FAST SOLVERS FOR DISCRETE DIFFUSION MODELS: THEORY AND APPLICATIONS OF HIGH-ORDER ALGORITHMS

Yinuo Ren[*]
ICME
Stanford University
yinuoren@stanford.edu

Haoxuan Chen[*†]
ICME
Stanford University
haoxuanc@stanford.edu

Yuchen Zhu[*]
Machine Learning Center
School of Mathematics
Georgia Institute of Technology
yzhu738@gatech.edu

Wei Guo[*]
Machine Learning Center
School of Aerospace Engineering
Georgia Institute of Technology
wei.guo@gatech.edu

Yongxin Chen
School of Aerospace Engineering
Machine Learning Center
Georgia Institute of Technology
yongchen@gatech.edu

Grant M. Rotskoff
Department of Chemistry
ICME
Stanford University
rotskoff@stanford.edu

Molei Tao
School of Mathematics
Machine Learning Center
Georgia Institute of Technology
mtao@gatech.edu

Lexing Ying
Department of Mathematics
ICME
Stanford University
lexing@stanford.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 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.

1 INTRODUCTION

Diffusion and flow-based models on discrete spaces (Chen et al., 2022; Austin et al., 2021; Dieleman et al., 2022; Floto et al., 2023; Hoogeboom et al., 2022; 2021; Meng et al., 2022; Richemond et al.,

[*]Equal Contribution
[†]Corresponding author

2022; Sun et al., 2023; Santos et al., 2023) 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 (Kerby & Moon, 2024), proteins (Frey et al., 2023), and DNA sequences (Avdeyev et al., 2023; Guo et al., 2024b) under biophysical constraints, to the generation of high-fidelity text (Dat et al., 2024) and images (Hu et al., 2022) via autoregressive or masked transitions, *etc.*. Beyond standalone tasks, discrete diffusion models also synergize with methodologies, ranging from tensor networks (Causer et al., 2024) to guidance mechanisms Nisonoff et al. (2024); Li et al. (2024b); Schiff et al. (2024).

Discrete diffusion models, despite their broad applicability, face a critical bottleneck: *inference inefficiency*. Current inference methods include: (1) exact simulation methods (Zheng et al., 2024), 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 methods such as τ-leaping (Campbell et al., 2022), 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 aim to develop high-order numerical schemes tailored for discrete diffusion model inference. Drawing inspirations from acceleration techniques developed for ordinary differential equations (ODEs) (Butcher, 1987), stochastic differential equations (SDEs) (Burrage & Burrage, 1996; Anderson & Mattingly, 2011), chemical reaction simulations (Hu et al., 2011a), and most recently continuous diffusion models (Tachibana et al., 2021; Lu et al., 2022a;b), 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 an unprecedented efficient and versatile solution for discrete diffusion model inference. Below we summarize the main contributions of this paper:

- 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 θ-trapezoidal method and conditional second-order convergence of 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

We briefly review related works here and defer a more detailed discussion to appendix A and B.

Discrete Diffusion Models. Since its introduction, discrete diffusion models have undergone significant refinements, including the development of score-entropy loss (Lou et al., 2024) and flow-matching formulation (Campbell et al., 2024; Gat et al., 2024). These models generally fall into two categories based on their noise distribution: uniform (Lou et al., 2024; Schiff et al., 2024) and masked (absorbing state) (Ou et al., 2024; Shi et al., 2024a; Sahoo et al., 2024; Zheng et al., 2024), each offering unique advantages in modeling discrete distributions. Recent theoretical advances have emerged through numerous studies (Chen & Ying, 2024; Zhang et al., 2024; Ren et al., 2024).

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 Butcher (1987); Kloeden & Platen (1992); Kloeden et al. (2012). These schemes have recently been adapted to accelerate continuous diffusion model inference, encompassing approaches such as the exponential integrators (Zhang & Chen, 2023; Zhang et al., 2023c; Gonzalez et al., 2024), Adams-Bashforth methods (Lu et al., 2022b; Xue et al., 2024; Zhang et al., 2023b), Taylor methods (Tachibana et al., 2021; Dockhorn et al., 2022) and (stochastic) Runge-Kutta methods (Liu et al., 2022a; Lu et al., 2022a; Karras et al., 2022; Zheng et al., 2023b; Li et al., 2024a; Wu et al., 2024a).

High-Order Scheme for Chemical Reaction Systems. Regarding approximate methods developed for simulating compound Poisson processes and chemical reaction systems with state-dependent intensities, efforts have been made on the τ-leaping method (Gillespie, 2001), and its

extensions Cao et al. (2004); Burrage & Tian (2004); Hu et al. (2011a); Hu & Li (2009). For a quick review of the problem setting and these methods, one may refer to E et al. (2021). The adaption 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) $(\boldsymbol{x}_t)_{0\le t\le T}$ 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:

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

is the rate matrix at time t satisfying

(i) For any $x \in \mathbb{X}$, $Q_t(x,x) = -\sum_{y\neq x} Q_t(y,x)$;

(ii) For any $x \neq y \in \mathbb{X}$, $Q_t(x,y) \ge 0$.

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 (Kelly, 2011):

$$\frac{\mathrm{d}\overleftarrow{\boldsymbol{p}}_s}{\mathrm{d}s} = \overline{\boldsymbol{Q}}_s \overleftarrow{\boldsymbol{p}}_s, \tag{2.2}$$

where $\overleftarrow{*}_s$ denotes $*_{T-s}$ and the rate matrix is defined by

$$\overline{Q}_s(y,x) = \begin{cases} \frac{\overleftarrow{p}_s(y)}{\overleftarrow{p}_s(x)} \overleftarrow{Q}_s(x,y), & x \neq y \in \mathbb{X}, \\ -\sum_{y'\neq x} \overline{Q}_s(y',x), & x = y \in \mathbb{X}. \end{cases}$$

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 (Lou et al., 2024), 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 rather 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 (Lou et al., 2024; Benton et al., 2024b) for some weights $\psi_t \ge 0$ as follows:

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

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 $\widehat{\overline{Q}}_s^{\phi}(y,x) = \widehat{\overleftarrow{s}}_s{}^{\phi}(x,y) \overleftarrow{Q}_s(x,y)$ 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

According to Ren et al. (2024), discrete diffusion models can also be formulated as stochastic integrals, which is especially useful for their theoretical analysis. In this section, we briefly recapitulate relevant results therein and refer to appendix C for 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

$\mathbb{D} := \{x - y : x \neq y \in \mathbb{X}\}$. We first introduce the Poisson random measure with evolving intensities, 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 in discrete diffusion models (2.1) can thus be represented by the following stochastic integral:

$$x_t = x_0 + \int_0^t \int_{\mathbb{D}} \nu N[\lambda](\mathrm{d}t, \mathrm{d}\nu), \tag{2.4}$$

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:

$$y_s = y_0 + \int_0^s \int_{\mathbb{D}} \nu N[\mu](\mathrm{d}s, \mathrm{d}\nu), \tag{2.5}$$

where the intensity μ is defined as

$$\mu_s(\nu, \omega) = \overleftarrow{s}_s(y_{s^-}, y_{s^-} + \nu)\overleftarrow{Q}_s^0(y_{s^-}, y_{s^-} + \nu) \tag{2.6}$$

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 (2.5), 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 (2.6). . In the following, we will 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 $y_{s^-}(\omega)$.

3 Algorithm

In this section, we present the high-order solvers proposed for simulating discrete diffusion models and their associated stochastic integral formulations. We will primarily focus on two-stage algorithms aiming for second-order accuracy. Specifically, we will introduce the θ-RK-2 method and the θ-Trapezoidal method. Throughout this section, we assume a time discretization scheme $(s_i)_{i\in[0:N]}$ with

$$0 = s_0 < s_1 < \cdots < s_N = T - \delta,$$

where δ is the early stopping time. We will also 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 $\mathbb{D}$.

3.1 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 (B.3) 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 (Burrage & Burrage, 1996) and the Poisson (Burrage & Tian, 2004) RK methods. A summary of this method is given in algorithm 1.

Intuitively, the θ-RK-2 method is a two-stage algorithm that firstly 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. Then it runs another step of τ-leaping for a full step Δ_n using a weighted sum of the intensities at the current time point s_n and the θ-section point ρ_n. In order to rigorously analyze and better illustrate the θ-RK-2 method, we need the following definition:

Algorithm 1: θ-RK-2 Method for Discrete Diffusion Model Inference

Input: $\widehat{y}_0 \sim q_0$, $\theta \in [\frac{1}{2}, 1]$, time discretization $(s_n, \rho_n)_{n\in[0:N-1]}$, $\widehat{\mu}$, $\widehat{\mu}^*$ as defined in proposition 3.2.

