DIFFUSION LANGUAGE MODELS AS OPEN QUANTUM SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models generate text by reversing a classical diffusion process over token sequences. We show that their continuous-time forward and reverse dynamics can be embedded in the framework of open quantum systems, which describes quantum systems interacting with their environment. This connection provides a basis for extending diffusion sampling to superpositions of sentence states and for modifying the underlying generation dynamics. We construct a τ-leaping unraveling scheme that simulates stochastic trajectories of these states while updating multiple token positions at each step. This enables generation conditioned on superposed prompts or thoughts, which we call Superposition of Prompts (SoP) and Superposition of Thoughts (SoT). The same formulation yields Quantum-Inspired Remasking (QIR): perturbing the masked-diffusion dynamics produces a confidence-based rule for revising previously generated tokens. Experiments with pretrained diffusion language models show that superposition-based generation can retain information from conditioning components not selected at the final measurement, while QIR improves LLM-assessed text quality when applied at an appropriate stage of generation.

1 INTRODUCTION

Diffusion models have achieved remarkable success in generative tasks such as image and video synthesis (Dhariwal & Nichol, 2021; Ho et al., 2020; 2022; Rombach et al., 2022; Singer et al., 2023; Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Song et al., 2021b). Diffusion-based approaches are also increasingly being explored for text generation as an alternative to autoregressive models (Lou et al., 2024; Nie et al., 2025; Ou et al., 2025; Sahoo et al., 2025; Sun et al., 2023; Xu et al., 2025; Zhao et al., 2025), including recent commercial systems (Khanna et al., 2025; Song et al., 2025).

From a theoretical point of view, diffusion models are deeply connected to stochastic processes in non-equilibrium statistical mechanics (Sohl-Dickstein et al., 2015). This connection has motivated a growing body of work that analyzes diffusion models using concepts and formulations developed in physics (Hirono et al., 2024b; Ikeda et al., 2024; Yaguchi et al., 2025; Yu & Huang, 2024). Conventional diffusion models can be viewed as analogs of *classical* thermal systems, whose dynamics can be described within a more general *quantum* framework.

In this paper, we formulate diffusion language models within the framework of *open quantum systems* (Breuer & Petruccione, 2002; Gorini et al., 1976; Lindblad, 1976). Specifically, we show that their continuous-time dynamics can be embedded in the Lindblad equation, which describes the evolution of a quantum system interacting with its environment. This formulation opens the way to extensions of diffusion sampling based on concepts from quantum physics. One such concept is *superposition*. As illustrated in Figure 1, distinct sentences can be represented as basis states, and a linear combination of these states defines a superposition state $|\psi_t\rangle$. Superposition is fundamentally different from concatenation: the sentences remain distinct alternatives within a quantum state, rather than becoming successive parts of a longer text. The corresponding density operator evolves according to the Lindblad equation. This representation introduces a form of generation in which multiple sentence components and their relative weights evolve together, with the components influencing one another rather than evolving as independent samples.

We develop a sampling procedure based on τ-leaping unraveling, which evolves superpositions of sentence states by updating multiple token positions in each step. It builds on *stochastic unraveling*,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

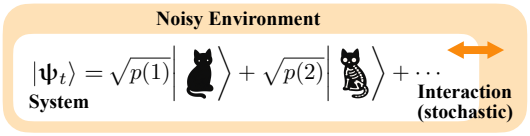


Figure 1: A superposition of sentence states viewed as an open quantum system interacting with its environment.

an established technique for representing Lindblad dynamics through stochastic pure-state trajectories (Carmichael, 1993; Dalibard et al., 1992; Dum et al., 1992). The idea is analogous to simulating trajectories of a stochastic differential equation instead of directly solving the corresponding Fokker–Planck equation. A final measurement selects an output sentence according to the Born rule. The same formulation also leads to quantum-inspired remasking: a perturbation of the masked-diffusion dynamics yields a confidence-based rule that preferentially remasks tokens the model is less confident in reconstructing from their context.

Our main contributions are summarized as follows:

- We show that continuous-time discrete diffusion language models can be embedded in the Lindblad equation, with reverse diffusion recovered in the diagonal sector of the reverse-time dynamics. This embedding allows us to extend diffusion language models by modifying the dynamics or allowing superpositions of sentence states.
- By allowing superpositions in the same formulation, we introduce **Superposition of Prompts (SoP)** and its application to thought-conditioned generation, **Superposition of Thoughts (SoT)**. These methods generate a text through stochastic unraveling followed by measurement. Our experiments show that the output can retain information from conditioning components that are not selected in the final measurement.
- By perturbing the embedded masked diffusion dynamics, we derive **Quantum-Inspired Remasking (QIR)**. Our experiments show that appropriately timed QIR improves LLM-assessed text quality, whereas applying it too early increases repetition despite lowering perplexity.

These methods illustrate the broader design space opened by our formulation, which brings concepts from quantum physics into the design of language generation processes. Exploring this space may lead to further sampling methods on classical hardware, as well as future implementations on quantum computers.

1.1 Related Works

Diffusion Language Models. Diffusion models were initially introduced for binary data (Sohl-Dickstein et al., 2015) and later extended to discrete state spaces. Our mathematical foundation is the continuous-time Markov formulation of discrete diffusion (Campbell et al., 2022; Sun et al., 2023), particularly that of Lou et al. (2024). Two widely studied approaches are *uniform diffusion* and *masked diffusion*.

Uniform diffusion allows tokens to be revised throughout generation, providing a mechanism for correcting earlier predictions (Hoogeboom et al., 2021). Although early models faced challenges in language modeling (Austin et al., 2021; Lou et al., 2024), recent advances have improved training (Sahoo et al., 2025) and led to practical open-weight models (DiffusionGemma Team et al., 2026). Masked diffusion generates a text by progressively replacing mask tokens and has been widely adopted for its stable training (Austin et al., 2021; Lou et al., 2024). Ou et al. (2025) showed that the structure of masked diffusion permits time-independent modeling and reduces the number of function evaluations during sampling. Masked diffusion also underlies large-scale language models such as LLaDA (Nie et al., 2025).

Standard masked diffusion cannot revise tokens once unmasked. *Remasking* allows these tokens to be masked again and regenerated. Nie et al. (2025) retain a prescribed number of token predictions based on confidence at each step. WINO (Hong et al., 2026) also revisits previously generated tokens, remasking those below a verification confidence threshold. Our QIR instead derives stochastic remasking rates from perturbed Lindblad dynamics, assigning higher rates to tokens with lower reconstruction confidence. Alternatively, Wang et al. (2026) reformulate sampling to incorporate remasking, drawing on DDIM (Song et al., 2021a).

Quantum Computation and Diffusion Models. The connection between quantum computation and generative modeling has been explored in several directions. Early approaches include quantum generative models such as quantum Boltzmann machines and variational quantum circuits (Amin et al., 2018; Benedetti et al., 2019; Biamonte et al., 2017), where probability distributions are represented by quantum states and sampled via measurement.

Diffusion Models and Physics. Diffusion models have been studied from the perspectives of non-equilibrium statistical mechanics and stochastic processes (Hirono et al., 2024a; Ikeda et al., 2024; Sohl-Dickstein et al., 2015; Yaguchi et al., 2025; Yu & Huang, 2024). Connections between classical diffusion and quantum dynamics also arise through imaginary-time evolution and path-integral formulations (Feynman & Hibbs, 1965). Nasu et al. (2025) connect Lindblad dynamics to classical reverse diffusion through a semiclassical approximation. Our work considers an exact embedding of discrete token dynamics and uses it to develop sampling procedures for diffusion language models.

2 Background

We review discrete diffusion models and the Lindblad description of open quantum systems as the background for the formulation in the next section. Our notation is summarized in Section A.

2.1 Discrete Diffusion Models

In continuous-time discrete diffusion models, the forward process gradually corrupts token sequences. The corresponding probability distribution evolves according to the master equation

$$\frac{d}{dt}p_t(\mathbf{m}) = \sum_{\mathbf{n}\in\mathbb{V}^N} Q_t(\mathbf{m},\mathbf{n})p_t(\mathbf{n}), \quad \begin{cases} Q_t(\mathbf{m},\mathbf{n}) \geq 0 & (\mathbf{m}\neq\mathbf{n}) \\ \sum_{\mathbf{m}\in\mathbb{V}^N} Q_t(\mathbf{m},\mathbf{n}) = 0 \end{cases}, \tag{1}$$

where $\mathbf{m},\mathbf{n}\in\mathbb{V}^N$ are token sequences of length N, and Q_t is the transition-rate matrix (see Section A for notations). Generation proceeds by reversing this diffusion process. The probability distribution then evolves according to

$$-\frac{d}{dt}p_t(\mathbf{m}) = \sum_{\mathbf{n}\in\mathbb{V}^N} \tilde{Q}_t(\mathbf{m},\mathbf{n})p_t(\mathbf{n}), \quad \begin{cases} \tilde{Q}_t(\mathbf{m},\mathbf{n}) = Q_t(\mathbf{n},\mathbf{m})\frac{p_t(\mathbf{m})}{p_t(\mathbf{n})} & (\mathbf{m}\neq\mathbf{n}) \\ \sum_{\mathbf{m}\in\mathbb{V}^N} \tilde{Q}_t(\mathbf{m},\mathbf{n}) = 0 \end{cases}. \tag{2}$$

Simulating the reverse process requires the probability ratios $p_t(\mathbf{m})/p_t(\mathbf{n})$, known as concrete scores Lou et al. (2024), which the model is trained to estimate. A common choice is to let each transition change only one token position. The forward and reverse generators then take the form

$$Q_t(\mathbf{m},\mathbf{n}) = \sigma(t)\sum_{p=1}^{N} 1_{\mathbf{m}_{-p}\mathbf{n}_{-p}} Q^{\text{tok}}(m_p,n_p), \tag{3}$$

$$\tilde{Q}_t(\mathbf{m},\mathbf{n}) = \sigma(t)\sum_{p=1}^{N} 1_{\mathbf{m}_{-p}\mathbf{n}_{-p}} \tilde{Q}^{(p)\text{tok}}(m_p,\mathbf{n}),\ \tilde{Q}^{(p)\text{tok}}(m_p,\mathbf{n}) = Q^{\text{tok}}(n_p,m_p)\overbrace{\frac{p_t(\mathbf{n}_{p\to m_p})}{p_t(\mathbf{n})}}^{s(\mathbf{n},t)_{p\to m_p}}. \tag{4}$$

where $\sigma(t)$ controls the time-dependent transition rate, and p indexes token positions. The indicator $1_{\mathbf{m}_{-p}\mathbf{n}_{-p}}$ equals one when $\mathbf{m}$ and $\mathbf{n}$ agree at all positions other than p, and zero otherwise. The sequence $\mathbf{n}_{p\to m_p}$ is obtained from $\mathbf{n}$ by replacing its p-th token with m_p. With this transition structure, the model only needs to estimate the scores $s(\mathbf{n},t)_{p\to m_p}$ associated with single-token changes, rather than probability ratios between arbitrary pairs of sequences.

2.2 Open Quantum Systems and the Lindblad Equation

An open quantum system describes the effective dynamics of a quantum system interacting with its environment. Its state is represented by a density operator ρ_t on a Hilbert space $\mathcal{H}$. Classical probability distributions are recovered as a special case. Given the sentence basis $\{|\mathbf{m}\rangle\}_{\mathbf{m}\in\mathbb{V}^N}$, a

probability distribution $p_t(\mathbf{m})$ over sequences can be embedded into a diagonal density operator,

$$\rho_t = \sum_{\mathbf{m}\in\mathbb{V}^N} p_t(\mathbf{m}) \left|\mathbf{m}\right\rangle \left\langle\mathbf{m}\right|. \tag{5}$$

Thus, a discrete diffusion process can be viewed as open-system dynamics restricted to the diagonal sector. Going beyond this diagonal sector amounts to allowing coherent superpositions of token sequences, which will be exploited later in our sampling construction.

The Markovian dynamics of an open quantum system is governed by the Lindblad equation,

$$\frac{d}{dt}\rho_t = -i[H_t, \rho_t] + \sum_\mu L_t^\mu \rho_t L_t^{\mu\dagger} - \frac{1}{2}\left\{\sum_\mu L_t^{\mu\dagger} L_t^\mu, \rho_t\right\}. \tag{6}$$

Here, H_t is a Hermitian operator, called the Hamiltonian, and L_t^μ are jump operators. The Hamiltonian term generates coherent evolution, while the jump terms describe dissipative transitions induced by the environment. The Lindblad equation preserves positivity and unit trace of ρ_t, and is therefore the quantum analogue of a classical master equation (1).

The open-system formulation also admits a reverse-time dynamics, paralleling the reverse diffusion equation (2). Given the forward Lindblad dynamics and its solution ρ_t, one can define time-reversed operators $\tilde{H}_t$ and $\tilde{L}_t^\mu$ such that

$$-\frac{d}{dt}\rho_t = -i[\tilde{H}_t, \rho_t] + \sum_\mu \tilde{L}_t^\mu \rho_t \tilde{L}_t^{\mu\dagger} - \frac{1}{2}\left\{\sum_\mu \tilde{L}_t^{\mu\dagger} \tilde{L}_t^\mu, \rho_t\right\}. \tag{7}$$

The explicit expressions for $\tilde{H}_t$ and $\tilde{L}_t^\mu$ are reviewed in Section C. For diagonal embeddings of the form (5), this reverse-time Lindblad equation reduces to the reverse-time dynamics of discrete diffusion models. This connection will be used below to formulate inference as a stochastic unraveling of the reverse open-system dynamics.

3 A Quantum Formulation of Diffusion Language Models

3.1 Embedding Diffusion Language Models into Open Quantum Systems

We now show that diffusion language models can be embedded in the Lindblad framework for open quantum systems. Specifically, the discrete diffusion dynamics can be reproduced by choosing jump operators that act on individual token positions, following the transition structure in Equation (3). The following theorem establishes this correspondence. A proof is given in Section D.

Theorem 3.1: Embedding of Diffusion Language Models Define the Hamiltonian and jump operators with $\mu = (p, m, n)$ acting on the sentence Hilbert space $\mathcal{H}$ as follows:

$$H_t = 0, \quad L_t^{\mu=(p,m,n)} = I \otimes \cdots \otimes \sqrt{\sigma(t)} L_{m,n}^{(p)\text{tok}} \otimes \cdots \otimes I, \tag{8}$$

where we define operators acting on the one-token Hilbert space $\mathcal{H}_\mathbb{V}$ by

$$L_{m,n}^{\text{tok}} = \sqrt{Q^{\text{tok}}(m,n)} \left|m\right\rangle \left\langle n\right|, \quad (m \neq n), \tag{9}$$

where $m, n \in \mathbb{V}$ are tokens before and after the jump. Then the Lindblad equation in Equation (6) reduces to be equivalent to the classical diffusion equation in Equation (1) with Equation (3).

Theorem 3.2: Time-reversal Jump Operators Under our setting in Theorem 3.1, we have the operators for the time-reversal equation in Equation (7) as follows:

$$\tilde{H}_t = 0, \quad \tilde{L}_t^{\mu=(p,m,n)} = \sum_{\mathbf{k}\in\mathbb{V}^N} \left|k_1\right\rangle \left\langle k_1\right| \otimes \cdots \otimes \sqrt{\sigma(t)} \tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k}) \otimes \cdots \otimes \left|k_N\right\rangle \left\langle k_N\right|,$$

where we define operators acting on the one-token Hilbert space $\mathcal{H}_\mathbb{V}$ by

$$\tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k}) = \sqrt{\tilde{Q}^{(p)\text{tok}}(n, \mathbf{k})} 1_{mk_p} \left|n\right\rangle \left\langle k_p\right|. \tag{10}$$

> With these operators, the time-reversed Lindblad equation (7) reproduces the reverse diffusion dynamics in Equation (2), with the generator given by Equation (4).

