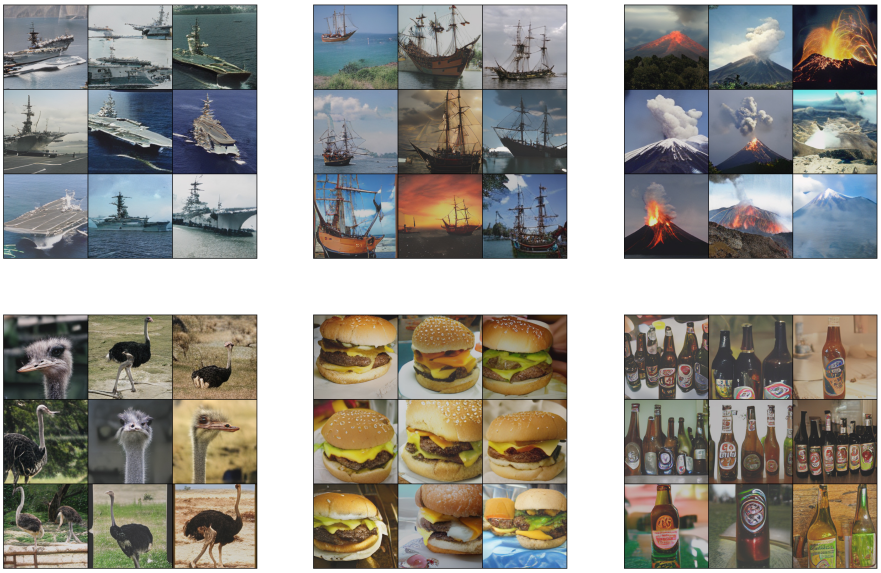

Figure 7: Visualization of samples generated by θ-Trapezoidal. **Upper Left**: Aircraft carrier (ImageNet-1k class: **933**); **Upper Middle**: Pirate (ImageNet-1k class: **724**); **Upper Right**: Volcano (ImageNet-1k class: **980**); **Lower Left**: Ostrich (ImageNet-1k class: **009**); **Lower Middle**: Cheeseburger (ImageNet-1k class: **933**); **Lower Right**: Beer bottle (ImageNet-1k class: **440**).

D.4 Diffusion Large Language Model and Math Reasoning

For the evaluation of diffusion LLMs on math reasoning datasets, we use the checkpoint of LLaDA [64] after instruction tuning, LLaDA-Instruct 8B[5], one of the state-of-the-art diffusion-based LLMs trained natively based on masked discrete diffusion models [33]. For the math reasoning datasets, we consider GSM8K,[6] a standard benchmark for math reasoning consisting of elementary-school-level word problems. Similar to RADD, since LLaDA is also a masked discrete diffusion model, we can freely choose the noise schedule $\sigma(t)$ used in the inference process. We adopt the same log-linear schedule described in (D.1), and we choose $\epsilon = 10^{-3}$. The computation of the score function using LLaDA follows the same procedure as is depicted in App. D.2.

[5] `https://huggingface.co/GSAI-ML/LLaDA-8B-Instruct`

[6] `https://huggingface.co/datasets/openai/gsm8k/viewer/main/train?row=7294`

We choose an early stopping time $\delta = 10^{-3}$, and adopt a uniform discretization of the time interval $(\delta, 1]$ for θ-Trapezoidal, with $\theta = \frac{1}{2}$. For Semi-AR with random remasking or confidence-based remasking strategies, we set the block length to 256, the same as the generation length, to avoid exploiting LLaDA's inherent preference for auto-regressive generation order and ensure a fair comparison with our proposed method. However, we note that our proposed method can also be generalized to a blockwise sampling setting, and we include additional results here for a proof of concept and to demonstrate that LLaDA indeed prefers a left-to-right generation order. We set the sampling temperature to 0.5 for all evaluated inference methods. The evaluation of generated responses follows a standard pipeline with lm-evaluation harness,[7] a standard LLM evaluation kit, following the instruction in the LLaDA repository.[8] Full results are presented in Tab. 7. All experiments are run on 1 NVIDIA A100.

Table 7: Response accuracy on GSM8K with different NFEs.

Accuracy (%)	NFE $= 64$	NFE $= 128$	NFE $= 256$
Semi-AR (Conf.)	33.6	32.0	39.1
Semi-AR (Rand.)	33.8	34.3	40.3
θ-Trapezoidal	35.1	38.4	39.7
θ-Trapezoidal (Blocksize 8)	50.4	57.4	62.5

[7] `https://github.com/EleutherAI/lm-evaluation-harness`

[8] `https://github.com/ML-GSAI/LLaDA`

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work performed by the authors.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper provides the full set of assumptions and a complete (and correct) proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper opens access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper reports error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no such risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets, used in the paper, are properly credited and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets introduced in the paper are well documented and the documentation is provided alongside the assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.