Output: A sample $\widehat{y}_{s_N} \sim \widehat{q}^{\mathrm{RK}}_{t_N}$.

for $n = 0$ **to** $N-1$ **do**

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

$\quad \widehat{y}_{s_{n+1}} \leftarrow \widehat{y}_{s_n} + \sum_{\nu\in\mathbb{D}} \nu\mathcal{P}\left(\left(\left(1-\frac{1}{2\theta}\right)\widehat{\mu}_{s_n} + \frac{1}{2\theta}\widehat{\mu}^*_{\rho_n}\right)(\nu)\Delta_n\right)$;

end

Definition 3.1 (Intermediate Process)**.** *We define the intermediate process $\widehat{y}^*_s$ piecewisely on $(s_n, s_{n+1}]$ as follows*

$$\widehat{y}^*_s = \widehat{y}_{s_n} + \int_{s_n}^{s}\int_{\mathbb{D}} \nu N\left[\widehat{\mu}_{s_n}\right](\mathrm{d}s, \mathrm{d}\nu), \tag{3.1}$$

where the intensity $\widehat{\mu}_{s_n}$ is given by

$$\widehat{\mu}_{s_n}(\nu, \widehat{y}_{s_n}) = \overleftarrow{\widehat{s}}_{s_n}(\widehat{y}_{s_n}, \widehat{y}_{s_n}+\nu)\overleftarrow{\widehat{Q}}^0_{s_n}(\widehat{y}_{s_n}, \widehat{y}_{s_n}+\nu). \tag{3.2}$$

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. See appendix D.1 for the proof.

Proposition 3.2 (Stochastic Integral Formulation of θ-RK-2 Method)**.** *The θ-RK-2 method (algorithm 1) is equivalent to solving the following stochastic integral:*

$$\widehat{y}^{\mathrm{RK}}_s = \widehat{y}^{\mathrm{RK}}_0 + \int_0^s\int_{\mathbb{D}} \nu N\left[\widehat{\mu}^{\mathrm{RK}}\right](\mathrm{d}s, \mathrm{d}\nu), \tag{3.3}$$

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

$$\widehat{\mu}^{\mathrm{RK}}_s(\nu) = (1-\tfrac{1}{2\theta})\widehat{\mu}_{\lfloor s\rfloor}(\nu, \widehat{y}^{\mathrm{RK}}_{\lfloor s\rfloor}) + \tfrac{1}{2\theta}\widehat{\mu}^*_{\rho_s}(\nu, \widehat{y}^*_{\rho_s}), \tag{3.4}$$

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

$$\widehat{\mu}^*_s(\nu, \widehat{y}^*_s) = \overleftarrow{\widehat{s}}_s(\widehat{y}^*_s, \widehat{y}^*_s+\nu)\overleftarrow{\widehat{Q}}^0_s(\widehat{y}^*_s, \widehat{y}^*_s+\nu), \tag{3.5}$$

with the intermediate process $\widehat{y}^*_s$ defined in (3.1) for the corresponding interval. We will call the process $\widehat{y}^{\mathrm{RK}}_s$ the interpolating process *of the θ-RK-2 method and denote the distribution of $\widehat{y}^{\mathrm{RK}}_s$ by $\widehat{q}^{\mathrm{RK}}_s$.*

We emphasize that our method is different from the midpoint method proposed in Gillespie (2001) for simulating chemical reactions, where the Poisson random variable in the first step is replaced by its expected magnitude. We remark that such modification is in light of the lack of continuity and orderliness of the state space.

3.2 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 (Anderson & Mattingly, 2011) and chemical reactions (Hu et al., 2011a). 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 algorithm 2. Intuitively, this method 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 two different intensities designed in a balanced way.

Algorithm 2: θ-Trapezoidal 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 proposition 3.3.

Output: A sample $\widehat{y}_{s_N} \sim \widehat{q}^{\text{trap}}_{t_N}$.

for $n = 0$ **to** $N-1$ **do**

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

$\quad \widehat{y}_{s_{n+1}} \leftarrow \widehat{y}^*_{\rho_n} + \sum_{\nu\in\mathbb{D}} \nu\mathcal{P}\left(\left(\alpha_1\widehat{\mu}^*_{\rho_n} - \alpha_2\widehat{\mu}_{s_n}\right)_+(\nu)(1-\theta)\Delta_n\right)$;

end

Compared to the θ-RK-2 method, the θ-trapezoidal method is also a two-stage algorithm 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 $(\alpha_1, -\alpha_2)$, which performs 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 an unconditionally high-order scheme effectively.

The following proposition establishes the stochastic integral formulation of the θ-trapezoidal method, whose proof can be found in appendix D.1.

Proposition 3.3 (Stochastic Integral Formulation of θ-Trapezoidal Method)**.** *The θ-trapezoidal method (algorithm 2) is equivalent to solving the following stochastic integral:*

$$\widehat{y}^{\text{trap}}_s = \widehat{y}^{\text{trap}}_0 + \int_0^s \int_{\mathbb{D}} N[\widehat{\mu}^{\text{trap}}](\mathrm{d}s, \mathrm{d}\nu) \tag{3.6}$$

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

$$\widehat{\mu}^{\text{trap}}_s(\nu) = \mathbf{1}_{s<\rho_s}\widehat{\mu}_{\lfloor s\rfloor}(\nu, \widehat{y}^{\text{trap}}_{\lfloor s\rfloor}) + \mathbf{1}_{s\geq\rho_s}\left(\alpha_1\widehat{\mu}^*_{\rho_s}(\nu, \widehat{y}^*_{\rho_s}) - \alpha_2\widehat{\mu}_{\lfloor s\rfloor}(\nu, \widehat{y}^{\text{trap}}_{\lfloor s\rfloor})\right)_+. \tag{3.7}$$

Above, $\mathbf{1}_{(\cdot)}$ denotes the indicator function and the intermediate process $\widehat{y}^*_s$ is defined in (3.1) for the corresponding interval. We will call the process $\widehat{y}^{\text{trap}}_s$ the interpolating process *of the θ-trapezoidal method and denote the distribution of $\widehat{y}^{\text{trap}}_s$ by $\widehat{q}^{\text{trap}}_s$.*

4 Theoretical Analysis

In this section, we provide the theoretical results of the θ-trapezoidal and θ-RK-2 methods. We will first present the assumptions and guarantees of the algorithms. Then we will present the error bounds of the algorithms and discuss the implications of the results. We note that assumptions used here are all deferred to appendix D.2. The following theorem summarizes our theoretical guarantees for the θ-trapezoidal method:

Theorem 4.1 (Second Order Convergence of θ-Trapezoidal Method)**.** *Suppose $\theta \in (0,1]$ and $\alpha_1\widehat{\mu}^*_{\rho_s} - \alpha_2\widehat{\mu}_{\lfloor s\rfloor} \geq 0$ in (3.7) for all $s \in [0, T-\delta]$, then the following error bound holds under assumptions D.1, D.2 and D.3:*

$$D_{\text{KL}}(p_\delta \| \widehat{q}^{\text{trap}}_{T-\delta}) \lesssim \exp(-T) + (\epsilon_{\text{I}} + \epsilon_{\text{II}})T + \kappa^2 T, \tag{4.1}$$

where δ is the early stopping time, $\kappa = \max_{n\in[0:N-1]}\Delta_n$, i.e.*, the largest stepsize, and T is the time horizon.*

The complete proof is presented in appendix D.4. The outline is to first bound $D_{\text{KL}}(p_\delta \| \widehat{q}^{\text{trap}}_{T-\delta})$ by the KL divergence between the corresponding path measures, as established in theorem D.4, and then decompose the integral in the log-likelihood and bound respectively, where the primary technique used is *Dynkin's formula* (theorem D.9). With a term-by-term comparison with theorem B.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 sufficient time horizon T and accurate score estimation, with empirical validation presented in section 5. However, within the scope of our analysis, the θ-RK-2 method may not possess a theoretical guarantee as extensive as the θ-trapezoidal method for all $\theta \in (0, 1]$. We briefly summarize our understanding as follows.

Theorem 4.2 (Conditional Second-Order Convergence of θ-RK-2 Method)**.** *Suppose $\theta \in (0, \frac{1}{2}]$ and $(1 - \frac{1}{2\theta})\widehat{\mu}_{\lfloor s \rfloor} + \frac{1}{2\theta}\widehat{\mu}^*_{\rho_s} \geq 0$ in (3.4) for all $s \in [0, T - \delta]$, then the following error bound holds for the practical version (algorithm 4) of algorithm 1 under assumptions D.1, D.2 and D.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 T is the time horizon.*

The proof of the theorem above is provided in appendix D.5. The restricted range of θ is caused by one specific error term $(\mathrm{III}.4)$ (D.3) that permits bounding with *Jensen's inequality* only when $\theta \in (0, \frac{1}{2}]$, similar to its counterpart $(\mathrm{II}.4)$ (D.5) 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. 3 of appendix E.4, where the performance of θ-RK-2 method clearly peaks when $\theta \in (0, \frac{1}{2}]$.