This embedding provides a starting point for extending diffusion sampling to superposition states and perturbed jump dynamics, which lead to the sampling schemes introduced below. Below, we show how the well known diffusion model settings are embedded in the Lindblad equation.

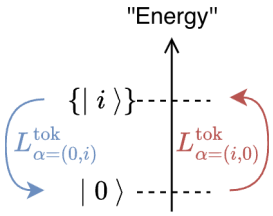
Uniform Diffusion. If we consider $\alpha = (m, n) \in \mathbb{V}^2$ where $m \neq n$ and consider jump operators

$$L^{\text{tok}}_{\alpha=(m,n)} = |m\rangle\langle n| \quad \Rightarrow \quad \tilde{L}^{(p)\text{tok}}_{\alpha=(m,n),t}(\mathbf{k}) = \sqrt{\frac{p_t(\mathbf{k}_{p\to n})}{p_t(\mathbf{k}_{p\to m})}}\, 1_{k_p m}\, |n\rangle\langle m| \,. \tag{11}$$

Masked Diffusion. If we consider $\alpha = (0, i) \in \mathbb{V}^2$ where $i \neq 0$ and consider jump operators

$$L^{\text{tok}}_{\alpha=(0,i)} = |0\rangle\langle i| \quad \Rightarrow \quad \tilde{L}^{(p)\text{tok}}_{\alpha=(0,i),t}(\mathbf{k}) = \sqrt{\frac{p_t(\mathbf{k}_{p\to i})}{p_t(\mathbf{k}_{p\to 0})}}\, 1_{k_p 0}\, |i\rangle\langle 0| \,. \tag{12}$$

For uniform diffusion defined by Equation (11), transitions between token states occur at equal rates in both directions. This can be viewed as thermal dynamics among states of equal energy.
In masked diffusion, transitions occur only from non-mask states $|i\rangle$ to the absorbing mask state $|0\rangle$. A thermal interpretation is obtained by assigning a lower energy to $|0\rangle$: when the energy gap is large compared with the temperature, upward transitions are suppressed relative to downward transitions. Masked diffusion corresponds to the limiting case in which upward transitions are absent.



Perturbed Masked Diffusion. To mitigate this energy gap, one approach would be to introduce corresponding jump operators operating in the opposite direction into the system, i.e., with $0 \leq \gamma \leq 1$:

$$L^{\text{tok}}_{\alpha=(i,0)} = \sqrt{\gamma}\,|i\rangle\langle 0| \quad \Rightarrow \quad \tilde{L}^{(p)\text{tok}}_{\alpha=(i,0),t}(\mathbf{k}) = \sqrt{\gamma}\sqrt{\frac{p_t(\mathbf{k}_{p\to 0})}{p_t(\mathbf{k}_{p\to i})}}\, 1_{k_p i}\, |0\rangle\langle i| \,, \tag{13}$$

where γ regulates the energy gap, with the gap completely closing when $\gamma = 1$. When $\gamma \approx 0$, it can be physically interpreted as a normal Masked diffusion process with added perturbations. Leveraging this latter property, the second half of this paper proposes a "quantum-inspired remasking" method based on Equation (13) and demonstrates its effectiveness.

3.2 Simulations: Stochastic Unraveling

The Lindblad formulation also provides a trajectory-level description of the dynamics. In ordinary diffusion language models, sampling is performed by simulating a Markov process over token sequences. The open-system counterpart is a stochastic process over wave functions, known as *stochastic unraveling*. It represents the same density-operator dynamics through random pure-state trajectories. In this language, a stochastic time-evolution operator $U^{\text{eff}}_{t-dt|t}$ updates the state vector as

$$|\psi_{t-dt}\rangle = \tilde{U}^{\text{eff}}_{t-dt|t}\,|\psi_t\rangle \propto (1 - i\tilde{H}^{\text{eff}}_t dt)\,|\psi_t\rangle + \sum_\mu (\tilde{L}^\mu_t - 1) dN^\mu_t\,|\psi_t\rangle \,, \tag{14}$$

where $\tilde{H}^{\text{eff}}_t = -\frac{i}{2}\sum_\mu \tilde{L}^{\mu\dagger}_t \tilde{L}^\mu_t$. For explicit expressions, see Section C.4. A direct application of these infinitesimal unravelings is inefficient for the same reason as the naive simulation of discrete diffusion: in a single step, they affect only one token position. Motivated by the τ-leaping sampler for discrete diffusion, we introduce an unraveling method that allows multiple token positions to be updated within one step. $|\psi_{t-\Delta t}\rangle = \tilde{U}^{\tau\text{-leap}}_{t-\Delta t|t}\,|\psi_t\rangle$. We refer to this method as *τ-leaping unraveling*. This expression is the unraveling analogue of the τ-leaping sampler in Equation (45)[1]. It reproduces the solution of the reverse-time Lindblad equation Equation (7) up to first order in Δt; see Theorem D.8.

[1] Notably, the latter $\tilde{U}^{\tau\text{-leap (P)}}_{t-\Delta t|t}$ becomes essentially equivalent to conventional τ-leaping with $|\psi\rangle = |\mathbf{m}\rangle$.

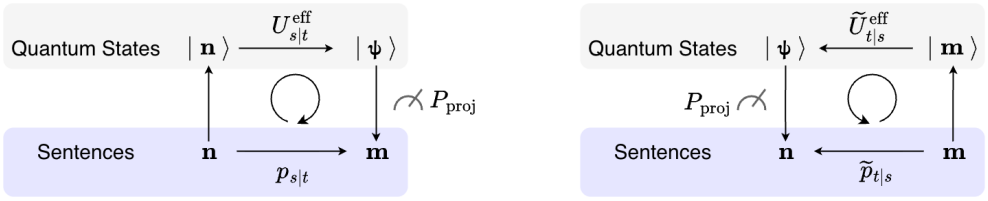


Figure 2: Sampling by stochastic unraveling followed by projective measurement. The ensemble of pure-state trajectories reproduces the corresponding density-operator dynamics, while the final measurement yields a discrete token sequence.

A central feature of stochastic unraveling is that an ensemble of stochastic pure-state trajectories reproduces the solution of the Lindblad equation at the level of the density matrix. In the sampling procedure, we need to perform a "projective measurement," since the wave function $|\boldsymbol{\psi}_t\rangle$ is, in general, a superposition of multiple sentences. According to the Born rule, the probability of measuring a particular sentence is given by the squared magnitude of its coefficient. For details, see Section D.3. This holds for both forward- and reverse-time dynamics, as illustrated in Figure 2.

3.3 Quantum-Inspired Samplings

Embedding diffusion language models into the Lindblad framework opens up new possibilities for extending both their states and dynamics. We explore two such extensions here. Superpositions of sentence states lead to Superposition of Prompts (SoP) and its thought-conditioned variant, Superposition of Thoughts (SoT), while perturbations of the jump operators yield Quantum-Inspired Remasking (QIR).

Superposition of Prompts (SoP). SoP applies stochastic unraveling to a superposition of sentence states conditioned on different prompts. In the context of score-based diffusion models, it is common to generate based on conditional probability $\mathbf{m} = [\texttt{Prompt}\cdots] \sim p(\cdots|\texttt{Prompt})$. This procedure is a counterpart of sampling $\mathbf{m} = [\texttt{Prompt}\cdots]$, namely, we generate a reverse-time dynamics of

$$|\boldsymbol{\psi}_t\rangle = \sqrt{p_t(1)}\Big|\underbrace{\texttt{Prompt 1}}_{\text{fixed}}\ \underbrace{\cdots_t}_{\text{generated}}\Big\rangle + \sqrt{p_t(2)}\Big|\underbrace{\texttt{Prompt 2}}_{\text{fixed}}\ \underbrace{\cdots_t}_{\text{generated}}\Big\rangle. \tag{15}$$

The part $[\cdots_{t=1}]$ is initialized by sampling from the terminal probability $p_{t=1}$. After running reverse-time unraveling procedure defined in Theorem D.10 from $t = 1$ to $t \approx 0$, we apply the projective observation process to $|\boldsymbol{\psi}_{t\approx 0}\rangle$ and make a single sentence $\mathbf{m} \sim [p_{t\approx 0}(1), p_{t\approx 0}(2)]$. Afterwards, we apply usual denoising process optionally[2].

Superposition of Thoughts (SoT). We consider Superposition of Thoughts (SoT), an enhanced version of SoP, in which the prompt portion of SoP is replaced with reasoning tokens. As an application of the unraveling with text superposition states, we can construct superposition of reasoning/thought tokens, i.e.,

$$\texttt{Prompt} \to \sqrt{p_t(1)}\Big|\underbrace{\texttt{<think>}\ \texttt{Thought 1}\ \texttt{</think>}}_{\text{fixed}}\ \underbrace{\cdots_t}_{\text{generated}}\Big\rangle + \sqrt{p_t(2)}\Big|\underbrace{\texttt{<think>}\ \texttt{Thought 2}\ \texttt{</think>}}_{\text{fixed}}\ \underbrace{\cdots_t}_{\text{generated}}\Big\rangle.$$

The sampling procedure is completely same as SoP, but it is expected to generate "mixed thought" outputs if the model supports reasoning mode. Depending on the model, the user's prompt may or may not be included in the text generated by the diffusion model. For example, iLLaDA (Nie et al. (2026)) treats user's prompt as a preconditioned sentences in generated sequences, but DiffusionGemma (DiffusionGemma Team et al. (2026)) does not.

Quantum-Inspired Remasking (QIR). Remasking is considered a technique for improving the quality of generated sentences in mask-based diffusion language models. Reconsidering this from

[2]If we use masked diffusion, the mask token `[m]` sometimes left behind. To clear it, we apply final denoizing step that jumps `[m]` to certain meaningful tokens.

the perspective of open quantum systems, "remasking" can be viewed as introducing a form of temperature when the mask token [m] is treated as the ground state; specifically, this corresponds to adding—in addition to the transition $[\mathtt{m}] \neq i \to [\mathtt{m}]$—a small-scale jump operator defined in Equation (13) that makes $[\mathtt{m}] \to i \neq [\mathtt{m}]$ parameterized by $0 < \gamma \ll 1$ to the forward process.

From Equation (13), for the unraveling that occurs when such a "perturbation" is applied, it is necessary to calculate the following values.

$$\frac{p_t(\mathbf{m}_{p\to 0})}{p_t(\mathbf{m}_{p\to i})} = \left(\frac{p_t(\mathbf{m}_{p\to i})}{p_t(\mathbf{m}_{p\to 0})} \right)^{-1}. \tag{16}$$

The highlighted ratio measures how strongly the model favors restoring the current token i when position p is masked. We estimate this score by applying the pretrained model to $\mathbf{m}_{p\to 0}$ and use its reciprocal to determine the remasking rate. This approximation neglects the change in the score induced by the perturbation, which we assume to be small for sufficiently small γ. Thus, tokens that the model is less confident in reconstructing from their current context are more likely to be remasked. The perturbation therefore yields a confidence-based remasking mechanism.

4 Experiments

4.1 Experiment 1: Sentence Generation with Text Superposition

In this section, we examine whether the superposition of prompts (SoP) and thoughts (SoT) manifests as a semantic blending in the generated text. The goal of these experiments is not to outperform existing conditional generation methods, but rather to validate the fundamental property of the proposed quantum-inspired sampling: that a superposition of input states (prompts or thoughts) can indeed lead to the generation of a 'mixed' output state that is not simply one or the other, but a probabilistic blend of both.

Setup. We use masked diffusion for the experiments in this section[3]. The input prompts are qualitatively designed to reflect their respective superposition targets: **For SoP**, we employ descriptive prompts that define environmental settings to observe the superposition of semantic contexts. **For SoT**, we utilize prescriptive prompts that impose stylistic and instructional constraints to investigate the superposition of reasoning-driven generation strategies. In this section, we show results on two sets of **forest-snow** prompt pairs. See `story__forest_snow` in Table 3 for SoP and `story_reasoning__forest_snow` in Table 4 for SoT for details.

Implementation. In the superposition-based unraveling process, each sentence undergoes not only denoising but also updates to the weight coefficients $p(i)$. The temporal evolution of these coefficients can occur rapidly depending on the output of the trained model. In such cases, the effect of superposition may disappear early in the denoising process. (See Appendix F.4 for details.) To mitigate this effect—while slightly departing from theoretical principles—we implement a regularization scheme that softens the coefficient updates at each iteration: $c(i)_p \to c(i)_p^{\beta}$.

Results. Figure 3 provides a visual overview of the generated sentences' distribution. The scatter plots indicate that the **SoP**-generated sentences tend to occupy a region intermediate to the single-prompt outputs. This qualitative observation is quantitatively supported by the average similarity values presented in Table 5, which show a statistically significant increase in similarity to the non-selected condition compared to single-prompt generation. To qualitatively assess the effect of **SoT**, we present a representative generated sample in Table 1. The generated text effectively synthesally fuses the divergent stylistic constraints of both "**forest**" and "**snow**" prompts into a single narrative. This supports our hypothesis that the superposition of thought tokens allows the model to navigate between conflicting reasoning paths during the denoising process.

[3] In our experiments, uniform diffusion exhibits faster concentration of superposition weights onto a single component than masked diffusion, suggesting that superpositions are less stable under uniform diffusion. See Section F.4 for details.

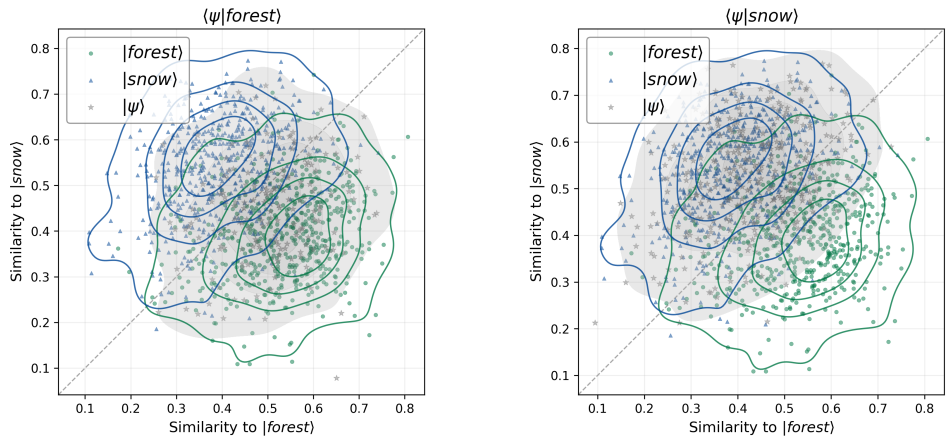


Figure 3: Cosine similarity of SEDD-small outputs to forest-only (horizontal) and snow-only (vertical) reference texts, computed using MiniLM embeddings. Green and blue show single-condition baselines; gray shows **SoP** outputs with forest (left) or snow (right) selected at final measurement. Superposition increases average similarity to the unselected condition.

Prompt	Write a short story.	
Thoughts	**[forest]** The narrative must take place in a dense, vibrant summer forest. [...] The tone should feel alive, earthy, and teeming with natural energy.	**[snow]** The narrative must take place in a desolate, frozen winter tundra. [...] The tone should feel silent, harsh, and starkly isolated.
Generated ⟨ψ|**forest**⟩	As the overcliff dawn drifts from **sprawling forest of trees north** – I stand on the edge of the **frozen world**, adrift. [...] The air itself toys him away, like an elemental furnace or **pine-scented forest earth**. **Snow** piling up over the world, carrying **liquid silver**, and where the sun's frame unfurls a **chill**; dimming.	