Remark 4.3 (Comparison between Trapezoidal and RK-2 Methods)**.** *Trapezoidal-type methods were originally proposed by Anderson & Mattingly (2011) as a minimal second-order scheme in the weak sense for simulating SDEs. In simulating chemical reaction contexts, Hu et al. (2011a) claimed that trapezoidal-type 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, though we defer theoretical investigation of covariance error convergence in discrete diffusion models to future work.*

5 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 amount of function evaluations. This section empirically validates the anticipated effectiveness of our proposed θ-trapezoidal method (algorithm 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 (Ou et al., 2024), τ-leaping (Campbell et al., 2022), Tweedie τ-leaping (Lou et al., 2024), and Parallel Decoding (Chang et al., 2022). We conduct evaluations on both uniform and masked discrete diffusion models, with detailed experimental protocols provided in appendix E.

5.1 15-Dimensional Toy Model

We first evaluate the performance of the θ-trapezoidal method using a 15-dimensional toy model. 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 matrix 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 method 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 $\widehat{\boldsymbol{q}}_T$ with bootstrap confidence intervals. 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 w.r.t. the step count, confirming our theories.

5.2 Text Generation

For the text generation task, we employ the pre-trained score function from RADD (Ou et al., 2024) as our base model for benchmarking inference algorithms. RADD is a masked discrete diffusion

model with GPT-2-level text generation capabilities (Radford et al., 2019) and is trained on the OpenWebText dataset (Gokaslan & Cohen, 2019). 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 the Euler method, τ-leaping, Tweedie τ-leaping, and our proposed θ-trapezoidal method. We generate text sequences of 1024 tokens and measure their generative perplexity following the evaluation protocol established in Ou et al. (2024). Table table 1 presents a comprehensive list of results for a wide range of NFE values. which demonstrate that the θ-trapezoidal method consistently produces better samples under a fixed computation budget compared with 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. These results validate the practical efficiency and accuracy of algorithm 2.

5.3 Image Generation

Our experiments on the image generation task utilize the pre-trained score function from MaskGIT (Chang et al., 2022; Besnier & Chen, 2023) as the base model, which can be converted into a masked discrete diffusion model by introducing a noise schedule (see appendix E.4). MaskGIT employs a masked image transformer architecture trained on ImageNet (Deng et al., 2009) of 256×256 resolution, where each image amounts to a sequence of 256 discrete image tokens following VQ-GAN tokenization (Esser et al., 2021b). We evaluate the θ-trapezoidal method against the Euler method, τ-leaping, and parallel decoding under equivalent NFE budgets ranging from 4 to 64. For each, we generate 5×10^4 images and compute their Fréchet Inception Distance (FID) against the ImageNet validation split, following the setting in Chang et al. (2022). Figure 5 reveals that θ-trapezoidal method (algorithm 2) consistently achieves lower (and thus better) FID values compared to both Euler method and τ-leaping across all NFE values. While parallel decoding shows advantages at extremely low NFE (≤ 8), its performance saturates with increased computational resources, making it less favorable compared to our rapidly converging method. Additional results are detailed in appendix E.

6 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 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.

Acknowledgments

YC is supported by the National Science Foundation under Award No. DMS-2206576. GMR is supported by a Google Research Scholar Award. MT is grateful for partial support by the National Science Foundation under Award No. DMS-1847802. LY acknowledges the support of the National Science Foundation under Award No. DMS-2208163.

References

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

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*, pp. 2023–09, 2023.

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

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.

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

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.

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.

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.

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.

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.

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.

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*, pp. 1276–1301. PMLR, 2023.

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.

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

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*, 2024a. URL `https://openreview.net/forum?id=r5njV3BsuD`.

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, 2024b.

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

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*, pp. 170–188. Springer, 2022.

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.

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.

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

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.

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.

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.

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, 2004a.

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, 2004b.

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

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.

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.

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*, pp. 2824–2834, 2024.

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

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.

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, 2005a.

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

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

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.

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.

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*, pp. 11315–11325, 2022.

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.

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.

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*, 2024a.

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

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, 2024b.

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

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*, 2024c.

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.

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*, pp. 6059–6069, 2023.

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.

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.

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

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

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.

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

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

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.

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.

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.

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

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.

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, 2021a.

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*, pp. 12873–12883, 2021b.

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

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

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.

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.

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.

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.

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.

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

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

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.

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

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.

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.

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.

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*, pp. 10696–10706, 2022.

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, 2024a.

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

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

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.

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.

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

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.

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

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

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.

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.

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*, pp. 11502–11511, 2022.

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

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), 2011a.

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

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

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*, pp. 10167–10176, 2023.

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.

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

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.

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

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

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.

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

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

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.

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*, 2024a.

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.

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

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*, 2024b.

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.

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, 2024c.

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.

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

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

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.

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*, 2022b.

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.

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.

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

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

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.

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

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*, pp. 15762–15772, 2024.

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.

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*, pp. 14297–14306, 2023.

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

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

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

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

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.

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.

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.

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*, pp. 4474–4484. PMLR, 2020.

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

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.

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

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.

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.

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

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

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

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.

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.

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.

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

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.

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

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

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

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.

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

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.

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

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*, pp. 9034–9059. PMLR, 2023.

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.

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

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.

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

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

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.

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*, 2024a.

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*, 2024b.

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

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

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

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

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

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.

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

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.

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.

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

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.

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.

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

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.

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

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.

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.

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.

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.

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

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)*, pp. 11336–11340. IEEE, 2024b.

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.

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.

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

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.

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, 2023a.

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

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.

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

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*, pp. 7226–7236, 2023a.

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

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.

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

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

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

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

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, 2024b.

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

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*, pp. 42390–42402. PMLR, 2023a.

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, 2023b.

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.

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

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.

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*, pp. 7777–7786, 2024.

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.

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.

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, that are highly related to our work.

Discrete Diffusion Models: Methodology, Theory, and Applications. Discrete diffusion and flow-based models (Chen et al., 2022; Austin et al., 2021; Floto et al., 2023; Hoogeboom et al., 2021; Meng et al., 2022; Richemond et al., 2022; Campbell et al., 2022; Sun et al., 2023; Santos et al., 2023) 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 (Igashov et al., 2023), solving inverse problems (Murata et al., 2024), combinatorial optimization (Li et al., 2024c; Sun & Yang, 2023), designing molecules, proteins, and DNA sequences (Alamdari et al., 2023; Avdeyev et al., 2023; Emami et al., 2023; Frey et al., 2023; Penzar et al., 2023; Watson et al., 2023; Yang et al., 2023b; Campbell et al., 2024; Stark et al., 2024; Kerby & Moon, 2024; Yi et al., 2024; Zhu et al., 2024), image synthesis (Esser et al., 2021a; Lezama et al., 2022; Gu et al., 2022), text summarization (Dat et al., 2024), as well as the generation of graph (Seff et al., 2019; Niu et al., 2020; Shi et al., 2020; Qin et al., 2023; Vignac et al., 2022; Haefeli et al., 2022; Qin et al., 2024; Kim et al., 2024), layout (Inoue et al., 2023; Zhang et al., 2023a), motion (Chi et al., 2024; Lou et al., 2023), sound (Campbell et al., 2022; Yang et al., 2023a), image (Hu et al., 2022; Bond-Taylor et al., 2022; Tang et al., 2022; Zhu et al., 2022), speech (Wu et al., 2024b), electronic health record (Han et al., 2024), tabular data (Shi et al., 2024b) and text (He et al., 2022; Savinov et al., 2021; Wu et al., 2023; Gong et al., 2023; Zheng et al., 2023c; Zhou et al., 2023; Shi et al., 2024a; Sahoo et al., 2024; Xu et al., 2024; Guo et al., 2024b). 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 Chen & Ying (2024); Zhang et al. (2024); Ren et al. (2024).

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 Park et al. (2024), developing fast samplers (Chen et al., 2024c), designing correctors based on information learnt by the model (Zhao et al., 2024c), simplifying the loss function for training Zhao et al. (2024a), adding editing-based refinements (Reid et al., 2023), synergizing these models with other techniques and methodologies like distillation Hayakawa et al. (2024), Ehrenfest processes (Winkler et al., 2024), Glauber dynamics (Varma et al., 2024), tensor networks (Causer et al., 2024), enhanced guidance mechanisms (Gruver et al., 2024; Nisonoff et al., 2024; Li et al., 2024b; Schiff et al., 2024), structured preferential generation (Rissanen et al., 2024), the plan-and-denoise framework (Liu et al., 2024) and alternative metrics, *e.g.*, the Fisher information metric (Davis et al., 2024). 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 worthwhile 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 (Mil'shtejn, 1975), explicit methods (Abdulle & Cirilli, 2008), multistep methods (Buckwar & Winkler, 2006), extrapolation-type methods (Talay & Tubaro, 1990; Anderson & Mattingly, 2011), stochastic Runge Kutta methods (Burrage & Burrage, 1996; 2000; Burrage & Tian, 2002; Rössler, 2003; Rößler, 2010), splitting methods (Foster et al., 2024), methods based on gaussian mixtures (Li et al., 2021), randomized midpoint method Shen & Lee (2019), parallel sampling methods Anari et al. (2024); Yu & Dalalyana (2024) as well as high-order methods for stochastic gradient Markov Chain Monte Carlo Chen et al. (2015); Durmus et al. (2016), underdamped and overdamped Langevin Monte Carlo Li et al. (2019); Sabanis & Zhang (2019); Mou et al. (2021); Monmarché (2021); Foster et al. (2021). For a more comprehensive list

of related numerical methods, one may refer to (Kloeden & Platen, 1992; Burrage et al., 2004a; Milstein & Tretyakov, 2004; Kloeden et al., 2012; E et al., 2021).

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 Bortz et al. (1975) 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) (Gillespie, 1976; 1977) and its variants for multiscale modeling (Cao et al., 2005a;c; E et al., 2005; 2007), the next reaction method and its variants Gibson & Bruck (2000); Anderson (2007), uniformization-based methods Beentjes & Baker (2019), etc. The second class of methods are approximate simulation methods, including but not limited to the τ-leaping method (Gillespie, 2001) and its variants Rathinam et al. (2003); Gillespie & Petzold (2003); Cao et al. (2004); Burrage & Tian (2004); Burrage et al. (2004b); Cao et al. (2005b); Auger et al. (2006); Cao et al. (2007); Bayati et al. (2009); Cao & Petzold (2008); Xu & Cai (2008); Hu & Li (2009); Hu et al. (2011a); Anderson & Higham (2012); Moraes et al. (2014); Padgett & Ilie (2016); Lipková et al. (2019). For a subset of the methods listed above, numerical analysis has also been performed in many works Rathinam et al. (2005); Li (2007); Hu et al. (2011b); Anderson et al. (2014); Chen & Liu (2017) to justify their validity.

Continuous Diffusion Models: Methodology, Theory, and Acceleration. Continuous diffusion and probability flow-based models (Sohl-Dickstein et al., 2015; Zhang et al., 2018; Song & Ermon, 2019; Ho et al., 2020; Song et al., 2020; 2021; Lipman et al., 2022; Liu et al., 2022b; Albergo & Vanden-Eijnden, 2022; Albergo et al., 2023) 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 (Chen et al., 2024a; Ren et al., 2024). 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 (Zhou et al., 2024), restart sampling Xu et al. (2023), gaussian mixture solvers Guo et al. (2024a), self-consistency Heek et al. (2024); Song et al. (2023); Song & Dhariwal (2023); Lu & Song (2024), knowledge distillation Luhman & Luhman (2021); Meng et al. (2023); Salimans & Ho (2022), combination with underdamped Langevin dynamics Dockhorn et al. (2021), operator learning Zheng et al. (2023a) and more recently ideas from accelerating large language models (LLMs) like caching (Ma et al., 2024) and speculative decoding De Bortoli et al. (2025). Among all the proposed accelerating methods, one major class of methods are developed based on techniques from numerical analysis like adaptive step sizes Jolicoeur-Martineau et al. (2021), exponential integrators (Zhang & Chen, 2023; Zhang et al., 2023c), predictor-corrector solver Zhao et al. (2024b), Adams-Bashforth methods (Lu et al., 2022b; Xue et al., 2024; Zhang et al., 2023b), Taylor methods (Tachibana et al., 2021; Dockhorn et al., 2022), Picard iteration and parallel sampling (Shih et al., 2024; Chung et al., 2023; Tang et al., 2024; Cao et al., 2024; Selvam et al., 2024; Chen et al., 2024a), (stochastic) Runge-Kutta methods (Liu et al., 2022a; Lu et al., 2022a; Karras et al., 2022; Zheng et al., 2023b; Li et al., 2024a; Wu et al., 2024a) and randomized midpoint method (Kandasamy & Nagaraj, 2024; Gupta et al., 2024). In contrast, there has been much fewer studies on the acceleration of discrete diffusion models via techniques from numerical analysis, which inspires the study undertaken in this paper.

B Numerical Schemes for Discrete Diffusion Model Inference

In this section, we discuss existing numerical schemes for discrete diffusion models, including exact simulation methods and the τ-leaping method.

B.1 Exact Simulation Methods

Unlike continuous diffusion models, where exact simulation is beyond reach, discrete diffusion models permit inference without discretization error. Notable examples of unbiased samplers include the uniformization algorithm (Chen & Ying, 2024) for the uniform state case and the First-Hitting Sampler (FHS) (Zheng et al., 2024) 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 Ren et al. (2024) re-

veals 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 $\Omega(d)$, where d is the data dimension. 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 significant concentration occurring during the terminal phase of the backward process, precisely 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 true score function, making accurate neural network estimation particularly challenging during the terminal phase of the backward process (*cf.* Assumption 4.4 in Ren et al. (2024)).

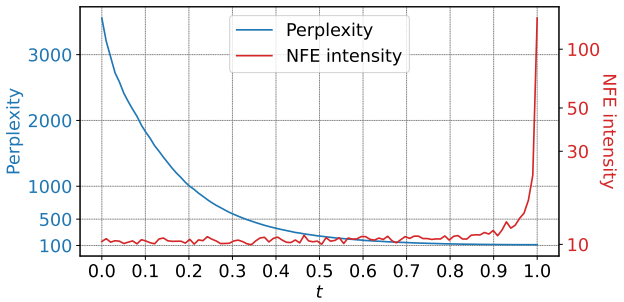

Figure 1: An illustrative application of the uniformization algorithm to discrete diffusion models for text generation. The x-axis denotes the time of the backward process, and the y-axis denotes the frequency of jumps reflected by NFE. Perplexity convergence occurs well before the NFE experiences unbounded growth.

Figure 1 illustrates an application of the uniformization algorithm to discrete diffusion model inference for text generation, with detailed experimental parameters presented in section 5.3 and appendix E.4. As the process approaches the target distribution ($t \to T$), the number of required jumps grows unbounded, while perplexity improvements become negligible. This skewed distribution of computational effort results in numerous *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 the parameter selection of δ, 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.

B.2 Approximate Method: τ-Leaping Method

The τ-leaping method (Gillespie, 2001; Campbell et al., 2022) represents 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:

$$\widehat{y}_{t+\delta} = \widehat{y}_t + \sum_{\nu \in \mathbb{D}} \nu \mathcal{P}\left(\widehat{\mu}_t(\nu)\delta\right). \tag{B.1}$$

In general, one may design different discretization schemes for τ-leaping, and the summation in (B.1) is parallelizable, underscoring the method's flexibility and efficiency. We refer to algorithm 3 and appendix C.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 Campbell et al. (2022); Ren et al. (2024) 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 appendix C.2.

Theorem B.1 (Thm. 4.7 in Ren et al. (2024))**.** *For the state space $\mathbb{X} = [S]^d$, with S sites along each dimension, under certain discretization scheme and assumptions and given an ϵ-accurate score*

function, the following error bound holds:

$$D_{\mathrm{KL}}(p_\delta \| \widehat{q}_{T-\delta}) \lesssim \exp(-T) + \epsilon + \kappa T, \tag{B.2}$$

where $\delta \ll 1$ is the early stopping time, κ is the parameter controlling the step size, and T is the time horizon. The notation $\lesssim$ means that the left-hand side is bounded by the right-hand side up to a constant factor as $\kappa \to 0$.

The error bound (B.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 $\mathcal{O}(\kappa T)$.

B.3 Approximate Method: High-Order 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 (Butcher, 1987) 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:

$$\begin{aligned} \widehat{x}^*_{t+\theta\delta} &= \widehat{x}_t + f_t(\widehat{x}_t)\theta\delta, \\ \widehat{x}_{t+\delta} &= \widehat{x}_t + \left[(1-\tfrac{1}{2\theta})f_t(\widehat{x}_t) + \tfrac{1}{2\theta} f_{t+\theta\delta}(\widehat{x}^*_{t+\theta\delta})\right]\delta. \end{aligned} \tag{B.3}$$

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 $f \in C^2(\mathbb{R})$,

$$\left[\left(1-\tfrac{1}{2\theta}\right) f(a) + \tfrac{1}{2\theta} f(a+\theta\delta)\right]\delta$$

offers a second-order approximation of $\int_a^{a+\delta} f(x)\mathrm{d}x$ in contrast to $f(a)\delta$, which is only first-order. This approach has been successfully adapted for SDE simulation (Burrage & Burrage, 1996) and continuous diffusion model inference (Karras et al., 2022; Lu et al., 2022a;b; Zheng et al., 2023b; Wu et al., 2024a). 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.