Table 1: Representative generated samples by iLLaDA-8B-Instruct (Nie et al. (2026)) with **SoT**. Coloring in the generated texts is generated by gemma-4-26B-A4B (Gemma Team et al. (2026)).

4.2 Experiment 2: Quantum-Inspired Remasking

We evaluate the quantum-inspired remasking (QIR) method introduced in the previous section. We apply this method to pretrained diffusion language models without updating their weights. By varying when remasking begins during generation, we examine how its timing affects the quality of the generated text using several evaluation metrics.

Setup. We evaluate SEDD-medium (Lou et al., 2024), RADD-medium (Ou et al., 2025), and iLLaDA-8B-Base (Nie et al., 2026). We generate 256 unconditional sequences per condition for SEDD-medium and RADD-medium, and 128 for iLLaDA-8B-Base, with 1024 tokens, 1024 sampling steps, and $\gamma = 0.02$. To reduce the cost of evaluating the reciprocal scores in Equation (16), we uniformly sample up to $K = 8$ non-mask positions per remasking calculation. Remasking begins after approximately 10%, 50%, or 90% of sampling (`start`, `middle`, and `end`) and remains active thereafter. The baseline, `none`, uses the same stochastic-unraveling decoder without remasking. We verified that, without remasking, our implementation yields results consistent with those of the original SEDD Euler sampler under the same sampling settings.

Metrics. We report generative perplexity (GenPPL), computed by evaluating the generated texts with GPT-2-large (Radford et al., 2019), token-frequency entropy, and a CoLA acceptability score (Morris et al., 2020). Entropy monitors excessive loss of token diversity; it is not a quantity to minimize. Repetition is measured by ΔR_n, the same-seed difference from no remasking in repeated token n-gram fractions, averaged over 128-token windows and expressed in percentage points (pp). We evaluate $n = 2, \ldots, 8$ and report $n = 7$ as a representative example in Table 2. Qwen3-32B compares each sample with its same-checkpoint, same-seed baseline for fluency, coherence, progression, and repetition, without knowing the generation method. Wins, ties, and losses score

Model	Metric	none	end	middle	start
SEDD(medium) Lou et al. (2024)	GenPPL	$33.12_{\pm 0.58}$	$32.73_{\pm 0.57}$	$27.52_{\pm 0.43}$	$12.24_{\pm 0.26}$
	Entropy	$7.57_{\pm 0.02}$	$7.57_{\pm 0.02}$	$7.52_{\pm 0.03}$	$6.42_{\pm 0.05}$
	CoLA	$0.285_{\pm 0.006}$	$0.285_{\pm 0.006}$	$0.335_{\pm 0.006}$	$0.336_{\pm 0.007}$
	Qwen (%)	–	$51.5_{\pm 0.8}$	$\mathbf{62.0}_{\pm 1.3}$	$53.3_{\pm 1.9}$
	ΔR_7 (pp)	$0.000_{\pm 0.000}$	$-0.005_{\pm 0.004}$	$0.025_{\pm 0.013}$	$1.405_{\pm 0.210}$
RADD(medium) Ou et al. (2025)	GenPPL	$33.66_{\pm 0.60}$	$33.34_{\pm 0.59}$	$28.08_{\pm 0.44}$	$13.71_{\pm 0.28}$
	Entropy	$7.59_{\pm 0.02}$	$7.58_{\pm 0.02}$	$7.53_{\pm 0.02}$	$6.55_{\pm 0.05}$
	CoLA	$0.281_{\pm 0.005}$	$0.282_{\pm 0.005}$	$0.316_{\pm 0.006}$	$0.328_{\pm 0.007}$
	Qwen (%)	–	$49.5_{\pm 0.8}$	$\mathbf{62.4}_{\pm 1.3}$	$58.0_{\pm 1.8}$
	ΔR_7 (pp)	$0.000_{\pm 0.000}$	$0.002_{\pm 0.007}$	$0.007_{\pm 0.010}$	$1.258_{\pm 0.333}$
iLLaDA-8B-Base Nie et al. (2026)	GenPPL	$21.14_{\pm 1.56}$	$21.25_{\pm 1.57}$	$19.83_{\pm 1.48}$	$18.19_{\pm 6.58}$
	Entropy	$6.46_{\pm 0.08}$	$6.46_{\pm 0.08}$	$6.43_{\pm 0.07}$	$5.70_{\pm 0.09}$
	CoLA	$0.460_{\pm 0.019}$	$0.466_{\pm 0.019}$	$0.487_{\pm 0.018}$	$0.563_{\pm 0.019}$
	Qwen (%)	–	$50.0_{\pm 1.4}$	$\mathbf{61.8}_{\pm 1.9}$	$55.7_{\pm 2.5}$
	ΔR_7 (pp)	$0.000_{\pm 0.000}$	$-0.072_{\pm 0.054}$	$-0.081_{\pm 0.113}$	$4.042_{\pm 0.531}$

Table 2: Results at $\gamma = 0.02$. Each condition contains 256 sequences for SEDD and RADD and 128 for iLLaDA. The no-remasking column is the shared control. All entries give mean $\pm$ standard error; Qwen preference scores are expressed as percentages, and ΔR_7 in percentage points (pp). For ΔR_7, the no-remasking baseline is zero by definition.

1, $1/2$, and 0, averaged over both presentation orders; preference above 50% favors remasking. Definitions and evaluation details are given in Appendices G.2 and G.3.

Results. Enabling QIR from the middle of generation yields Qwen preference scores of 62.0% for SEDD-medium, 62.4% for RADD-medium, and 61.8% for iLLaDA-8B-Base against their no-remasking baselines (Table 2). This schedule achieves the highest mean preference among the evaluated schedules for both models, demonstrating improved judge-assessed text quality without retraining. Starting remasking earlier further lowers perplexity and raises CoLA scores, but reduces judge preference and token diversity while increasing repetition. The increase in repetition under `start` is also observed for the other evaluated n-gram lengths. These results show that lower generative perplexity and higher grammatical acceptability do not necessarily translate into better overall text quality. Supplementary repetition results are provided in Appendix G.4.

5 Conclusion and Outlook

We embedded continuous-time diffusion language models into Lindblad dynamics, enabling sampling from superposed conditioning states. In the forest/snow experiment, superposition outputs show higher similarity to the unselected condition than single-condition baselines, suggesting that its influence persists after final measurement.

Within this framework, we introduced quantum-inspired remasking through perturbations of the jump operators. Experiments using pretrained models show that the timing of remasking affects generated text quality: earlier remasking lowers perplexity but can increase repetition, whereas an intermediate onset achieves the highest mean judge preference across the tested checkpoints. These results highlight the importance of evaluating text quality beyond perplexity alone.

Future work includes reducing the computational cost of remasking and extending its evaluation to conditional generation. More broadly, this framework opens a path toward quantum-native language models in which superposition and interference play an active role in generation. Such models could evolve multiple candidate sequences coherently, with interference shaping the probabilities of the resulting text. Exploring how these dynamics can be designed and learned may lead to new generative mechanisms beyond classical diffusion.

AI USE STATEMENT

Generative AI tools were used for brainstorming, improving the readability and organization of the manuscript, and assisting with software implementation. Suggestions produced by these tools were independently examined and verified by the authors. Generative AI tools were not used as an autonomous source of scientific claims, proofs, or result interpretation. The authors take full responsibility for the final content of this work.

REFERENCES

Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum boltzmann machine. *Physical Review X*, 8(2):021050, 2018. doi: 10.1103/PhysRevX.8.021050.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems 34*, pp. 17981–17993, 2021. https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019. doi: 10.1088/2058-9565/ab4eb5.

Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549:195–202, 2017. doi: 10.1038/nature23474.

H.-P. Breuer and F. Petruccione. *The Theory of Open Quantum Systems*. Oxford University Press, 2002. doi: 10.1093/acprof:oso/9780199213900.001.0001.

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.

H. J. Carmichael. *An Open Systems Approach to Quantum Optics*. Springer, 1993. doi: 10.1007/978-3-540-47650-7.

Jean Dalibard, Yvan Castin, and Klaus Mølmer. Wave-function approach to dissipative processes in quantum optics. *Physical Review Letters*, 68(5):580–583, 1992. doi: 10.1103/PhysRevLett.68.580.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems 34*, pp. 8780–8794, 2021. https://papers.nips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

DiffusionGemma Team, Adrien Ali Taïga, James Assiene, Daniele Calandriello, Rahma Chaabouni, João Gante, Tamara von Glehn, Nate Keating, Chris Knutsen, Martin Kukla, et al. Diffusiongemma technical report. *arXiv preprint arXiv:2608.00146*, 2026. https://arxiv.org/abs/2608.00146.

R. Dum, P. Zoller, and H. Ritsch. Monte carlo simulation of the atomic master equation for spontaneous emission. *Physical Review A*, 45(7):4879–4887, 1992. doi: 10.1103/PhysRevA.45.4879.

Richard P. Feynman and A. R. Hibbs. *Quantum Mechanics and Path Integrals*. McGraw-Hill, 1965.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. https://arxiv.org/abs/2607.02770.

V. Gorini, A. Kossakowski, and E. C. G. Sudarshan. Completely positive dynamical semigroups of n-level systems. *Journal of Mathematical Physics*, 17(5):821–825, 1976. doi: 10.1063/1.522979.

Yuji Hirono, Akinori Tanaka, and Kenji Fukushima. Understanding diffusion models by feynman's path integral. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 18324–18351. PMLR, 2024a. URL https://proceedings.mlr.press/v235/hirono24a.html.

Yuji Hirono, Akinori Tanaka, and Kenji Fukushima. Understanding diffusion models by feynman's path integral. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 18324–18351, 2024b. https://proceedings.mlr.press/v235/hirono24a.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Deep unsupervised learning using nonequilibrium thermodynamics. In *Advances in Neural Information Processing Systems 33*, pp. 6840–6851, 2020. https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems 35*, pp. 8633–8646, 2022. https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html.

Feng Hong, Geng Yu, Yushi Ye, Haicheng Huang, Huangjie Zheng, Ya Zhang, Yanfeng Wang, and Jiangchao Yao. Wide-in, narrow-out: Revokable decoding for efficient and effective dllms. In *International Conference on Learning Representations*, volume 2026, pp. 40132–40150, 2026.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forr'e, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems 34*, pp. 12454–12465, 2021. https://proceedings.neurips.cc/paper/2021/hash/67d96d458abdef21792e6d8e590244e7-Abstract.html.

Kotaro Ikeda, Tomoya Uda, Daisuke Okanohara, and Sosuke Ito. Speed-accuracy trade-off for the diffusion models: Wisdom from nonequilibrium thermodynamics and optimal transport. *Physical Review X*, 2024. https://doi.org/10.1103/x5vj-8jq9.

Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion. *ArXiv*, abs/2506.17298, 2025. https://api.semanticscholar.org/CorpusID:280000358.

Hyukjoon Kwon and Myungshik S. Kim. Fluctuation theorems for a quantum channel. *Physical Review X*, 2018. https://api.semanticscholar.org/CorpusID:201317121.

Göran Lindblad. On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics*, 48(2):119–130, 1976. doi: 10.1007/BF01608499.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 32819–32848, 2024. https://proceedings.mlr.press/v235/lou24a.html.

Daniel Manzano. A short introduction to the lindblad master equation. *AIP Advances*, 2019. URL https://api.semanticscholar.org/CorpusID:184487806.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 119–126, 2020.

Ryota Nasu, Gota Tanaka, and Asato Tsuchiya. Quantum reversibility meets classical reverse diffusion, 2025. URL https://arxiv.org/abs/2510.18512.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Jirong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems 38*, 2025. https://openreview.net/forum?id=KnqiC0znVF.

Shen Nie, Qiyang Min, Shaoxuan Xu, Zihao Huang, Yuxuan Song, Yong Shan, Yankai Lin, Wayne Xin Zhao, Chongxuan Li, and Ji-Rong Wen. Improved large language diffusion models, 2026. URL https://arxiv.org/abs/2606.25331.

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. In *Proceedings of the 13th International Conference on Learning Representations*, 2025. https://openreview.net/forum?id=sMyXP8Tanm.

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.

Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022. https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 52584–52619, 2025. https://proceedings.mlr.press/v267/sahoo25a.html.

Uriel Singer, Adam Polyak, Thomas Hayes, Xiaoyue Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *Proceedings of the 11th International Conference on Learning Representations*, 2023. https://openreview.net/forum?id=nJfylDvgzlq.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2256–2265, 2015. https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32*, 2019. https://papers.nips.cc/paper_files/paper/2019/hash/3001ef257407d5a371a96dcd947c7d93-Abstract.html.

Yang Song, Jascha Narain Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the 9th International Conference on Learning Representations*, 2021b. https://openreview.net/forum?id=PxTIG12RRHS.

Yuxuan Song, Zheng Zhang, Cheng Luo, Peng Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, Yuwei Fu, Jing Su, Ge Zhang, Wenhao Huang, Mingxuan Wang, Lin Yan, Xiaoying Jia, Jingjing Liu, Weizhi Ma, Yaqing Zhang, Yong-Xu Wu, and Hao-Min Zhou. Seed diffusion: A large-scale diffusion language model with high-speed inference. *ArXiv*, abs/2508.02193, 2025. https://api.semanticscholar.org/CorpusID:280421264.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *Proceedings of the 11th International Conference on Learning Representations*, 2023. https://openreview.net/forum?id=BYWWwSY2G5s.

Guanghan Wang, Yair Schiff, Subham Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *Advances in Neural Information Processing Systems*, 38: 147282–147339, 2026.

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. In *Proceedings of the 13th International Conference on Learning Representations*, 2025. https://openreview.net/forum?id=sL2F9YCMXf.

Manato Yaguchi, Kotaro Sakamoto, Ryosuke Sakamoto, Masato Tanabe, Masatomo Akagawa, Yusuke Hayashi, Masahiro Suzuki, and Yutaka Matsuo. The geometry of phase transitions in diffusion models: Tubular neighbourhoods and singularities. *Transactions on Machine Learning Research*, 2025. https://openreview.net/forum?id=ahVFKFLYk2.

Zhendong Yu and Haiping Huang. Nonequilbrium physics of generative diffusion models. *Physical Review E*, 2024. https://doi.org/10.1103/PhysRevE.111.014111.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. In *Advances in Neural Information Processing Systems 38*, 2025. https://openreview.net/forum?id=7ZVRlBFuEv.

A NOTATIONS

This section summarizes the notation used throughout this paper.

Diffusion Model In discrete language diffusion models, we use the following notation:

- Time-forward quantities are denoted without tildes ($a, b, c, \dots$), while time-backward quantities are denoted with tildes ($\tilde{a}, \tilde{b}, \tilde{c}, \dots$).
- $\mathbb{V} = \{m \mid m = 0, 1, \dots, V\}$ denotes the vocabulary set.
- $\mathbb{V}^N$ denotes the set of all possible token sequences of fixed length N:

$$\mathbb{V}^N = \{\mathbf{m} = [m_1, m_2, \dots, m_N] \,|\, m_p \in \mathbb{V} \text{ for } p = 1, 2, \dots, N\}.$$

Each element $\mathbf{m} \in \mathbb{V}^N$ corresponds to a discrete sentence or token sequence.

Mathematical Notation We also introduce the following shorthand:

- The Kronecker delta (indicator function) is defined as

$$1_{mn} = \begin{cases} 1 & m = n, \\ 0 & \text{otherwise}, \end{cases} \qquad m, n \in \mathbb{V}.$$

- The delta function excluding the p-th component is

$$1_{\mathbf{m}_{-p}\mathbf{n}_{-p}} = \prod_{\substack{q \in [N] \\ q \neq p}} 1_{m_q n_q}, \qquad \mathbf{m}, \mathbf{n} \in \mathbb{V}^N.$$

- $\mathbf{n}_{p \to m}$ denotes the sequence obtained by replacing the p-th token of $\mathbf{n}$ with m:

$$\mathbf{n}_{p \to m} = [n_1, \dots, n_{p-1}, m, n_{p+1}, \dots, n_N].$$

Open Quantum System Formulation For the quantum-mechanical formulation, we use:

- Operators $A, B, C, \dots$, with backward-time operators denoted by $\tilde{A}, \tilde{B}, \tilde{C}, \dots$.
- The single-token Hilbert space

$$\mathcal{H}_{\mathbb{V}} = \left\{ \sum_{m \in \mathbb{V}} c_m \,|m\rangle \,\middle|\, c_m \in \mathbb{C} \right\}.$$

- The Hilbert space for token sequences of fixed length N

$$\mathcal{H} = \left\{ \sum_{\mathbf{m} \in \mathbb{V}^N} c_{\mathbf{m}} \,|\mathbf{m}\rangle \,\middle|\, c_{\mathbf{m}} \in \mathbb{C} \right\} = \mathcal{H}_{\mathbb{V}}^{\otimes N}.$$

B QUICK TOUR ON DIFFUSION LANGUAGE MODELS

This section provides a self-contained review of the continuous-time formulation of diffusion language models used in Section 3. We first recall finite-state Markov dynamics, its time reversal, and the transition kernels induced by infinitesimal generators. We then specialize this framework to the factorized transition structure used in language diffusion models and explain how this structure motivates τ-leaping sampling.

B.1 DIFFUSION PROCESS

We consider probability distributions $p(\mathbf{m})$ on the finite state space $\mathbb{V}^N$.

Definition B.1: Markov Kernel A Markov kernel on $\mathbb{V}^N$ is a matrix $T(\mathbf{m}, \mathbf{n})$ satisfying

$$T(\mathbf{m}, \mathbf{n}) \geq 0, \qquad \sum_{\mathbf{m} \in \mathbb{V}^N} T(\mathbf{m}, \mathbf{n}) = 1 \quad \text{for every } \mathbf{n} \in \mathbb{V}^N.$$

It acts on a probability distribution p as

$$(Tp)(\mathbf{m}) = \sum_{\mathbf{n} \in \mathbb{V}^N} T(\mathbf{m}, \mathbf{n}) p(\mathbf{n}),$$

yielding another probability distribution.

A continuous-time diffusion process on $\mathbb{V}^N$ is described by

$$\frac{d}{dt} p_t = L_t p_t,$$

where L_t is a time-dependent linear operator on $\mathbb{R}^{|\mathbb{V}^N|}$. The evolution must preserve non-negativity and normalization for every initial probability distribution. Equivalently, for any $s \leq t$, the solution can be written as $p_t = T_{t|s} p_s$, where $T_{t|s}$ is a Markov kernel.

Theorem B.2: Diffusion Process Consider the linear evolution equation

$$\frac{d}{dt} p_t(\mathbf{m}) = (L_t p_t)(\mathbf{m}) = \sum_{\mathbf{n} \in \mathbb{V}^N} Q_t(\mathbf{m}, \mathbf{n}) p_t(\mathbf{n}). \tag{17}$$

Suppose that Q_t is continuous in t and that the evolution from any time s to any later time t is represented by a Markov kernel $T_{t|s}$ as defined in Definition B.1. Then

$$Q_t(\mathbf{m}, \mathbf{n}) \geq 0 \quad (\mathbf{m} \neq \mathbf{n}), \qquad \sum_{\mathbf{m} \in \mathbb{V}^N} Q_t(\mathbf{m}, \mathbf{n}) = 0. \tag{18}$$

Proof

Fix a time t and a state $\mathbf{k}$, and choose the initial distribution $p_t(\mathbf{m}) = 1_{\mathbf{mk}}$. For $h > 0$, the evolution equation gives

$$(T_{t+h|t} p_t)(\mathbf{m}) = 1_{\mathbf{mk}} + h Q_t(\mathbf{m}, \mathbf{k}) + o(h). \tag{19}$$

Since $T_{t+h|t}$ is a Markov kernel, the left-hand side is nonnegative. For $\mathbf{m} \neq \mathbf{k}$, this implies

$$0 \leq h Q_t(\mathbf{m}, \mathbf{k}) + o(h).$$

Dividing by h and taking $h \to 0^+$ yields $Q_t(\mathbf{m}, \mathbf{k}) \geq 0$.
Similarly, preservation of normalization gives

$$1 = \sum_{\mathbf{m} \in \mathbb{V}^N} (T_{t+h|t} p_t)(\mathbf{m}) = 1 + h \sum_{\mathbf{m} \in \mathbb{V}^N} Q_t(\mathbf{m}, \mathbf{k}) + o(h). \tag{20}$$

Subtracting 1, dividing by h, and taking $h \to 0^+$ yields $\sum_{\mathbf{m}} Q_t(\mathbf{m}, \mathbf{k}) = 0$. Since $\mathbf{k}$ was arbitrary, both conditions follow. □

We now consider the time reversal of the diffusion process in Equation (17). Reversing the direction of time gives

$$-\frac{d}{dt} p_t(\mathbf{m}) = -\sum_{\mathbf{n} \in \mathbb{V}^N} Q_t(\mathbf{m}, \mathbf{n}) p_t(\mathbf{n}). \tag{21}$$

Although $-Q_t$ reproduces this evolution of p_t, it is not generally a valid Markov generator: its off-diagonal entries are negative wherever those of Q_t are positive. A valid reverse-time generator is instead obtained by reversing each transition and weighting its rate by a probability ratio, as follows.

Theorem B.3: Time-Reversed Diffusion Process Let p_t solve Equation (17), and suppose that $p_t(\mathbf{n}) > 0$ for every $\mathbf{n}$ on the time interval considered. Define

$$\tilde{Q}_t(\mathbf{m}, \mathbf{n}) = Q_t(\mathbf{n}, \mathbf{m}) \frac{p_t(\mathbf{m})}{p_t(\mathbf{n})} \quad (\mathbf{m} \neq \mathbf{n}), \qquad \tilde{Q}_t(\mathbf{m}, \mathbf{m}) = -\sum_{\mathbf{n} \neq \mathbf{m}} \tilde{Q}_t(\mathbf{n}, \mathbf{m}). \tag{22}$$

Then $\tilde{Q}_t$ satisfies the Markov generator conditions in Equation (18), and

$$-\frac{d}{dt} p_t(\mathbf{m}) = \sum_{\mathbf{n} \in \mathbb{V}^N} \tilde{Q}_t(\mathbf{m}, \mathbf{n}) p_t(\mathbf{n}). \tag{23}$$

Thus, the reverse process follows the same family of distributions p_t in the direction of decreasing t.

Proof

We rewrite the right-hand side of the original equation:

$$-\sum_{\mathbf{n}} Q_t(\mathbf{m}, \mathbf{n})\, p_t(\mathbf{n}) = -Q_t(\mathbf{m}, \mathbf{m}) p_t(\mathbf{m}) - \sum_{\mathbf{n} \neq \mathbf{m}} Q_t(\mathbf{m}, \mathbf{n}) p_t(\mathbf{n}). \tag{24}$$

Using the column-sum condition

$$\sum_{\mathbf{n}} Q_t(\mathbf{n}, \mathbf{m}) = 0, \tag{25}$$

we have

$$Q_t(\mathbf{m}, \mathbf{m}) = -\sum_{\mathbf{n} \neq \mathbf{m}} Q_t(\mathbf{n}, \mathbf{m}), \tag{26}$$

and therefore

$$-Q_t(\mathbf{m}, \mathbf{m}) p_t(\mathbf{m}) = \left(\sum_{\mathbf{n} \neq \mathbf{m}} Q_t(\mathbf{n}, \mathbf{m}) \right) p_t(\mathbf{m}). \tag{27}$$

Hence,

$$-\sum_{\mathbf{n}} Q_t(\mathbf{m}, \mathbf{n})\, p_t(\mathbf{n}) = \sum_{\mathbf{n} \neq \mathbf{m}} Q_t(\mathbf{n}, \mathbf{m}) p_t(\mathbf{m}) - \sum_{\mathbf{n} \neq \mathbf{m}} Q_t(\mathbf{m}, \mathbf{n}) p_t(\mathbf{n}). \tag{28}$$

The first term can be rewritten as

$$Q_t(\mathbf{n}, \mathbf{m}) p_t(\mathbf{m}) = \left(Q_t(\mathbf{n}, \mathbf{m}) \frac{p_t(\mathbf{m})}{p_t(\mathbf{n})} \right) p_t(\mathbf{n}), \tag{29}$$

so that

$$-\sum_{\mathbf{n}} Q_t(\mathbf{m}, \mathbf{n})\, p_t(\mathbf{n}) = \sum_{\mathbf{n} \neq \mathbf{m}} \left[Q_t(\mathbf{n}, \mathbf{m}) \frac{p_t(\mathbf{m})}{p_t(\mathbf{n})} \right] p_t(\mathbf{n}) - \sum_{\mathbf{n} \neq \mathbf{m}} Q_t(\mathbf{m}, \mathbf{n}) p_t(\mathbf{n}). \tag{30}$$

Now we define, for $\mathbf{m} \neq \mathbf{n}$,

$$\tilde{Q}_t(\mathbf{m}, \mathbf{n}) := Q_t(\mathbf{n}, \mathbf{m}) \frac{p_t(\mathbf{m})}{p_t(\mathbf{n})}, \tag{31}$$

and set the diagonal part so that the column sum vanishes:

$$\tilde{Q}_t(\mathbf{m}, \mathbf{m}) := -\sum_{\mathbf{k} \neq \mathbf{m}} \tilde{Q}_t(\mathbf{k}, \mathbf{m}). \tag{32}$$

Then the second term above is absorbed into the diagonal contribution, and we obtain

$$-\sum_{\mathbf{n}} Q_t(\mathbf{m}, \mathbf{n})\, p_t(\mathbf{n}) = \sum_{\mathbf{n}} \tilde{Q}_t(\mathbf{m}, \mathbf{n})\, p_t(\mathbf{n}). \tag{33}$$

This proves Equation (23). □

B.2 SAMPLING

B.2.1 SAMPLING 1-TOKEN PER STEP

In practice, directly evolving the full distribution p_t is computationally intractable because the state space grows exponentially with the sequence length. Instead, we simulate individual sample trajectories using transition probabilities derived from the forward or reverse evolution equations in Equations (17) and (23). For example, to first order in dt, Equation (17) yields

$$p_{t+dt}(\mathbf{m}) = p_t(\mathbf{m}) + dt \sum_{\mathbf{n}\in\mathbb{V}^N} Q_t(\mathbf{m},\mathbf{n}) p_t(\mathbf{n}) = \sum_{\mathbf{n}\in\mathbb{V}^N} \left(\underbrace{1_{\mathbf{mn}} + Q_t(\mathbf{m},\mathbf{n})dt}_{p_{t+dt|t}(\mathbf{m}|\mathbf{n})} \right) p_t(\mathbf{n}). \tag{34}$$

Similarly, to first order in dt, the reverse-time evolution in Equation (23) yields

$$p_{t-dt}(\mathbf{m}) = p_t(\mathbf{m}) + dt \sum_{\mathbf{n}\in\mathbb{V}^N} \tilde{Q}_t(\mathbf{m},\mathbf{n}) p_t(\mathbf{n}) = \sum_{\mathbf{n}\in\mathbb{V}^N} \left(\underbrace{1_{\mathbf{mn}} + \tilde{Q}_t(\mathbf{m},\mathbf{n})dt}_{\tilde{p}_{t-dt|t}(\mathbf{m}|\mathbf{n})} \right) p_t(\mathbf{n}). \tag{35}$$

In practice, samples are generated using the learned reverse-time generator $\tilde{Q}_t^{\text{model}}$, with transition probabilities given by

$$\tilde{p}^{\text{model}}_{t-dt|t}(\mathbf{m}|\mathbf{n}) = 1_{\mathbf{mn}} + \tilde{Q}_t^{\text{model}}(\mathbf{m},\mathbf{n})dt. \tag{36}$$

B.2.2 FORWARD SOLUTION

The sampling schemes above are accurate to first order in dt. However, they are inefficient for language generation because the generator permits only single-token transitions:

$$Q_t(\mathbf{m},\mathbf{n}) = \sigma(t) \sum_{p=1}^{N} 1_{\mathbf{m}_{-p}\mathbf{n}_{-p}} Q^{\text{tok}}(m_p, n_p), \quad 1_{\mathbf{m}_{-p}\mathbf{n}_{-p}} = \begin{cases} 1 & \mathbf{m} = \mathbf{n} \text{ except for } m_p, n_p, \\ 0 & \text{otherwise,} \end{cases} . \tag{37}$$

Consequently, each update changes at most one token, requiring many sequential steps to generate a sequence. To develop more efficient sampling schemes, we first derive the solution of the forward diffusion equation.

Theorem B.4 With the diffusion matrix in Equation (37), we get the formal solution of the diffusion equation

$$p_{t|s}(\mathbf{m}|\mathbf{n}) = \prod_{p=1}^{N} p_{t|s}(m_p|n_p), \tag{38}$$

where

$$p_{t|s}(m_p|n_p) = \exp\left[(\bar{\sigma}(t) - \bar{\sigma}(s))Q^{\text{tok}}\right](m_p, n_p), \quad \bar{\sigma}(t) = \int_0^t \sigma(\tau)d\tau. \tag{39}$$

Proof

The decomposition of $Q_t(\mathbf{m},\mathbf{n})$ in Equation (37) means it defines matrix

$$Q_t = \sigma(t) \sum_{p=1}^{N} I \otimes \cdots \overset{\text{p-th}}{Q^{\text{tok}}} \cdots \otimes I. \tag{40}$$

Because the time dependency is only on the scaler function $\sigma(t)$, two matrices Q_t and Q_s commutes for any t, s, which means we can solve the ODE by taking exponential:

$$p_{t|s}(\mathbf{m}|\mathbf{n}) = \left[\exp\left(\int_s^t Q_\tau d\tau\right)\right](\mathbf{m},\mathbf{n}) = \left[\exp\left(\underbrace{(\bar{\sigma}(t)-\bar{\sigma}(s))}_{=:\Delta\bar{\sigma}(t|s)}\sum_{p=1}^{N} I\otimes\cdots\overset{\text{p-th}}{Q^{\text{tok}}}\cdots\otimes I\right)\right](\mathbf{m},\mathbf{n}). \tag{41}$$

Now, it is well known

$$\exp(A\otimes I + I\otimes B) = \exp(A)\otimes\exp(B), \tag{42}$$

for any square matrices A, B, and applying it recursively, we have

$$\begin{aligned} &p_{t|s}(\mathbf{m}|\mathbf{n}) \\ &= \left[\exp\left(\Delta\bar{\sigma}(t|s)\overset{p=1}{Q^{\text{tok}}}\right)\otimes\exp\left(\Delta\bar{\sigma}(t|s)\overset{p=2}{Q^{\text{tok}}}\right)\otimes\cdots\otimes\exp\left(\Delta\bar{\sigma}(t|s)\overset{p=N}{Q^{\text{tok}}}\right)\right](\mathbf{m},\mathbf{n}) \\ &= \prod_{p=1}^{N}\exp\left[\Delta\bar{\sigma}(t|s)Q^{\text{tok}}\right](m_p,n_p), \end{aligned} \tag{43}$$

which proves the theorem. □

The theorem shows that the forward transition kernel factorizes across token positions. Approximating each token-level transition probability to first order in Δt gives

$$p_{t+\Delta t|t}(\mathbf{m}|\mathbf{n}) \approx \prod_{p=1}^{N}\left(\underbrace{1_{m_p n_p} + \sigma(t)Q^{\text{tok}}(m_p,n_p)\Delta t}_{p_{t+\Delta t|t}(m_p|n_p)}\right). \tag{44}$$

Given the current sequence $\mathbf{n}$, this factorized approximation allows all tokens m_p to be sampled independently, so that multiple tokens can change within a single time step. This motivates the τ-leaping scheme for reverse-time sampling introduced below.