C 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.

C.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 C.1 (Poisson Random Measure (Ren et al., 2024, 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 C.2 (Poisson Random Measure with Evolving Intensity (Ren et al., 2024, 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 *$\lambda_t(y)$ 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 C.3 (Construction of Poisson Random Measure with Evolving Intensity)**.** *As discussed in Thm. A.4 in Ren et al. (2024) and originally proposed by Protter (1983), 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

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

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 C.4 (Change of Measure for Poisson Random Measure with Evolving Density (Ren et al., 2024, 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:*

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

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 $\lambda_t(\nu)h_t(\nu)$.

C.2 Error Analysis of τ-leaping

The τ-leaping method was originally proposed by Gillespie (2001) and adopted for the inference of discrete diffusion models by Campbell et al. (2022). A summary of the algorithm is given in algorithm 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.

Proof of theorem B.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

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 proposition 3.2.
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
```

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 Ren et al. (2024) with $C = 1$ and $\underline{D} = \overline{D} = 2d$. Assumption 4.3(ii) is also satisfied, as shown in Example B.10 of Ren et al. (2024).

Then we may apply Thm. 4.7 in Ren et al. (2024) 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 eventually by scaling the transition matrix $\boldsymbol{Q}$ by $\frac{1}{d}$, equivalent to scaling the time by d. □

D 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 appendix D.1. Then we will provide the proofs of the main results for the θ-trapezoidal method in appendix D.4 and the θ-RK-2 method in appendix D.5. 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 appendix D.6, we provide the detailed lemmas and computations omitted in the proofs of theorem 4.1 and theorem 4.2.

D.1 STOCHASTIC INTEGRAL FORMULATIONS OF HIGH-ORDER SCHEMES

Proof of proposition 3.2 and proposition 3.3. 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 (3.1) 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. □

D.2 Assumptions

We will primarily consider the following assumptions for the analysis of the θ-trapezoidal and θ-RK-2 methods.

Assumption D.1 (Convergence of Forward Process)**.** *The forward process converges to the stationary distribution exponentially fast,* i.e.*, $D_{\mathrm{KL}}(p_T \| p_\infty) \leq \exp(-\rho T)$, where $\rho > 0$ is the exponential convergence rate.*

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.* Ren et al. (2024)). The exponential rate aligns with continuous diffusion models (*cf.* in Benton et al. (2024a)).

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

This essentially assumes two key requirements of the scores: (1) the forward process evolution maintains sufficient smoothness, which is achievable through appropriate time reparametrization; and (2) 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. This assumption corresponds to Assumps. 4.3(i), 4.4 and 4.5 in Ren et al. (2024).

Assumption D.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 $s \in \cup_{n\in[0:N-1]}\{s_n, \rho_n\}$,*

$$\begin{aligned}&\text{(i) } \mathbb{E}\left[\int_{\mathbb{D}} \left(\mu_s(\nu)\log\tfrac{\mu_s(\nu)}{\widehat{\mu}_s(\nu)} - \mu_s(\nu) + \widehat{\mu}_s(\nu)\right)\gamma(\mathrm{d}\nu)\right] \leq \epsilon_{\mathrm{I}}; \\ &\text{(ii) } \mathbb{E}\left[\int_{\mathbb{D}} |\mu_s(\nu) - \widehat{\mu}_s(\nu)|\,\gamma(\mathrm{d}\nu)\right] \leq \epsilon_{\mathrm{II}}.\end{aligned}$$

This assumption quantifies the proximity of the estimated intensity $\widehat{\mu}$ to the true intensity μ after sufficient training. Compared with Ren et al. (2024), the additional L^∞ part in (ii) is required for technical reasons, which is similar to Chen et al. (2024b); Wu et al. (2024a). In practice, such additional assumptions may be realized by adding extra penalty terms to the objective function during training.

D.3 Convergence Guarantees

D.4 Convergence Analysis of the θ-Trapezoidal Method

Theorem D.4. *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 (2.5) and the interpolating process (3.6) of the θ-trapezoidal method (algorithm 2), then it holds that*

$$\begin{aligned}&D_{\mathrm{KL}}(\overleftarrow{p}_{T-\delta}\|\widehat{q}^{\mathrm{trap}}_{T-\delta}) \leq D_{\mathrm{KL}}(\overleftarrow{p}_{0:T-\delta}\|\widehat{q}^{\mathrm{trap}}_{0:T-\delta}) \\ \leq &D_{\mathrm{KL}}(\overleftarrow{p}_0\|\widehat{q}_0) + \mathbb{E}\left[\int_0^{T-\delta}\int_{\mathbb{D}}\left(\mu_s(\nu)\log\frac{\mu_s(\nu)}{\widehat{\mu}^{\mathrm{trap}}_s(\nu)} - \mu_s(\nu) + \widehat{\mu}^{\mathrm{trap}}_s(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right],\end{aligned} \tag{D.1}$$

where the intensity $\widehat{\mu}^{\mathrm{trap}}$ is defined in (3.6), and the expectation is taken w.r.t. both paths generated by the backward process (2.5) 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 (3.1), 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 (3.1) and the corresponding intensity $\widehat{\mu}_s^*$ is defined in (3.5).

Given the stochastic integral formulation of the backward process (2.5) and the interpolating process of the θ-trapezoidal method (3.6), we have by theorem C.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

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

i.e.,

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

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

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

and the proof is complete. □

In the following, we will provide the outline of the proof of theorem 4.1, where we leave the proof of several lemmas and detailed calculations to appendix D.6 for the clarity of presentation.

Proof of theorem 4.1. Throughout this proof, including the subsequent lemmas and propositions that will be detailed in appendix D.6, 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 (2.5) and set it as the underlying paths of the expectation in (D.1) as required by theorem D.4. 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 (3.1) as y_s^*, and the corresponding intermediate intensity as appeared in (3.5) 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 (3.7)

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

we can rewrite the corresponding part of the integral in (D.1) as

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

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. We note that such positivity assumption also exists in the analysis performed by Anderson & Mattingly (2011) and Hu et al. (2011a) and a more detailed discussion on such assumption is deferred to remark D.5.

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

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

where each term is defined as

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

(ii) The second term is decomposed as

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

where each term is defined as

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

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

Bounding the Error Terms. Then we briefly summarize the intuitions and related techniques used in the bound of the terms above, and the detailed calculations and proofs of the lemmas and propositions are deferred to appendix D.6.

(i) *Error due to estimation error associated with the intensity:* The terms $(\mathrm{I}.1)$ and $(\mathrm{II}.1)$ are bounded by the assumption on the estimation error of the intensity $\widehat{\mu}_s$ (assumption D.3), as

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

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

The term $(\mathrm{II}.4)$ is bounded by proposition D.8, as

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

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

(ii) *Error related to the smoothness of intensity:* By corollary D.12, the terms $(\mathrm{I}.2)$ and $(\mathrm{II}.2)$ are bounded by

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

By corollary D.13, the terms $(\mathrm{I}.4)$ and $(\mathrm{II}.6)$ are bounded by

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

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 (assumption D.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 (theorem D.9). The third-order accuracy here directly follows from the intuition provided in appendix B.3 based on numerical quadrature.

(iii) *Error involving the intermediate process:* The terms $(\mathrm{II}.3)$ and $(\mathrm{II}.5)$ are bounded by proposition D.17 and corollary D.18 respectively as follows

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

The term $(\mathrm{I}.3)$ is bounded by proposition D.19 as below

$$
\mathbb{E}\left[(\mathrm{I}.3)\right] \lesssim \Delta_n^3 \epsilon_{\mathrm{II}} + \Delta_n^4 \lesssim \Delta_n^2 \epsilon_{\mathrm{II}} + \Delta_n^3.
$$

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 theorem D.4 and then the upper bound derived above to each interval $(s_n, s_{n+1}]$, which yields

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

as desired. □

Remark D.5 (Discussion on the Positivity Assumption)**.** *In the statement of theorem 4.1, we have assumed that*

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