B.2.3 τ-LEAPING SAMPLING

For the reverse process, the generator $\tilde{Q}_t$ depends on time through the probability ratios, so Theorem B.4 does not apply directly. Nevertheless, the forward-process construction motivates an approximate treatment of the reverse dynamics, leading to an interpretation analogous to Equation (44). Specifically, we approximate the reverse-time evolution as follows:

$$\tilde{p}^{\tau\text{-leap}}_{t-\Delta t|t}(\mathbf{m}|\mathbf{n}) := \prod_{p=1}^{N}\left(1_{m_p n_p} + \sigma(t)\underbrace{\tilde{Q}^{\text{tok}}_t(m_p,\mathbf{n})}_{\text{defined right}}\Delta t\right), \tag{45}$$

$$\tilde{Q}^{\text{tok}}_t(m_p,\mathbf{n}) := Q^{\text{tok}}(n_p,m_p)\frac{p_t(\mathbf{n}_{p\to m_p})}{p_t(\mathbf{n})}. \tag{46}$$

This distribution approximates the exact reverse-time transition, as can be seen by expanding it in Δt:

$$\begin{aligned} \tilde{p}^{\tau\text{-leap}}_{t-\Delta t|t}(\mathbf{m}|\mathbf{n}) &= 1_{\mathbf{mn}} + \sigma(t)\tilde{Q}_t(\mathbf{m},\mathbf{n})\Delta t + O(\Delta t^2) \\ &= \tilde{p}_{t-\Delta t|t}(\mathbf{m}|\mathbf{n}) + O(\Delta t^2). \end{aligned} \tag{47}$$

B.3 TRANSFORMATION OF THE $\mathbf{n}_0$-PREDICTOR INTO A SCORE ESTIMATOR

Some discrete diffusion models estimate the conditional probability of the clean state $\mathbf{n}_0$ given the corrupted state $\mathbf{n}$ at time t:

$$q_\theta(\mathbf{n}_0 \mid \mathbf{n}). \tag{48}$$

Examples include DiffusionGemma (DiffusionGemma Team et al., 2026), LLaDA (Nie et al., 2025), and iLLaDA (Nie et al., 2026). In this section, we briefly review this formulation and derive its relation to the score-based formulation in terms of s.

To relate the two formulations, consider the joint conditional distribution at times $t \geq s \geq 0$:

$$p_{t,s|0}(\mathbf{n}, \mathbf{m} \mid \mathbf{n}_0) = p_{t|s}(\mathbf{n} \mid \mathbf{m}) p_{s|0}(\mathbf{m} \mid \mathbf{n}_0). \tag{49}$$

The forward transition probabilities on the right-hand side can be evaluated using Theorem B.4. Conditioning on $\mathbf{n}$ and marginalizing over $\mathbf{n}_0$ using the denoising model in Equation (48) then yields an approximation to the reverse transition probability (Austin et al., 2021):

$$\tilde{q}_{\theta\, s|t}(\mathbf{m}|\mathbf{n}) \propto \sum_{\mathbf{n}_0} p_{t,s,0}(\mathbf{n}, \mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n}). \tag{50}$$

By setting $s = t - \Delta t$, the model yields the reverse conditional model

$$\begin{aligned} \tilde{q}_{\theta\, t-\Delta t|t}(\mathbf{m}|\mathbf{n}) &= 1_{\mathbf{mn}} + \sigma(t)\tilde{Q}_{\theta\, t}(\mathbf{m}, \mathbf{n})\Delta t + O(\Delta t^2) \\ &= 1_{\mathbf{mn}} + \sigma(t) \begin{cases} Q(\mathbf{n}, \mathbf{m}) s_\theta(\mathbf{n}, t)_{\mathbf{m}} \Delta t & (\mathbf{m} \neq \mathbf{n}) \\ -\sum_{\mathbf{k} \neq \mathbf{m}} Q(\mathbf{m}, \mathbf{k}) s_\theta(\mathbf{m}, t)_{\mathbf{k}} \Delta t & (\mathbf{m} = \mathbf{n}) \end{cases} + O(\Delta t^2). \end{aligned} \tag{51}$$

The piecewise expression above can equivalently be written as follows:

$$\tilde{q}_{\theta\, t-\Delta t|t}(\mathbf{m}|\mathbf{n}) = 1_{\mathbf{mn}} + \sigma(t)\Big(Q(\mathbf{n}, \mathbf{m}) s_\theta(\mathbf{n}, t)_{\mathbf{m}} - 1_{\mathbf{mn}} \sum_{\mathbf{k}} Q(\mathbf{m}, \mathbf{k}) s_\theta(\mathbf{n}, t)_{\mathbf{k}}\Big)\Delta t + O(\Delta t^2). \tag{52}$$

Theorem B.5: Transformation formula By ensuring the consistency between Equation (50) and Equation (52), the following holds:

$$s_\theta(\mathbf{n}, t)_{\mathbf{m}} = \frac{\sum_{\mathbf{n}_0} p_{t|0}(\mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n})}{\sum_{\mathbf{n}_0} p_{t|0}(\mathbf{n}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n})}. \tag{53}$$

Proof

First, we rewrite the right-hand side of Equation (50) by using Equation (49) as

$$\begin{aligned} \tilde{q}_{\theta\, s|t}(\mathbf{m}|\mathbf{n}) &\propto \sum_{\mathbf{n}_0} p_{t,s,0}(\mathbf{n}, \mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n}) \\ &= \sum_{\mathbf{n}_0} p_{t|s}(\mathbf{n}|\mathbf{m}) p_{s|0}(\mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n}). \end{aligned} \tag{54}$$

Substituting $s = t - \Delta t$ into each probability, we obtain:

$$\begin{aligned} p_{t|s}(\mathbf{n}|\mathbf{m}) &= p_{t|t-\Delta t}(\mathbf{n}|\mathbf{m}) \\ &= 1_{\mathbf{mn}} + \sigma(t) Q(\mathbf{n}, \mathbf{m})\Delta t + O(\Delta t^2), \end{aligned} \tag{55}$$

$$\begin{aligned} p_{s|0}(\mathbf{m}|\mathbf{n}_0) &= p_{t-\Delta t|0}(\mathbf{m}|\mathbf{n}_0) = p_{t|0}(\mathbf{m}|\mathbf{n}_0) - \frac{\partial}{\partial t} p_{t|0}(\mathbf{m}|\mathbf{n}_0)\Delta t + O(\Delta t^2) \\ &= p_{t|0}(\mathbf{m}|\mathbf{n}_0) - \sum_{\mathbf{k}} \sigma(t) Q(\mathbf{m}, \mathbf{k}) p_{t|0}(\mathbf{k}|\mathbf{n}_0)\Delta t + O(\Delta t^2), \end{aligned} \tag{56}$$

therefore, we get

$$\begin{aligned}
&\sum_{\mathbf{n}_0} p_{t|s}(\mathbf{n}|\mathbf{m}) p_{s|0}(\mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n}) \\
&= \sum_{\mathbf{n}_0} \Big(1_{\mathbf{mn}} + \sigma(t) Q(\mathbf{n},\mathbf{m})\Delta t\Big)\Big(p_{t|0}(\mathbf{m}|\mathbf{n}_0) - \sum_{\mathbf{k}} \sigma(t) Q(\mathbf{m},\mathbf{k}) p_{t|0}(\mathbf{k}|\mathbf{n}_0)\Delta t\Big) q_\theta(\mathbf{n}_0|\mathbf{n}) \\
&= 1_{\mathbf{mn}} \sum_{\mathbf{n}_0} p_{t|0}(\mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n}) \\
&\quad + \sigma(t)\Big(Q(\mathbf{n},\mathbf{m}) \sum_{\mathbf{n}_0} p_{t|0}(\mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n}) - 1_{\mathbf{mn}} \sum_{\mathbf{k}} Q(\mathbf{m},\mathbf{k}) \sum_{\mathbf{n}_0} p_{t|0}(\mathbf{k}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n})\Big)\Delta t.
\end{aligned} \tag{57}$$

Next, we determine the proportionality constant in Equation (50). The necessary condition is $\sum_{\mathbf{m}} \tilde{q}_{\theta s|t}(\mathbf{m}|\mathbf{n}) = 1$, which can be realized by dividing the right hand side by its summation over $\mathbf{m}$. Since the Chapman-Kolmogorov equation holds, we have

$$\sum_{\mathbf{m}} \sum_{\mathbf{n}_0} p_{t|s}(\mathbf{n}|\mathbf{m}) p_{s|0}(\mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n}) = \sum_{\mathbf{n}_0} p_{t|0}(\mathbf{n}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n}). \tag{58}$$

Combining all, we have

$$\begin{aligned}
&\tilde{q}_{\theta s|t}(\mathbf{m}|\mathbf{n}) \\
&= \frac{\sum_{\mathbf{n}_0} p_{t|s}(\mathbf{n}|\mathbf{m}) p_{s|0}(\mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n})}{\sum_{\mathbf{m}} \sum_{\mathbf{n}_0} p_{t|s}(\mathbf{n}|\mathbf{m}) p_{s|0}(\mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n})} \\
&= 1_{\mathbf{mn}} \underbrace{\frac{\sum_{\mathbf{n}_0} p_{t|0}(\mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n})}{\sum_{\mathbf{n}_0} p_{t|0}(\mathbf{n}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n})}}_{1} \\
&\quad + \sigma(t)\Big(Q(\mathbf{n},\mathbf{m}) \frac{\sum_{\mathbf{n}_0} p_{t|0}(\mathbf{m}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n})}{\sum_{\mathbf{n}_0} p_{t|0}(\mathbf{n}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n})} - 1_{\mathbf{mn}} \sum_{\mathbf{k}} Q(\mathbf{m},\mathbf{k}) \frac{\sum_{\mathbf{n}_0} p_{t|0}(\mathbf{k}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n})}{\sum_{\mathbf{n}_0} p_{t|0}(\mathbf{n}|\mathbf{n}_0) q_\theta(\mathbf{n}_0|\mathbf{n})}\Big)\Delta t.
\end{aligned}$$

Comparing it to the form in Equation (52), we get teh formula. □

Since we use the diffusion matrix decomposition in Equation (37), the score function takes the form $s_\theta(\mathbf{n},t)_{\mathbf{n}_{p\to m}}$ in short $s_\theta(\mathbf{n},t)_{p\to m}$. In addition, assuming the decomposition of the q_θ model as $q_\theta(\mathbf{n}_0|\mathbf{n}) = \prod_{q=1}^{N} q_\theta([\mathbf{n}_0]_i|\mathbf{n})$, we can marginalize over all but the p-th component, yielding the following:

$$s_\theta(\mathbf{n},t)_{p\to m} = \frac{\sum_{\mathbf{n}_0} p_{t|0}(m|[\mathbf{n}_0]_p) q_\theta(\mathbf{n}_0|\mathbf{n})}{\sum_{\mathbf{n}_0} p_{t|0}(n_p|[\mathbf{n}_0]_p) q_\theta(\mathbf{n}_0|\mathbf{n})} = \frac{\sum_{[\mathbf{n}_0]_p} p_{t|0}(m|[\mathbf{n}_0]_p) q_\theta([\mathbf{n}_0]_p|\mathbf{n})}{\sum_{[\mathbf{n}_0]_p} p_{t|0}(n_p|[\mathbf{n}_0]_p) q_\theta([\mathbf{n}_0]_p|\mathbf{n})}. \tag{59}$$

Uniform diffusion If we take $Q^{\text{tok}}(m,n) = 1$ $(m \neq n)$, then the token-level transition can be calculated as

$$p_{t|0}(m|n) = e^{-V\bar{\sigma}(t)} 1_{mn} + (1 - e^{-V\bar{\sigma}(t)})\frac{1}{V}, \tag{60}$$

and

$$s_\theta(\mathbf{n},t)_{p\to m} = \frac{e^{-V\bar{\sigma}(t)} q_\theta(m|\mathbf{n}) + (1 - e^{-V\bar{\sigma}(t)})\frac{1}{V}}{e^{-V\bar{\sigma}(t)} q_\theta(n_p|\mathbf{n}) + (1 - e^{-V\bar{\sigma}(t)})\frac{1}{V}}. \tag{61}$$

Masked diffusion If we take $Q^{\text{tok}}(m,n) = 1_{m0} 1_{n\neq 0}$, then the token-level transition can be calculated as

$$p_{t|0}(m|n) = e^{-\bar{\sigma}(t)} 1_{mn} + (1 - e^{-\bar{\sigma}(t)}) 1_{m0}, \tag{62}$$

and

$$s_\theta(\mathbf{n}, t)_{p\to m} = \frac{e^{-\bar{\sigma}(t)} q_\theta(m|\mathbf{n}) + (1 - e^{-\bar{\sigma}(t)}) 1_{m0}}{e^{-\bar{\sigma}(t)} q_\theta(n_p|\mathbf{n}) + (1 - e^{-\bar{\sigma}(t)}) 1_{n_p 0}} = \frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} q_\theta(m|\mathbf{n}), \tag{63}$$

for $n_p = 0, m \neq 0$.

C A Brief Review of Open Quantum Systems

This section reviews the concepts of open quantum systems underlying the formulation of diffusion language models in Section 3.

C.1 Hilbert Space and Notation

We work with a finite-dimensional complex Hilbert space $\mathcal{H}$, whose vectors represent pure quantum states when normalized.[4] We use the following notation:

State vectors and bra-ket notation A vector in $\mathcal{H}$ is written as $|\boldsymbol{\psi}\rangle$, called a ket. Its Hermitian conjugate $\langle\boldsymbol{\psi}| = (|\boldsymbol{\psi}\rangle)^\dagger$ is called a bra and belongs to the dual space $\mathcal{H}^*$. This convention is known as **bra-ket notation**.

Inner product The inner product of two vectors is written as $\langle\boldsymbol{\psi}_1|\boldsymbol{\psi}_2\rangle \in \mathbb{C}$. It satisfies conjugate symmetry: $\langle\boldsymbol{\psi}_1|\boldsymbol{\psi}_2\rangle^* = \langle\boldsymbol{\psi}_2|\boldsymbol{\psi}_1\rangle$, where $*$ denotes complex conjugation.

Norm The norm of $|\boldsymbol{\psi}\rangle$ is $\| |\boldsymbol{\psi}\rangle \| = \sqrt{\langle\boldsymbol{\psi}|\boldsymbol{\psi}\rangle}$. A normalized state vector satisfies $\langle\boldsymbol{\psi}|\boldsymbol{\psi}\rangle = 1$.

Operators and outer products A linear map $L : \mathcal{H} \to \mathcal{H}$ is called an operator. The outer product of two vectors, $|\boldsymbol{\psi}_1\rangle \langle\boldsymbol{\psi}_2|$, defines an operator whose action is

$$\left(|\boldsymbol{\psi}_1\rangle \langle\boldsymbol{\psi}_2|\right) |\boldsymbol{\psi}\rangle = |\boldsymbol{\psi}_1\rangle \langle\boldsymbol{\psi}_2|\boldsymbol{\psi}\rangle .$$

C.2 Quantum State and Measurement

A pure quantum state is represented by a normalized vector in $\mathcal{H}$. More generally, statistical uncertainty or interactions with an environment require a description in terms of density operators:

> **Definition C.1: Quantum State** An operator ρ on $\mathcal{H}$ is called a density operator, or quantum state, if it satisfies
>
> 1. Hermiticity: $\rho^\dagger = \rho$,
> 2. positive semidefiniteness: $\rho \geq 0$, and
> 3. unit trace: $\mathrm{Tr}_{\mathcal{H}} \rho = 1$.

By the spectral theorem, any density operator can be written as $\rho = \sum_i p_i |e_i\rangle \langle e_i|$, where $\{|e_i\rangle\}$ is an orthonormal basis, $p_i \geq 0$, and $\sum_i p_i = 1$. Thus, its eigenvalues form a probability distribution, while the basis specifies the corresponding quantum states.

> **Definition C.2: Pure State** A normalized vector $|\boldsymbol{\psi}\rangle \in \mathcal{H}$, with $\langle\boldsymbol{\psi}|\boldsymbol{\psi}\rangle = 1$, defines a quantum state
>
> $$\rho = |\boldsymbol{\psi}\rangle \langle\boldsymbol{\psi}| . \tag{64}$$
>
> A state of this form is called a pure state.

As noted above, any density operator admits a decomposition $\rho = \sum_i p_i |e_i\rangle \langle e_i|$. It can therefore be represented as a statistical ensemble of pure states. A state that is not pure is called a mixed state.

[4] In the language-model formulation of Section 3, we use $\mathcal{H} = \bigotimes_{p=1}^{N} \mathbb{C}^V$, where V is the number of token states and N is the sequence length.

A density operator determines the probabilities of measurement outcomes. For a measurement in a chosen orthonormal basis, the outcome is sampled according to the Born rule, as defined below.[5]

Definition C.3: Projective Measurement Let $\{|e\rangle\}_{e\in E}$ be an orthonormal basis of $\mathcal{H}$. For a quantum state ρ, the probability of obtaining outcome e is

$$P(e) = \mathrm{Tr}_{\mathcal{H}}\big(|e\rangle\langle e|\,\rho\big) = \langle e|\,\rho\,|e\rangle\,.$$

C.3 Dynamics of Open Quantum Systems

An open quantum system interacts with its environment, which can induce dissipation and fluctuations. Its evolution is described by maps that preserve the validity of quantum states, including when the system is entangled with an auxiliary system.

Definition C.4: Completely Positive Trace-Preserving (CPTP) Map A linear map $\mathcal{T}$ on operators on $\mathcal{H}$ is completely positive and trace-preserving if it admits a Kraus representation

$$\mathcal{T}(\rho) = \sum_i K_i \rho K_i^\dagger, \qquad \sum_i K_i^\dagger K_i = 1,$$

where the K_i are operators on $\mathcal{H}$ and 1 is the identity operator.

CPTP maps are quantum analogues of Markov kernels: they map density operators to density operators. Complete positivity additionally ensures that $\mathcal{T}\otimes\mathrm{id}$ preserves positivity for any auxiliary system, where id is the identity map on that system.

The Kraus representation makes positivity and trace preservation explicit. For any $|\psi\rangle\in\mathcal{H}$, let $|\psi_i\rangle = K_i^\dagger|\psi\rangle$. Then

$$\langle\psi|\,\mathcal{T}(\rho)\,|\psi\rangle = \sum_i \langle\psi_i|\,\rho\,|\psi_i\rangle \geq 0. \tag{65}$$

Using the cyclicity of the trace, we also obtain

$$\mathrm{Tr}_{\mathcal{H}}\mathcal{T}(\rho) = \sum_i \mathrm{Tr}_{\mathcal{H}}\big(K_i\rho K_i^\dagger\big) = \mathrm{Tr}_{\mathcal{H}}\left(\rho\sum_i K_i^\dagger K_i\right) = \mathrm{Tr}_{\mathcal{H}}\rho = 1. \tag{66}$$

In classical diffusion, the generator L_t in $\frac{d}{dt}p_t = L_t(p_t)$ is chosen so that the evolution from any time s to $t\geq s$ defines a Markov kernel. The quantum counterpart is an evolution equation for ρ_t whose propagators are CPTP maps.

Theorem C.5: Lindblad Equation Consider the evolution equation

$$\frac{d}{dt}\rho_t = \mathcal{L}_t(\rho_t), \tag{67}$$

where $\mathcal{L}_t$ is a continuous family of linear maps on operators on a finite-dimensional Hilbert space. Suppose that, for every $t\geq s$, the solution defines a CPTP map from ρ_s to ρ_t. Then there exist a Hermitian operator H_t, called the *Hamiltonian*, and operators $\{L_t^\mu\}_{\mu\in M}$, called *jump operators*, such that

$$\mathcal{L}_t(\rho_t) = -i[H_t,\rho_t] + \sum_\mu \left(L_t^\mu\rho_t(L_t^\mu)^\dagger - \frac{1}{2}\left\{(L_t^\mu)^\dagger L_t^\mu,\rho_t\right\}\right), \tag{68}$$

where $[A,B] = AB-BA$ and $\{A,B\} = AB+BA$ denote the commutator and anticommutator, respectively.

[5] More general measurements can be described by positive operator-valued measures (POVMs). Here, we restrict attention to projective measurements in an orthonormal basis.

Proof

We omit the proof here; see, for example, Manzano (2019). □

As we will see below, Equation (67) generalizes classical diffusion dynamics to quantum states. A corresponding time-reversed formulation can also be constructed (Kwon & Kim, 2018), providing a quantum counterpart to the reverse-time dynamics used in diffusion models.

Theorem C.6: Time reversal of Lindblad equation Under the time-backward operators

$$\tilde{H}_t = -\frac{1}{2}\left(\sqrt{\rho_t}H_t + i\frac{d\sqrt{\rho_t}}{dt} + \frac{i}{2}\sum_\mu \sqrt{\rho_t}L_t^{\mu\dagger}L_t^\mu\right)\frac{1}{\sqrt{\rho_t}} + (\text{1st term})^\dagger, \tag{69}$$

$$\tilde{L}_t^\mu = \sqrt{\rho_t}L_t^{\mu\dagger}\frac{1}{\sqrt{\rho_t}}, \tag{70}$$

the Lindblad equation can be represented by

$$-\frac{d}{dt}\rho_t = -i[\tilde{H}_t, \rho_t] + \sum_\mu \tilde{L}_t^\mu \rho_t \tilde{L}_t^{\mu\dagger} - \frac{1}{2}\left\{\sum_\mu \tilde{L}_t^{\mu\dagger}\tilde{L}_t^\mu, \rho_t\right\}. \tag{71}$$

Proof

For a proof, see the appendix of Kwon & Kim (2018). □

C.4 Stochastic unraveling

C.4.1 Unraveling 1-jump per step

There are several approaches to simulating the Lindblad equation. Here, we introduce the quantum jump method, which is based on a stochastic differential equation driven by Poisson processes. To simplify the notation, we define the effective Hamiltonian as

$$H_t^{\text{eff}} = H_t - \frac{i}{2}\sum_\mu L_t^{\mu\dagger}L_t^\mu. \tag{72}$$

Theorem C.7: Quantum Jump Let us consider one of stochastic time evolutions for pure state vector with Poisson processes dN_t^μ,

$$\begin{aligned}|\boldsymbol{\psi}_{t+dt}\rangle &= U_{t+dt|t}^{\text{eff (P)}}|\boldsymbol{\psi}_t\rangle \\ &\propto (1 - iH_t^{\text{eff}}dt)|\boldsymbol{\psi}_t\rangle + \sum_\mu (L_t^\mu - 1)dN_t^\mu|\boldsymbol{\psi}_t\rangle,\end{aligned} \tag{73}$$

where dN_t^μ are multiple Poisson processes with

$$\text{Prob}\Big(\forall\mu, dN_t^\mu = 0\Big) = 1 - \sum_\mu \langle\boldsymbol{\psi}_t| L_t^{\mu\dagger}L_t^\mu|\boldsymbol{\psi}_t\rangle\, dt, \tag{74}$$

$$\text{Prob}\Big(\exists^1\mu, dN_t^\mu = 1,\Big) = \langle\boldsymbol{\psi}_t| L_t^{\mu\dagger}L_t^\mu|\boldsymbol{\psi}_t\rangle\, dt. \tag{75}$$

then the update $|\boldsymbol{\psi}_t\rangle\langle\boldsymbol{\psi}_t| \to \underset{dN_t}{\mathbb{E}}|\boldsymbol{\psi}_{t+dt}\rangle\langle\boldsymbol{\psi}_{t+dt}|$ is given by Lindblad equation (67).

Proof

Throughout this calculation, we use the following Itô multiplication rules for the jump increments:

$$0 = dt\, dN_t^\mu = dt^2, \quad dN_t^\mu dN_t^\nu = \delta^{\mu\nu}dN_t^\mu. \tag{76}$$

To determine the normalization factor in Equation (73), we compute the squared norm of the unnormalized state:

$$
\begin{aligned}
&\|\text{rhs of Equation (73)}\|^2 \\
&= \langle \boldsymbol{\psi}_t | \, (1 + \underbrace{i H_t^{\text{eff}\dagger} dt - i H_t^{\text{eff}} dt}_{-\sum_\mu L_t^{\mu\dagger} L_t^\mu}) + \sum_{\mu,\nu} \underbrace{dN_t^\mu dN_t^\nu}_{\delta^{\mu\nu} dN_t^\mu} (L_t^{\mu\dagger} - 1)(L_t^\nu - 1) \, | \boldsymbol{\psi}_t \rangle \\
&\quad + \langle \boldsymbol{\psi}_t | \sum_\mu (L_t^\mu + L_t^{\mu\dagger} - 2) dN_t^\mu \, | \boldsymbol{\psi}_t \rangle \\
&= 1 - \sum_\mu \langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} dt + \sum_\mu (\langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} - 1) dN_t^\mu,
\end{aligned}
$$

which fixes the normalization as

$$
|\boldsymbol{\psi}_{t+dt}\rangle = \frac{(1 - i H_t^{\text{eff}} dt) \, |\boldsymbol{\psi}_t\rangle + \sum_\mu (L_t^\mu - 1) dN_t^\mu \, |\boldsymbol{\psi}_t\rangle}{\sqrt{1 - \sum_\mu \langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} dt + \sum_\mu (\langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} - 1) dN_t^\mu}}.
$$

This expression simplifies when we consider the no-jump and jump cases separately. If no jump occurs, $dN_t^\mu = 0$ for all μ, and

$$
|\boldsymbol{\psi}_{t+dt}\rangle = \frac{(1 - i H_t^{\text{eff}} dt) \, |\boldsymbol{\psi}_t\rangle}{\sqrt{1 - \sum_\mu \langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} dt}}.
$$

If a jump occurs in channel μ, then $dN_t^\mu = 1$ and $dN_t^\nu = 0$ for all $\nu \neq \mu$, giving

$$
|\boldsymbol{\psi}_{t+dt}\rangle = \frac{(1 - i H_t^{\text{eff}} dt) \, |\boldsymbol{\psi}_t\rangle + (L_t^\mu - 1) \, |\boldsymbol{\psi}_t\rangle}{\sqrt{1 - \sum_\mu \langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} dt + (\langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} - 1)}} = \frac{L_t^\mu \, |\boldsymbol{\psi}_t\rangle}{\sqrt{\langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t}}},
$$

neglecting terms of order dt. Thus,

$$
|\boldsymbol{\psi}_{t+dt}\rangle = \begin{cases} \frac{(1 - i H_t^{\text{eff}} dt) |\boldsymbol{\psi}_t\rangle}{\sqrt{1 - \sum_\mu \langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} dt}} & \text{with Prob}\Big(\forall \mu, dN_t^\mu = 0\Big) = 1 - \sum_\mu \langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} dt \\ \frac{L_t^\mu |\boldsymbol{\psi}_t\rangle}{\sqrt{\langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t}}} & \text{with Prob}\Big(\exists^1 \mu, dN_t^\mu = 1\Big) = \langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} dt \end{cases}, \tag{77}
$$

and

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{dN_t} |\boldsymbol{\psi}_{t+dt}\rangle \langle \boldsymbol{\psi}_{t+dt}| \\
&= \underbrace{\text{Prob}\Big(\forall \mu, \; dN_t^\mu = 0\Big)}_{1 - \sum_\mu \langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} dt} \frac{(1 - i H_t^{\text{eff}} dt) \, |\boldsymbol{\psi}_t\rangle}{\sqrt{1 - \sum_\mu \langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} dt}} \frac{\langle \boldsymbol{\psi}_t | \, (1 + i H_t^{\text{eff}\dagger} dt)}{\sqrt{1 - \sum_\mu \langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} dt}} \\
&\qquad\qquad + \sum_\mu \underbrace{\text{Prob}\Big(\exists^1 \mu, dN_t^\mu = 1, \Big)}_{\langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t} dt} \frac{L_t^\mu \, |\boldsymbol{\psi}_t\rangle}{\sqrt{\langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t}}} \frac{\langle \boldsymbol{\psi}_t | \, L_t^{\mu\dagger}}{\sqrt{\langle L_t^{\mu\dagger} L_t^\mu \rangle_{\boldsymbol{\psi}_t}}} \\
&= (1 - i H_t^{\text{eff}} dt) \, |\boldsymbol{\psi}_t\rangle \langle \boldsymbol{\psi}_t| \, (1 + i H_t^{\text{eff}\dagger} dt) + \sum_\mu L_t^\mu \, |\boldsymbol{\psi}_t\rangle \langle \boldsymbol{\psi}_t| \, L_t^{\mu\dagger} dt \\
&= |\boldsymbol{\psi}_t\rangle \langle \boldsymbol{\psi}_t| \\
&\quad + \underbrace{\Big(-i \Big[H_t, |\boldsymbol{\psi}_t\rangle \langle \boldsymbol{\psi}_t| \Big] - \frac{1}{2} \Big\{ \sum_\mu L_t^{\mu\dagger} L_t^\mu, |\boldsymbol{\psi}_t\rangle \langle \boldsymbol{\psi}_t| \Big\} + \sum_\mu L_t^\mu \, |\boldsymbol{\psi}_t\rangle \langle \boldsymbol{\psi}_t| \, L_t^{\mu\dagger} \Big) dt}_{\mathcal{L}_t\Big(|\boldsymbol{\psi}_t\rangle\langle\boldsymbol{\psi}_t|\Big)}.
\end{aligned}
$$

□

D PROOFS OF THE MAIN THEOREMS

This subsection provides the detailed explanation of Theorem 3.1 in the main text.

D.1 PROOF OF THEOREM 3.1

An important observation in this paper is that we can embed the usual diffusion process to the diagonal part of Lindblad equation Equation (67).