in (3.7) 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. Anderson & Mattingly (2011) showed that this approximation is at most of $\mathcal{O}(\Delta_n^3)$ within the corresponding interval and Hu et al. (2011a) further proved that for any order $p \geq 1$, there exists a step size Δ such that this approximation is at least p-th order, i.e.*, of order $\mathcal{O}(\Delta^p)$ for that step. Therefore, we believe the positive part approximation would not affect the performance of the proposed algorithm for the case of discrete diffusion models when the step size is not too large, which is also supported by our empirical studies.*

D.5 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 theorem D.6 below, whose proof is omitted here since it is similar to that of theorem D.4 above.

Theorem D.6. *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 (2.5) and the interpolating process (3.3) of the θ-RK-2 method (algorithm 1), then it holds that*

$$\begin{aligned}&D_{\mathrm{KL}}(\overleftarrow{p}_{T-\delta} \| \widehat{q}_{T-\delta}^{\mathrm{RK}}) \leq D_{\mathrm{KL}}(\overleftarrow{p}_{0:T-\delta} \| \widehat{q}_{0:T-\delta}^{\mathrm{RK}}) \\ \leq &D_{\mathrm{KL}}(\overleftarrow{p}_0 \| \widehat{q}_0) + \mathbb{E}\left[\int_0^{T-\delta} \int_{\mathbb{D}} \left(\mu_s(\nu) \log \frac{\mu_s(\nu)}{\widehat{\mu}_s^{\mathrm{RK}}(\nu)} - \mu_s(\nu) + \widehat{\mu}_s^{\mathrm{RK}}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \right],\end{aligned} \tag{D.2}$$

where the intensity $\widehat{\mu}^{\mathrm{RK}}$ is defined in (3.3), and the expectation is taken w.r.t. both paths generated by the backward process (2.5) and the randomnesss of the Poisson random measure used in the first step of each iteration of the algorithm, i.e.*, the construction of the intermediate process (3.1), which is assumed to be independent of that of the backward process.*

Following the same flow as in the proof of theorem 4.1, we will first provide an outline of the proof of theorem 4.2, and defer the proof of several key lemmas and detailed calculations are to appendix D.6 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 theorem 4.2. In the following proof sketch, we will be using the same notation as in the proof of theorem 4.1, 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 (3.1) as y_s^* and the corresponding intermediate intensity as appeared in (3.5) as $\widehat{\mu}_s^*$.

As defined in (3.4), the intensity $\widehat{\mu}^{\mathrm{RK}}(\nu)$ is given by

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

which helps us rewrite the corresponding part of the integral in (D.2) as

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

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 theorem 4.1 above.

Decomposition of the Integral. Then we perform a similar decomposition of the integral as in the proof of theorem 4.1 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 - \frac{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 + \frac{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 - \frac{1}{2\theta}\right) \left(\mu_{s_n}(\nu) \log \mu_{s_n}(\nu) - \mu_{s_n} \right) + \frac{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}} \frac{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 - \frac{1}{2\theta}\right) \mu_{s_n}(\nu) \log \widehat{\mu}_{s_n}(\nu) + \frac{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 - \frac{1}{2\theta}\right) \mu_{s_n}(\nu) + \frac{1}{2\theta} \mu_{\rho_n}(\nu) \right) \log \left(\left(1 - \frac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \frac{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 - \frac{1}{2\theta}\right) \mu_{s_n}(\nu) + \tfrac{1}{2\theta} \mu_{\rho_n}(\nu) \right) \log \left(\left(1 - \frac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \frac{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 - \frac{1}{2\theta}\right) \mu_{s_n}(\nu) + \frac{1}{2\theta} \mu_{\rho_n}(\nu) \right) \log \left(\left(1 - \frac{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 - \frac{1}{2\theta}\right) \mu_{s_n}(\nu) + \frac{1}{2\theta} \mu_{\rho_n}(\nu) \right) \log \left(\left(1 - \frac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \frac{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}} \mu_s(\nu) \log \left(\left(1 - \frac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \frac{1}{2\theta} \widehat{\mu}^*_{\rho_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 bound of the terms above,. Detailed calculations and proofs of the lemmas and propositions used here are deferred to appendix D.6.

(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 $\widehat{\mu}_s$ (assumption D.3) as follows

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

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

(ii) *Error related to the smoothness of intensity:* By corollary D.15 and corollary D.16, 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 proposition D.17 and corollary D.18. 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 appendix D.6 is $(\mathrm{III}.4)$, which is given by

$$\begin{aligned}\mathbb{E}\left[(\mathrm{III}.4)\right] = \mathbb{E}\Bigg[&\int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \frac{1}{2\theta}\right) \mu_{s_n}(\nu) \log \widehat{\mu}_{s_n}(\nu) + \frac{1}{2\theta} \mu_{\rho_n}(\nu) \log \widehat{\mu}_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\ - &\int_{s_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\left(1 - \frac{1}{2\theta}\right) \mu_{s_n}(\nu) + \frac{1}{2\theta} \mu_{\rho_n}(\nu) \right) \log \left(\left(1 - \frac{1}{2\theta}\right) \widehat{\mu}_{s_n}(\nu) + \frac{1}{2\theta} \widehat{\mu}_{\rho_n}(\nu) \right) \gamma(\mathrm{d}\nu)\mathrm{d}s \Bigg]\end{aligned} \tag{D.3}$$

Recall that in the proof of its counterpart (proposition D.8), we utilized the convexity of the loss function and the extrapolation nature of the second step in the θ-trapezoidal method (3.7) 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 (assumption D.3), we can reduce (D.3) to

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

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

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

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

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

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 remakr below. □

Remark D.7 (Discussions on the case when $\theta \in (\frac{1}{2}, 1]$)**.** *For $\theta \in (\frac{1}{2}, 1]$, the term (D.4) 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 (theorem D.9 and corollary D.10) to $\mu_s \log \widehat{\mu}_s$ in the term* (III.4)*, we have that the first integral in (D.3) can be expanded as follows*

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

Similarly, applying the Dynkin's formula to the following function

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

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 (D.3) as below

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

where

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

This further implies that

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

Comparing the first and second order terms in the two expansions of the two integrals in (D.3) above then implies that the term (III.4) *is of at most second order.*

D.6 LEMMAS AND PROPOSITIONS

In this section, we provide the detailed proofs of the lemmas and propositions omitted in the proof of theorem 4.1 and theorem 4.2.

Error due to the Intensity Estimation. Apart from the terms (I.1) and (II.1) in the proof of theorem 4.1 and the term (III.1) in the proof of theorem 4.2, 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 (3.7).

Proposition D.8. *For the interval $(s_n, s_{n+1}]$ for $n \in [0 : N-1]$, we have the following error bound:*

$$\begin{aligned}\mathbb{E}\left[(\mathrm{II}.4)\right] = \mathbb{E}\Bigg[\int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\alpha_1 \mu_{\rho_n}(\nu) \log \widehat{\mu}_{\rho_n}(\nu) - \alpha_2 \mu_{s_n}(\nu) \log \widehat{\mu}_{s_n}(\nu)\right) \gamma(\mathrm{d}\nu)\mathrm{d}s \\ - \int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\alpha_1 \mu_{\rho_n}(\nu) - \alpha_2 \mu_{s_n}(\nu)\right) \log\left(\alpha_1 \widehat{\mu}_{\rho_n}(\nu) - \alpha_2 \widehat{\mu}_{s_n}(\nu)\right) \gamma(\mathrm{d}\nu)\mathrm{d}s \Bigg] \lesssim \Delta_n \epsilon_{\mathrm{II}}.\end{aligned} \tag{D.5}$$

Proof. We first define and bound three error terms (II.4.1), (II.4.2), and (II.4.3) with score estimation error (assumption D.3) as follows:

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

Similarly, we also have

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

and

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

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

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

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

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

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 D.9 (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

$$\mathcal{L}f_t(y) = \lim_{\tau\to 0^+}\left[\frac{f_{t+\tau}(y_{t+\tau}) - f_t(y_t)}{\tau}\middle| y_t = y\right] = \partial_t f_t(y) + \int_{\mathbb{D}} \left(f_t(y+\nu) - f_t(y)\right)\mu_t(\nu, y)\gamma(\mathrm{d}\nu). \tag{D.6}$$

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 Øksendal & Sulem (2019).

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 Øksendal & Sulem (2019) 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 remark C.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 Øksendal & Sulem (2019), 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 D.10. *Suppose that the process $(y_t)_{t\in[0,\tau]}$ and the generator $\mathcal{L}$ are defined as in theorem D.9. 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