Theorem D.1: Pauli master equation Under the condition:

$$H = 0, \quad \forall \mathbf{k}, \mathbf{m}, \mathbf{n} \in \mathbb{V}^N \, (\mathbf{m} \neq \mathbf{n}), \begin{cases} 0 & = \sum_\mu \langle \mathbf{m}| \, L_t^\mu \, |\mathbf{k}\rangle \, \langle \mathbf{k}| \, L_t^{\mu\dagger} \, |\mathbf{n}\rangle \,, \\ 0 & = \sum_\mu \langle \mathbf{m}| \, L_t^{\mu\dagger} \, |\mathbf{k}\rangle \, \langle \mathbf{k}| \, L_t^\mu \, |\mathbf{n}\rangle \,, \end{cases} \tag{78}$$

then ρ_t can be diagonal and Equation (67) reduces to the diffusion ODE Equation (1), i.e.,

$$\rho_t = \sum_{\mathbf{m} \in \mathbb{V}^N} p_t(\mathbf{m}) \, |\mathbf{m}\rangle \, \langle \mathbf{m}| \,, \quad \frac{d}{dt} p_t(\mathbf{m}) = \sum_{\mathbf{n} \in \mathbb{V}^N} Q_t(\mathbf{m}, \mathbf{n}) p_t(\mathbf{n}), \tag{79}$$

where

$$Q_t(\mathbf{m}, \mathbf{n}) = \sum_\mu |\, \langle \mathbf{m}| \, L_t^\mu \, |\mathbf{n}\rangle \,|^2 \quad (\mathbf{m} \neq \mathbf{n}), \quad Q_t(\mathbf{m}, \mathbf{m}) = - \sum_{\mathbf{n} \in \mathbb{V}^N - \{\mathbf{m}\}} Q_t(\mathbf{n}, \mathbf{m}) \tag{80}$$

Proof

Basic idea is summarized in the following steps, 1) regarding $p_t(\mathbf{m}) = \langle \mathbf{m}| \, \rho_t \, |\mathbf{m}\rangle$, and 2) fixing $0 = \langle \mathbf{m}| \, \rho_t \, |\mathbf{n}\rangle$ for $\mathbf{m} \neq \mathbf{n}$. Let us check the first step. The time evolution of $p_t(\mathbf{m}) = \langle \mathbf{m}| \, \rho_t \, |\mathbf{m}\rangle$ under $0 = \langle \mathbf{m}| \, \rho_t \, |\mathbf{n}\rangle \quad (\mathbf{m} \neq \mathbf{n})$ reduces to the classical master equation as follows.

$$\frac{d}{dt} p_t(\mathbf{m}) = \langle \mathbf{m}| \, \frac{d}{dt} \rho_t \, |\mathbf{m}\rangle = \sum_\mu \left(\langle \mathbf{m}| \, L_t^\mu \rho_t L_t^{\mu\dagger} \, |\mathbf{m}\rangle - \frac{1}{2} \begin{pmatrix} \langle \mathbf{m}| \, L_t^{\mu\dagger} L_t^\mu \rho_t \, |\mathbf{m}\rangle \\ + \langle \mathbf{m}| \, \rho_t L_t^{\mu\dagger} L_t^\mu \, |\mathbf{m}\rangle \end{pmatrix} \right)$$

$$= \sum_\mu \sum_{\mathbf{n},\mathbf{k}} \left(\langle \mathbf{m}| \, L_t^\mu \, |\mathbf{n}\rangle \underbrace{\langle \mathbf{n}| \, \rho_t \, |\mathbf{k}\rangle}_{p_t(\mathbf{n}) 1_{\mathbf{nk}}} \langle \mathbf{k}| \, L_t^{\mu\dagger} \, |\mathbf{m}\rangle - \frac{1}{2} \begin{pmatrix} \langle \mathbf{m}| \, L_t^{\mu\dagger} \, |\mathbf{n}\rangle \, \langle \mathbf{n}| \, L_t^\mu \, |\mathbf{k}\rangle \underbrace{\langle \mathbf{k}| \, \rho_t \, |\mathbf{m}\rangle}_{p_t(\mathbf{k}) 1_{\mathbf{km}}} \\ + \underbrace{\langle \mathbf{m}| \, \rho_t \, |\mathbf{n}\rangle}_{p_t(\mathbf{m}) 1_{\mathbf{mn}}} \langle \mathbf{n}| \, L_t^{\mu\dagger} \, |\mathbf{k}\rangle \, \langle \mathbf{k}| \, L_t^\mu \, |\mathbf{m}\rangle \end{pmatrix} \right)$$

$$= \sum_{\mathbf{n} \in \mathbb{V}^N - \mathbf{m}} \left(\underbrace{\sum_\mu |\, \langle \mathbf{m}| \, L_t^\mu \, |\mathbf{n}\rangle \,|^2 \, p_t(\mathbf{n})}_{Q_t(\mathbf{m},\mathbf{n})} - \underbrace{\sum_\mu |\, \langle \mathbf{n}| \, L_t^\mu \, |\mathbf{m}\rangle \,|^2 \, p_t(\mathbf{m})}_{Q_t(\mathbf{n},\mathbf{m})} \right) = \sum_{\mathbf{n} \in \mathbb{V}^N} Q_t(\mathbf{m}, \mathbf{n}) p_t(\mathbf{n}),$$

where we define $Q_t(\mathbf{m}, \mathbf{m}) = - \sum_{\mathbf{n} \in \mathbb{V}^N - \{\mathbf{m}\}} Q_t(\mathbf{n}, \mathbf{m})$. Next, we consider the condition for $\frac{d}{dt} \langle \mathbf{m}| \, \rho_t \, |\mathbf{n}\rangle = 0$ with $\mathbf{m} \neq \mathbf{n}$.

$$0 = \langle \mathbf{m}| \, \frac{d}{dt} \rho_t \, |\mathbf{n}\rangle = \sum_\mu \left(\langle \mathbf{m}| \, L_t^\mu \rho_t L_t^{\mu\dagger} \, |\mathbf{n}\rangle - \frac{1}{2} \begin{pmatrix} \langle \mathbf{m}| \, L_t^{\mu\dagger} L_t^\mu \rho_t \, |\mathbf{n}\rangle \\ + \langle \mathbf{m}| \, \rho_t L_t^{\mu\dagger} L_t^\mu \, |\mathbf{n}\rangle \end{pmatrix} \right)$$

$$= \sum_\mu \sum_{\mathbf{l},\mathbf{k}} \left(\langle \mathbf{m}| \, L_t^\mu \, |\mathbf{l}\rangle \underbrace{\langle \mathbf{l}| \, \rho_t \, |\mathbf{k}\rangle}_{p_t(\mathbf{l}) 1_{\mathbf{lk}}} \langle \mathbf{k}| \, L_t^{\mu\dagger} \, |\mathbf{n}\rangle - \frac{1}{2} \begin{pmatrix} \langle \mathbf{m}| \, L_t^{\mu\dagger} \, |\mathbf{l}\rangle \, \langle \mathbf{l}| \, L_t^\mu \, |\mathbf{k}\rangle \underbrace{\langle \mathbf{k}| \, \rho_t \, |\mathbf{n}\rangle}_{p_t(\mathbf{k}) 1_{\mathbf{kn}}} \\ + \underbrace{\langle \mathbf{m}| \, \rho_t \, |\mathbf{l}\rangle}_{p_t(\mathbf{m}) 1_{\mathbf{ml}}} \langle \mathbf{l}| \, L_t^{\mu\dagger} \, |\mathbf{k}\rangle \, \langle \mathbf{k}| \, L_t^\mu \, |\mathbf{n}\rangle \end{pmatrix} \right)$$

$$= \sum_\mu \sum_{\mathbf{k}} \left(\langle \mathbf{m}| \, L_t^\mu \, |\mathbf{k}\rangle \, \langle \mathbf{k}| \, L_t^{\mu\dagger} \, |\mathbf{n}\rangle \, p_t(\mathbf{k}) - \frac{1}{2} \begin{pmatrix} \langle \mathbf{m}| \, L_t^{\mu\dagger} \, |\mathbf{k}\rangle \, \langle \mathbf{k}| \, L_t^\mu \, |\mathbf{n}\rangle \, p_t(\mathbf{n}) \\ + \langle \mathbf{m}| \, L_t^{\mu\dagger} \, |\mathbf{k}\rangle \, \langle \mathbf{k}| \, L_t^\mu \, |\mathbf{n}\rangle \, p_t(\mathbf{m}) \end{pmatrix} \right)$$

Sufficient condition for vanishing it is, therefore

$$\forall \mathbf{k}, \mathbf{m}, \mathbf{n} \in \mathbb{V}^N \ (\mathbf{m} \neq \mathbf{n}) \quad \begin{cases} 0 = \sum_\mu \langle \mathbf{m}| \, L_t^\mu \, |\mathbf{k}\rangle \, \langle \mathbf{k}| \, L_t^{\mu\dagger} \, |\mathbf{n}\rangle \\ 0 = \sum_\mu \langle \mathbf{m}| \, L_t^{\mu\dagger} \, |\mathbf{k}\rangle \, \langle \mathbf{k}| \, L_t^{\mu} \, |\mathbf{n}\rangle \end{cases} .$$

□

We can prove the Theorem 3.1 as a corollary of this theorem. Under the constraint of jump operators in Equation (8) [6] and Equation (9), i.e.,

$$L_t^{\mu=(p,m',n')} = \sqrt{\sigma(t)} \begin{array}{c} I \\ \otimes \\ \vdots \\ \otimes \\ L^{(p)\text{tok}}_{m',n'} \\ \otimes \\ \vdots \\ \otimes \\ I \end{array} \quad \text{where } L^{(p)\text{tok}}_{m',n'} = \sqrt{Q^{\text{tok}}(m',n')} \, |m'\rangle \, \langle n'| \, . \tag{81}$$

Under these conditions, we have

$$\begin{aligned} &\sum_\mu \langle \mathbf{m}| \, L_t^\mu \, |\mathbf{k}\rangle \, \langle \mathbf{k}| \, L_t^{\mu\dagger} \, |\mathbf{n}\rangle \\ &= \sigma(t) \sum_{p,m',n'} \begin{pmatrix} \langle m_1| \, I \, |k_1\rangle & \langle k_1| \, I \, |n_1\rangle & = 1_{m_1 n_1} 1_{m_1 k_1} \\ \times & \times & \\ \vdots & \vdots & \vdots \\ \times & \times & \\ \sqrt{Q^{\text{tok}}(m',n')^2} \ \langle m_p|m'\rangle \, \langle n'|k_p\rangle & \langle k_p|n'\rangle \, \langle m'|n_p\rangle & = 1_{m_p n_p} 1_{m_p m'} 1_{k_p n'} \\ \times & \times & \\ \vdots & \vdots & \vdots \\ \times & \times & \\ \langle m_N| \, I \, |k_N\rangle & \langle k_N| \, I \, |n_N\rangle & = 1_{m_N n_N} 1_{m_N k_N} \end{pmatrix} \\ &\propto 1_{\mathbf{mn}} = 0 \quad (\text{for } \mathbf{m} \neq \mathbf{n}). \end{aligned}$$

We can repeat same procedure for checking another condition:

$$\begin{aligned} &\sum_\mu \langle \mathbf{m}| \, L_t^{\mu\dagger} \, |\mathbf{k}\rangle \, \langle \mathbf{k}| \, L_t^{\mu} \, |\mathbf{n}\rangle \\ &= \sigma(t) \sum_{p,m',n'} \begin{pmatrix} \langle m_1| \, I \, |k_1\rangle & \langle k_1| \, I \, |n_1\rangle & = 1_{m_1 n_1} 1_{m_1 k_1} \\ \times & \times & \\ \vdots & \vdots & \vdots \\ \times & \times & \\ \sqrt{Q^{\text{tok}}(m',n')^2} \ \langle m_p|n'\rangle \, \langle m'|k_p\rangle & \langle k_p|m'\rangle \, \langle n'|n_p\rangle & = 1_{m_p n_p} 1_{m_p n'} 1_{k_p m'} \\ \times & \times & \\ \vdots & \vdots & \vdots \\ \times & \times & \\ \langle m_N| \, I \, |k_N\rangle & \langle k_N| \, I \, |n_N\rangle & = 1_{m_N n_N} 1_{m_N n_N} \end{pmatrix} \\ &\propto 1_{\mathbf{mn}} = 0 \quad (\text{for } \mathbf{m} \neq \mathbf{n}). \end{aligned}$$

Therefore, we can apply Theorem D.1 and get the reduction of Lindblad equation to the diffusion equation. In the end, we calculate the value of $Q_t(\mathbf{m}, \mathbf{n})$ in Theorem D.1 with the jump operators in

[6] For computational purposes, we will represent the tensor product of tensors along each dimension vertically.

Equation (81)

$$\sum_{\mu} |\langle \mathbf{m}| L_t^{\mu} |\mathbf{n}\rangle|^2 = \sum_{p=1}^{N} \sum_{m',n'} \begin{array}{c} |\langle m_1| I |n_1\rangle|^2 \\ \times \\ \vdots \\ \times \\ |\langle m_p| L_{m',n'}^{(p)\text{tok}} |n_p\rangle|^2 \\ \times \\ \vdots \\ \times \\ |\langle m_N| I |n_N\rangle|^2 \end{array} = \sum_{p=1}^{N} 1_{\mathbf{m}_{-p}\mathbf{n}_{-p}} Q^{\text{tok}}(m_p, n_p), \tag{82}$$

which proves Theorem 3.1.

D.2 TIME-BACKWARD CASE

First, we prove the time-reversal Hamiltonian $\tilde{H}$ defined in Equation (70) is zero.

Theorem D.2 Under the following conditions:

$$H = 0, \tag{83}$$

$$\rho_t \text{ is diagonal}, \tag{84}$$

$$\forall \mathbf{k} \in \mathbb{V}^N, \quad 0 = \sum_{\mu} \langle \mathbf{m}| L_t^{\mu\dagger} |\mathbf{k}\rangle \langle \mathbf{k}| L_t^{\mu} |\mathbf{n}\rangle \quad \text{for } \mathbf{m} \neq \mathbf{n}, \tag{85}$$

then the time-reversal Hamiltonian

$$\tilde{H} = -\frac{1}{2}\left(\sqrt{\rho_t}H + i\frac{d\sqrt{\rho_t}}{dt} + \frac{i}{2}\sum_{\mu}\sqrt{\rho_t}L_t^{\mu\dagger}L_t^{\mu}\right)\frac{1}{\sqrt{\rho_t}} + (\text{1st term})^{\dagger} \tag{86}$$

is zero.

Proof

From Equation (83) in the conditions, we have

$$\tilde{H} = -\frac{i}{2}\left(+\frac{d\sqrt{\rho_t}}{dt}\frac{1}{\sqrt{\rho_t}} + \frac{1}{2}\sum_{\mu}\sqrt{\rho_t}L_t^{\mu\dagger}L_t^{\mu}\frac{1}{\sqrt{\rho_t}}\right) + (\text{1st term})^{\dagger}, \tag{87}$$

and it is sufficient to show that the operator inside () is Hermite. From Equation (84) in the given conditions, we have

$$\frac{d\sqrt{\rho_t}}{dt}\frac{1}{\sqrt{\rho_t}} = \sum_{\mathbf{m}\in\mathbb{V}^N} \frac{\frac{d}{dt}\sqrt{p_t(\mathbf{m})}}{\sqrt{p_t(\mathbf{m})}} |\mathbf{m}\rangle \langle \mathbf{m}|, \tag{88}$$

which is Hermite. To demonstrate the Hermiticity of 2nd term, we sum over $\mathbf{k}$ for Equation (85) that gives

$$0 = \sum_{\mu} \langle \mathbf{m}| L_t^{\mu\dagger}L_t^{\mu} |\mathbf{n}\rangle. \tag{89}$$

This means the Hermitian operator $\sum_{\mu} L_t^{\mu\dagger}L_t^{\mu}$ is diagonal. Then, the 2nd term is Hermite

$$\sum_{\mu}\sqrt{\rho_t}L_t^{\mu\dagger}L_t^{\mu}\frac{1}{\sqrt{\rho_t}} = \underbrace{\sqrt{\rho_t}}_{\text{diagonal}}\underbrace{\left(\sum_{\mu}L_t^{\mu\dagger}L_t^{\mu}\right)}_{\text{diagonal}}\underbrace{\frac{1}{\sqrt{\rho_t}}}_{\text{diagonal}} = \underbrace{\sum_{\mu}L_t^{\mu\dagger}L_t^{\mu}}_{\text{Hermite}}, \tag{90}$$

where we used that diagonal operators commute.