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

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.*, $y_t \equiv y$. corollary D.10 offers an analogous approach to perform the expansion when the process $(y_t)_{t\in[0,\tau]}$ is not constant.

Lemma D.11. *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 (2.5), 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 theorem D.9. By applying the Dynkin's formula (theorem D.9 and corollary D.10) 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

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

and the proof is complete. □

Then we are ready to bound some of the error terms in the proof of theorem 4.1 with lemma D.11.

Corollary D.12. *For the interval $(s_n, s_{n+1}]$ for $n \in [0:N-1]$, we have the following error bound:*

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

Proof. The bound is obtained by applying lemma D.11 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 D.13. *For the interval $(s_n, s_{n+1}]$ for $n \in [0 : N-1]$, we have the following error bound:*

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

Proof. Note that the intermediate process y^*_s defined in (3.1) 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 lemma D.11 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 $(\mathrm{III}.6)$ in theorem 4.2, for which we need the following variant of lemma D.11.

Lemma D.14. *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 (2.5), it holds that*

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

Proof. The proof is similar to that of lemma D.11. Specifically, we let $\mathcal{L}$ be the generator defined in theorem D.9, apply the Dynkin's formula (theorem D.9 and corollary D.10) to the function $f_t(y_t)$ and plug in the expression of the generator $\mathcal{L}$, which yields

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

as desired. □

Corollary D.15. *For the interval $(s_n, s_{n+1}]$ for $n \in [0 : N-1]$, we have the following error bound:*

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

Proof. By applying lemma D.14 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 D.16. *For any $n \in [0 : N-1]$ and the corresponding interval $(s_n, s_{n+1}]$, we have the following error bound:*

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

Proof. Following the arguments in the proof of corollary D.13, 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 lemma D.14 with f being the function

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

as desired. □

Error involving the Intermediate Process.

Proposition D.17. *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

$$\begin{aligned}
\mathbb{E}\left[(\mathrm{II}.3)\right] &= \mathbb{E}\left[\int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\widehat{\mu}^*_{\rho_n}(\nu) - \widehat{\mu}_{\rho_n}(\nu)\right)\gamma(\mathrm{d}\nu)\mathrm{d}s\right] \\
&\lesssim \int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left(\mathbb{E}\left[\widehat{\mu}^*_{\rho_n}(\nu)\right] - \mathbb{E}\left[\widehat{\mu}_{\rho_n}(\nu)\right]\right)\gamma(\mathrm{d}\nu)\mathrm{d}s.
\end{aligned} \tag{D.7}$$

Then we expand the integrand by applying the Dynkin's formula (theorem D.9 and corollary D.10) 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 (D.6) 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

$$\begin{aligned}
\mathcal{L}^* f_s(y) &= \partial_s f_s(y) + \int_{\mathbb{D}} \left(f_s(y+\nu) - f_s(y)\right)\widehat{\mu}_{s_n}(\nu)\gamma(\mathrm{d}\nu), \\
\mathcal{L} f_s(y) &= \partial_s f_s(y) + \int_{\mathbb{D}} \left(f_s(y+\nu) - f_s(y)\right)\mu_s(\nu)\gamma(\mathrm{d}\nu).
\end{aligned} \tag{D.8}$$

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

$$\begin{aligned}\mathbb{E}\left[|(\mathcal{L}^* - \mathcal{L})\widehat{\mu}_{s_n}(\nu)|\right] &= \mathbb{E}\left[\left|\int_{\mathbb{D}} \left(\widehat{\mu}_{s_n}(y+\nu) - \widehat{\mu}_{s_n}(y)\right)\left(\widehat{\mu}_{s_n}(\nu) - \mu_{s_n}(\nu)\right)\gamma(\mathrm{d}\nu)\right|\right] \\ &\lesssim \mathbb{E}\left[\int_{\mathbb{D}} \left|\widehat{\mu}_{s_n}(\nu) - \mu_{s_n}(\nu)\right|\gamma(\mathrm{d}\nu)\right] \lesssim \epsilon_{\mathrm{II}},\end{aligned} \tag{D.9}$$

where we used the assumption on the estimation error (assumption D.3) in the last inequality. Then we can further reduce (D.7) 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 D.18. *For the interval $(s_n, s_{n+1}]$ for $n \in [0 : N-1]$, we have the following error bound:*

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

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 (assumption D.2 (II))

$$\begin{aligned}\mathbb{E}\left[(\mathrm{II}.5)\right] \lesssim& \mathbb{E}\left[\int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left|\alpha_1\mu_{\rho_n}(\nu) - \alpha_2\mu_{s_n}(\nu)\right| \frac{1}{\alpha_1\widehat{\mu}_{\rho_n}(\nu) - \alpha_2\widehat{\mu}_{s_n}(\nu)}\alpha_1\left|\widehat{\mu}_{\rho_n}(\nu) - \widehat{\mu}^*_{\rho_n}(\nu)\right|\gamma(\mathrm{d}\nu)\mathrm{d}s\right] \\ \lesssim& \mathbb{E}\left[\int_{\rho_n}^{s_{n+1}} \int_{\mathbb{D}} \left|\widehat{\mu}_{\rho_n}(\nu) - \widehat{\mu}^*_{\rho_n}(\nu)\right|\gamma(\mathrm{d}\nu)\mathrm{d}s\right] \lesssim \Delta_n\mathbb{E}\left[\int_{\mathbb{D}} \left|\widehat{\mu}_{\rho_n}(\nu) - \widehat{\mu}^*_{\rho_n}(\nu)\right|\gamma(\mathrm{d}\nu)\right] \\ =& \Delta_n \int_{\mathbb{D}} \mathbb{E}\left[\left|\widehat{\mu}_{\rho_n}(\nu) - \widehat{\mu}^*_{\rho_n}(\nu)\right|\right]\gamma(\mathrm{d}\nu)\end{aligned} \tag{D.10}$$

Applying the same arguments as in proposition D.17, which uses the generators $\mathcal{L}$ and $\mathcal{L}^*$ defined in (D.8), we can bound the RHS above as follows

$$\begin{aligned}\mathbb{E}\left[\left|\widehat{\mu}^*_{\rho_n}(\nu) - \widehat{\mu}_{\rho_n}(\nu)\right|\right] =& \mathbb{E}\left[\left|\left(\widehat{\mu}_{s_n}(\nu) + \mathcal{L}^*\widehat{\mu}_{s_n}(\nu)\Delta_n + \mathcal{O}(\Delta_n^2)\right) - \left(\widehat{\mu}_{s_n}(\nu) + \mathcal{L}\widehat{\mu}_{s_n}(\nu)\Delta_n + \mathcal{O}(\Delta_n^2)\right)\right|\right] \\ \lesssim& \Delta_n\mathbb{E}\left[|(\mathcal{L}^* - \mathcal{L})\widehat{\mu}_{s_n}(\nu)|\right] + \mathcal{O}(\Delta_n^2) \lesssim \Delta_n\epsilon_{\mathrm{II}} + \mathcal{O}(\Delta_n^2)\end{aligned} \tag{D.11}$$

where the last inequality follows from (D.9). Substituting (D.11) into (D.10) then yields the desired upper bound. □

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

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

Proof. First, we observe by Dynkin's formula (theorem D.9) that

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

Secondly, applying the given assumption (assumption D.2 (II)) on the boundedness of the intensities yields

$$\begin{aligned}\mathbb{E}\left[\left|\log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)\right)-\log\widehat{\mu}_{s_n}(\nu)\right|\right]\lesssim&\frac{1}{\widehat{\mu}_{s_n}(\nu)}\mathbb{E}\left[\left|\alpha_1\widehat{\mu}^*_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)-\widehat{\mu}_{s_n}(\nu)\right|\right]\\ \lesssim&\mathbb{E}\left[\left|\alpha_1\widehat{\mu}^*_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)-\widehat{\mu}_{s_n}(\nu)\right|\right]\\ \leq&\mathbb{E}\left[\left|\widehat{\mu}^*_{\rho_n}(\nu)-\widehat{\mu}_{\rho_n}(\nu)\right|\right]\lesssim\Delta_n\epsilon_{\mathrm{II}}+\mathcal{O}(\Delta_n^2),\end{aligned}\tag{D.12}$$

where the last inequality follows from (D.11) proved above. Therefore, we may further deduce that

$$\begin{aligned}\mathbb{E}\left[(\mathrm{I}.3)\right]\leq&\int_{s_n}^{\rho_n}\int_{\mathbb{D}}\mathbb{E}\left[|\mu_s(\nu)-\mu_{s_n}(\nu)|\right]\mathbb{E}\left[\left|\log\left(\alpha_1\widehat{\mu}^*_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)\right)-\log\left(\alpha_1\widehat{\mu}_{\rho_n}(\nu)-\alpha_2\widehat{\mu}_{s_n}(\nu)\right)\right|\right]\gamma(\mathrm{d}\nu)\mathrm{d}s\\ \lesssim&\Delta_n^2(\Delta_n\epsilon_{\mathrm{II}}+\Delta_n^2)\lesssim\Delta_n^3\epsilon_{\mathrm{II}}+\Delta_n^4,\end{aligned}$$

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. □

E Details of Numerical Experiments

In this section, we describe in detail the setting for each numerical experiment. In appendix E.1, we discuss a revision of θ-RK-2 (algorithm 1) for a more practical and better-performing implementation in real cases. In appendices E.2 to E.4, we present additional numerical results for the 15-dimension toy model, text generation, and image generation respectively.

E.1 Practical Implementation of θ-Runge Kutta-2

As is mentioned in theorem 4.2, when we fix $\theta\in(0,\frac{1}{2}]$ for the θ-RK-2 method, the algorithm also enjoys a second order convergence in theory conditioned on the fact that the extrapolated transition rate matrix $(1-\frac{1}{2\theta})\widehat{\mu}_{s_n}+\frac{1}{2\theta}\widehat{\mu}^*_{\rho_n}$ is everywhere non-negative. In practice, we force this condition to be true by only taking the positive parts of this rate matrix, leading to the revised practical implementation in algorithm 4.

By introducing this modification, we manage to extend the θ range to $(0,1]$, the same as the θ-Trapezoidal algorithm. In the following sections, we will also present results for θ-RK-2, and it is realized by implementing the version of algorithm 4 with a feasible $\theta\in(0,1]$.

Algorithm 4: Practical Implementation of θ-Runge Kutta-2 Algorithm

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 proposition 3.2.
Output: A sample $\widehat{y}_{s_N}\sim\widehat{q}^{\mathrm{RK}}_{t_N}$.

```
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_{s_n} + Σ_{ν∈𝔻} ν𝒫((1 − 1/(2θ))μ̂_{s_n} + 1/(2θ) μ̂*_{ρ_n})_+ (ν)Δ_n);
end
```

E.2 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.

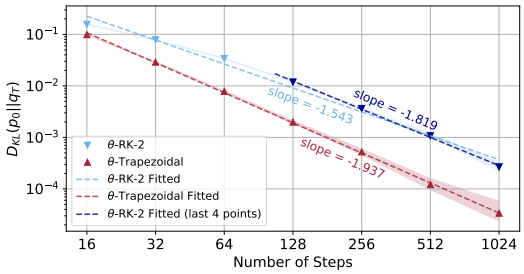

Figure 2: Empirical KL divergence between the true distribution and the generated distribution of the toy model vs. the number of steps. Data are fitted with linear regression and shaded with 95% confidence intervals by bootstrapping.

E.3 Text Generation

For text generation, we use the small version of RADD (Ou et al., 2024) 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,

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

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 Ou et al. (2024),

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

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 table 1. For each method, we generate 1024 samples with it and compute the averaged perplexities. All the experiments are run on a single NVIDIA A100 GPU.

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

Table 1: Generative perplexity of texts generated by different sampling algorithms. Lower values are better, with the best in **bold**.

Sampling Methods	NFE $= 16$	NFE $= 32$	NFE $= 64$	NFE $= 128$	NFE $= 256$	NFE $= 512$	NFE $= 1024$
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
θ-RK-2	≤ 127.363	≤ 109.351	≤ 86.102	≤ 64.317	≤ 49.816	≤ 40.375	≤ 33.971
θ-Trapezoidal	$\mathbf{\leq 123.585}$	$\mathbf{\leq 89.912}$	$\mathbf{\leq 66.549}$	$\mathbf{\leq 49.051}$	$\mathbf{\leq 39.959}$	$\mathbf{\leq 32.456}$	$\mathbf{\leq 27.553}$

From the table, we observe that θ-Trapezoidal consistently outperforms all other approaches and generates samplers with better perplexities across all NFEs. We also noticed that both the Euler method and Tweedie τ-leaping share a similar performance, which is beaten by a large margin by θ-RK-2 and τ-leaping.

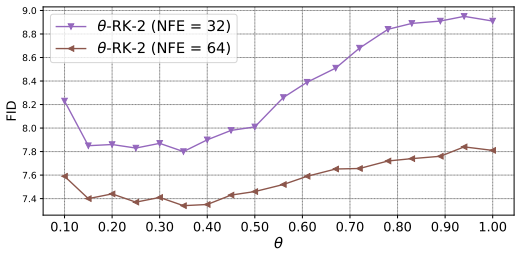

Figure 3: Sampling quality v.s. $\theta \in (0, 1]$ in θ-RK-2 algorithm. Sampling quality is quantified through FID.

In fig. 3, we present the performance of θ-RK-2 with respect to different choices of θ at NFE 32 and 64. We observe that the performance of θ-RK-2 has a flat landscape around the optimal θ choices, which falls in the range $[0.15, 0.4]$. In general, as is evident from the curve, the method performs better when using extrapolation to compute the transition rate matrix, which once again certifies the correctness of our theoretical results (theorem 4.2) and discussions therebelow.

E.4 Image Generation

For the image generation, we use the checkpoint of MaskGIT (Chang et al., 2022; Besnier & Chen, 2023) reproduced in Pytorch[2]. 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 (E.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 (E.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, the same as the recommended practice in Chang et al. (2022). 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 the generation quality and choose the guidance strength to be $w = 3$.

We present the full results for NFE ranging from 4 to 64 in fig. 4. 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

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

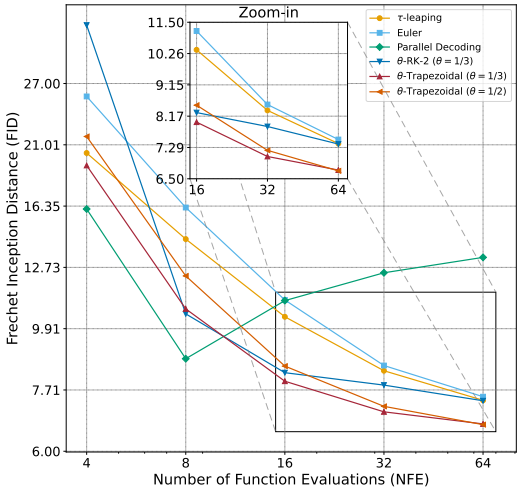

Figure 4: FID of images generated by sampling algorithms vs. number of function evaluations (NFE) with different parameter choices. Lower values are better.

competitive performance, it converges to the same generation quality as $\theta = \frac{1}{3}$ in 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.

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. 3. Again, similar to the behavior of θ-Trapezoidal, 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 is evident from the curve, the method performs better when using extrapolation to compute the transition rate matrix, which once again certifies the correctness of our theoretical results.

Finally, we visualize some images generated with θ-Trapezoidal on 6 different classes in fig. 5. θ-Trapezoidal consistently generates high-fidelity images that are visually similar to the ground truth ones and well aligned with the concept.

E.5 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. 6, we observe that the θ-trapezoidal method demonstrates notable robustness to θ, with a flat landscape near the optimal choice. Our empirical analysis suggests that $\theta \in [0.3, 0.5]$ consistently yield competitive performance across different tasks.

Figure 5: 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**).

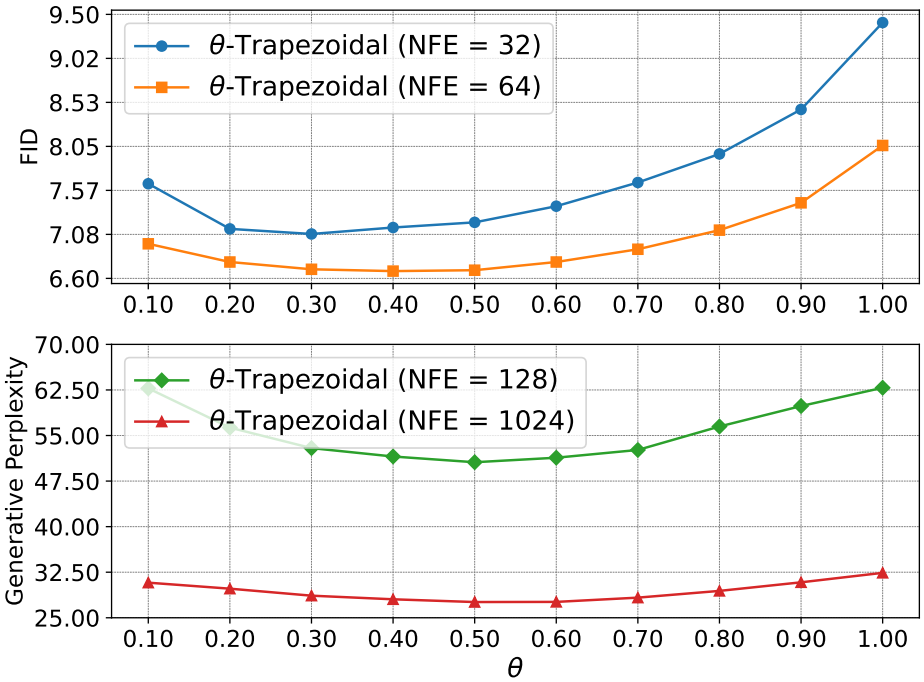

Figure 6: Sampling quality v.s. $\theta \in (0, 1]$ in θ-Trapezoid method. **Upper**: Image generation, the metric is FID; **Lower**: Text generation, the metric is generative perplexity. Lower values are better.