□

Next, we show the time-reversal jump operators satisfies the condition for Theorem D.1. We use $\rho_t = \sum_{\mathbf{m}\in\mathbb{V}^N} p_t(\mathbf{m}) \left|\mathbf{m}\right\rangle \left\langle\mathbf{m}\right|$ in the following calculations.

$$\begin{aligned}\sum_\mu \left\langle\mathbf{m}\right| \tilde{L}_t^\mu \left|\mathbf{k}\right\rangle \left\langle\mathbf{k}\right| \tilde{L}_t^{\mu\dagger} \left|\mathbf{n}\right\rangle &= \sum_\mu \sqrt{p_t(\mathbf{m})} \left\langle\mathbf{m}\right| L_t^{\mu\dagger} \left|\mathbf{k}\right\rangle \frac{1}{\sqrt{p_t(\mathbf{k})}} \frac{1}{\sqrt{p_t(\mathbf{k})}} \left\langle\mathbf{k}\right| L_t^\mu \left|\mathbf{n}\right\rangle \sqrt{p_t(\mathbf{n})} \\ &= \frac{\sqrt{p_t(\mathbf{m})p_t(\mathbf{n})}}{p_t(\mathbf{k})} \underbrace{\sum_\mu \left\langle\mathbf{m}\right| L_t^{\mu\dagger} \left|\mathbf{k}\right\rangle \left\langle\mathbf{k}\right| L_t^\mu \left|\mathbf{n}\right\rangle}_{=0 \ \forall \mathbf{k},\mathbf{m},\mathbf{n} \ \mathbf{m}\neq\mathbf{n}}. \end{aligned} \tag{91}$$

$$\begin{aligned}\sum_\mu \left\langle\mathbf{m}\right| \tilde{L}_t^{\mu\dagger} \left|\mathbf{k}\right\rangle \left\langle\mathbf{k}\right| \tilde{L}_t^\mu \left|\mathbf{n}\right\rangle &= \sum_\mu \frac{1}{\sqrt{p_t(\mathbf{m})}} \left\langle\mathbf{m}\right| L_t^\mu \left|\mathbf{k}\right\rangle \sqrt{p_t(\mathbf{k})}\sqrt{p_t(\mathbf{k})} \left\langle\mathbf{k}\right| L_t^{\mu\dagger} \left|\mathbf{n}\right\rangle \frac{1}{\sqrt{p_t(\mathbf{n})}} \\ &= \frac{p_t(\mathbf{k})}{\sqrt{p_t(\mathbf{m})p_t(\mathbf{n})}} \underbrace{\sum_\mu \left\langle\mathbf{m}\right| L_t^\mu \left|\mathbf{k}\right\rangle \left\langle\mathbf{k}\right| L_t^{\mu\dagger} \left|\mathbf{n}\right\rangle}_{=0 \ \forall \mathbf{k},\mathbf{m},\mathbf{n} \ \mathbf{m}\neq\mathbf{n}}. \end{aligned} \tag{92}$$

Therefore, we can apply Theorem D.1 to the time-reversal Lindblad equation, and get

$$-\frac{d}{dt}p_t(\mathbf{m}) = \sum_{\mathbf{n}\in\mathbb{V}^N} \tilde{Q}_t(\mathbf{m},\mathbf{n})p_t(\mathbf{n}), \quad \text{where} \quad \tilde{Q}_t(\mathbf{m},\mathbf{n}) = \sum_\mu |\left\langle\mathbf{m}\right| \tilde{L}_t^\mu \left|\mathbf{n}\right\rangle|^2. \tag{93}$$

D.3 Sampling by stochastic unraveling and measurement

Let us restate the original theorem's statement.

Theorem D.3: Sampling based on Stochastic Unraveling Suppose $t \geq s$. Given a state $\left|\mathbf{n}\right\rangle$ at time s, we consider sampling $\mathbf{m}$ at time t by first performing stochastic unraveling from s to t and then projective measurement at t. Specifically:

1. Stochastic Unraveling: Iteratively update a pure state from $\left|\mathbf{n}\right\rangle$ to $\left|\boldsymbol{\psi}_t\right\rangle$ using one of unraveling operators.
2. Projective Measurement: Sample sentence $\mathbf{m}$ from the probability $P_{\text{proj}}\big(\mathbf{m}\big|\boldsymbol{\psi}_t\big) = |\left\langle\mathbf{m}\right|\left.\boldsymbol{\psi}_t\right\rangle|^2$.

This procedure yields a sample $\mathbf{m}$ distributed according to the conditional probability $p_{t|s}(\mathbf{m}|\mathbf{n})$ defined in Equation (38).

Proof

The joint probability for realizing $\mathbf{m}, \left|\boldsymbol{\psi}_t\right\rangle$ during this 2 steps is

$$P_{t|s}\big(\mathbf{m},\boldsymbol{\psi}_t\big|\mathbf{n}\big) = \underbrace{P_{\text{proj}}\big(\mathbf{m}_t\big|\boldsymbol{\psi}_t\big)}_{\text{Process 2}} \underbrace{P_{t|s}^{\text{unraveling}}\big(\boldsymbol{\psi}_t\big|\mathbf{n}\big)}_{\text{Process 1}}, \tag{94}$$

its marginal to $\mathbf{m}$ is

$$\begin{aligned}P_{t|s}(\mathbf{m}|\mathbf{n}) &= \mathop{\mathbb{E}}_{\boldsymbol{\psi}_t\sim P_{t|s}^{\text{unraveling}}(\cdot|\mathbf{n})} \Big[P_{\text{proj}}\big(\mathbf{m}\big|\boldsymbol{\psi}_t\big)\Big] = \mathop{\mathbb{E}}_{\boldsymbol{\psi}_t\sim P_{t|s}^{\text{unraveling}}(\cdot|\mathbf{n})} \Big[\left\langle\mathbf{m}\right|\left.\boldsymbol{\psi}_t\right\rangle \left\langle\boldsymbol{\psi}_t\right|\left.\mathbf{m}\right\rangle\Big] \\ &= \left\langle\mathbf{m}\right| \underbrace{\left(\mathop{\mathbb{E}}_{\boldsymbol{\psi}_t\sim P_{t|s}^{\text{unraveling}}(\cdot|\mathbf{n})} \Big[\left|\boldsymbol{\psi}_t\right\rangle \left\langle\boldsymbol{\psi}_t\right|\Big]\right)}_{=:\rho_{t|s}(\mathbf{n})} \left|\mathbf{m}\right\rangle. \end{aligned} \tag{95}$$

Now, thanks to Theorem C.7, the density operator $\rho_{t|s}(\mathbf{n})$ satisfies the Lindblad equation for t. In addition, under the setting of Theorem 3.1, the Lindblad equation reduces to the equation for

$\langle \mathbf{m}| \rho_{t|s}(\mathbf{n}) |\mathbf{m}\rangle$ with the diffusion equation in Equation (1), which means Equation (95) satisfies the diffusion equation in Equation (1). In addition, $\rho_{t|s}(\mathbf{n})$ reduces to $|\mathbf{n}\rangle \langle \mathbf{n}|$ at time $t = s$, which means

$$P_{t=s|s}(\mathbf{m}|\mathbf{n}) = \langle \mathbf{m}|\mathbf{n}\rangle \langle \mathbf{n}|\mathbf{m}\rangle = 1_{\mathbf{m}=\mathbf{n}}, \tag{96}$$

and it is identical to the initial condition of $p_{t=s|s}(\mathbf{m}|\mathbf{n})$ defined in Equation (38). We show $P_{t|s}(\mathbf{m}|\mathbf{n})$ satisfies: 1) same ODE to $p_{t|s}(\mathbf{m}|\mathbf{n})$, and 2) same initial condition to $p_{t|s}(\mathbf{m}|\mathbf{n})$, which means $P_{t|s} = p_{t|s}$ by uniqueness of the ODE solution. □

D.4 Explicit Forms of τ-leaping quantum states

This section provides the necessary information for executing simulations, including concrete expressions for jump operators in reverse time (both uniform and masked), as well as all information required for executing SDE simulations. Here, token-level jump operators must satisfy the following format as in Equation (9):

$$L^{\text{tok}}_{m,n} = \sqrt{Q^{\text{tok}}(m,n)}\, |m\rangle \langle n|\,, \quad (m \neq n), \tag{97}$$

D.4.1 Explicit forms of token-level time-reversal operators

Theorem D.4: Form of Time-reversal Jump Operators Under our setting in Equation (8), we have

$$\tilde{L}_t^{\mu=(p,m,n)} = \sum_{\mathbf{k}} \begin{matrix} |k_1\rangle \langle k_1| \\ \otimes \\ \vdots \\ \otimes \\ \sqrt{\sigma(t)}\tilde{L}^{(p)\text{tok}}_{m,n,t}(\mathbf{k}) \\ \otimes \\ \vdots \\ \otimes \\ |k_N\rangle \langle k_N| \end{matrix} \quad \text{where } \tilde{L}^{(p)\text{tok}}_{m,n,t}(\mathbf{k}) = \sqrt{\tilde{Q}^{(p)\text{tok}}(n,\mathbf{k})} 1_{mk_p} |n\rangle \langle k_p|\,.$$

Proof

By just substituting our conditions $L_t^{\mu=(p,m,n)} = \sqrt{\sigma(t)} I \otimes \cdots \otimes I \otimes L_{m,n}^{(p)\mathrm{tok}} \otimes I \otimes \cdots \otimes I$, $\rho_t = \sum_{\mathbf{k}\in\mathbb{V}^N} p_t(\mathbf{k}) \left|\mathbf{k}\right\rangle \left\langle\mathbf{k}\right|$ to the explicit form of $\tilde{L}_t^{\mu}$ defined in Equation (70), we get

$$
\begin{aligned}
\sqrt{\rho_t} L_t^{\mu=(p,m,n)\dagger} \frac{1}{\sqrt{\rho_t}} &= \sum_{\mathbf{l},\mathbf{k}\in\mathbb{V}^N} \sqrt{\frac{p_t(\mathbf{l})}{p_t(\mathbf{k})}\sigma(t)} \left|\mathbf{l}\right\rangle \left\langle\mathbf{k}\right| 1_{\mathbf{l}_{-p}\mathbf{k}_{-p}} \frac{\left\langle l_p\right| L_{m,n}^{(p)\mathrm{tok}\dagger} \left|k_p\right\rangle}{\sqrt{Q^{\mathrm{tok}}(m,n)} 1_{l_p n} 1_{m k_p}} \\
&= \sum_{\mathbf{k}\in\mathbb{V}^N} \begin{pmatrix} \sum_{l_p\in\mathbb{V}} \sqrt{Q^{\mathrm{tok}}(m,n)\frac{p_t(\mathbf{k}_{p\to l_p})}{p_t(\mathbf{k})}\sigma(t)} 1_{l_p n} 1_{m k_p} & \begin{matrix} \left|k_1\right\rangle\left\langle k_1\right| \\ \otimes \\ \vdots \\ \otimes \\ \left|l_p\right\rangle\left\langle k_p\right| \\ \otimes \\ \vdots \\ \otimes \\ \left|k_N\right\rangle\left\langle k_N\right| \end{matrix} \end{pmatrix} \\
&= \sum_{\mathbf{k}\in\mathbb{V}^N} \sqrt{\underbrace{Q^{\mathrm{tok}}(k_p,n)\frac{p_t(\mathbf{k}_{p\to n})}{p_t(\mathbf{k})}\sigma(t)}_{=\tilde{Q}^{(p)\mathrm{tok}}(n,\mathbf{k})} 1_{m k_p}} \begin{matrix} \left|k_1\right\rangle\left\langle k_1\right| \\ \otimes \\ \vdots \\ \otimes \\ \left|n\right\rangle\left\langle k_p\right| \\ \otimes \\ \vdots \\ \otimes \\ \left|k_N\right\rangle\left\langle k_N\right| \end{matrix} .
\end{aligned}
$$

□

The operator $\tilde{L}_{m,n,t}^{(p)\mathrm{tok}}(\mathbf{k})$ satisfies the following conditions.

Theorem D.5 The following holds:

$$
\begin{aligned}
&\tilde{L}_{m,n,t}^{(p)\mathrm{tok}}(\mathbf{k})^\dagger \tilde{L}_{m,n,t}^{(p)\mathrm{tok}}(\mathbf{l}) \propto 1_{k_p l_p} \\
&\tilde{L}_{m,n,t}^{(p)\mathrm{tok}}(\mathbf{k})^\dagger \tilde{L}_{m,n,t}^{(p)\mathrm{tok}}(\mathbf{k}) = \tilde{Q}^{(p)\mathrm{tok}}(n,\mathbf{k}) 1_{m k_p} \left|k_p\right\rangle\left\langle k_p\right| .
\end{aligned}
$$

Proof

By direct calculation,

$$
\tilde{L}_{m,n,t}^{(p)\mathrm{tok}}(\mathbf{k})^\dagger \tilde{L}_{m,n,t}^{(p)\mathrm{tok}}(\mathbf{l}) = \sqrt{\tilde{Q}^{(p)\mathrm{tok}}(n,\mathbf{k})\tilde{Q}^{(p)\mathrm{tok}}(n,\mathbf{l})} \underbrace{1_{m k_p} 1_{m l_p}}_{1_{k_p l_p} 1_{m k_p}} \left|k_p\right\rangle \underbrace{\left\langle n|n\right\rangle}_{1} \left\langle l_p\right| ,
$$

which shows the statement.

□

We calculate the time-reversal effective Hamiltonian within our setup.

Theorem D.6: Time-reversal effective Hamiltonian We have

$$\tilde{H}_{\text{eff}} = \sum_{p=1}^{N} \sum_{\mathbf{k}} \begin{array}{c} |k_1\rangle\langle k_1| \\ \otimes \\ \vdots \\ \otimes \\ \sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{k}) \\ \otimes \\ \vdots \\ \otimes \\ |k_N\rangle\langle k_N| \end{array} \quad \text{where} \begin{cases} \tilde{H}_t^{(p)\text{tok}}(\mathbf{k}) = -\frac{i}{2}\sum_{m,n} \tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k})^\dagger \tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k}) \\ = -\frac{i}{2}\sum_{m,n} \tilde{Q}^{(p)\text{tok}}(n,\mathbf{k}) 1_{mk_p} |k_p\rangle\langle k_p| \end{cases} .$$

Proof

$$\tilde{H}_{\text{eff}} = \underbrace{\tilde{H}}_{0 \text{ by Theorem D.2}} - \frac{i}{2}\sum_{\mu} \underbrace{\tilde{L}_t^{\mu\dagger}}_{\text{Theorem D.4}} \underbrace{\tilde{L}_t^{\mu}}_{\text{Theorem D.4}}$$

$$= -\frac{i}{2}\sum_{p=1}^{N}\sum_{m,n}\sum_{\mathbf{k},\mathbf{l}} \begin{array}{c} |k_1\rangle\langle k_1|l_1\rangle\langle l_1| \\ \otimes \\ \vdots \\ \otimes \\ \sigma(t)\tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k})^\dagger \tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{l}) \\ \otimes \\ \vdots \\ \otimes \\ |k_N\rangle\langle k_N|l_N\rangle\langle l_N| \end{array} .$$

Thanks to $\langle k|l\rangle = 1_{kl}$, we get $1_{\mathbf{k}_{-p}\mathbf{l}_{-p}}$ in the summation. In addition, $\tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k})^\dagger \tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{l}) \propto 1_{k_p l_p}$ by Theorem D.5, which reduces the sum only for $\mathbf{k} = \mathbf{l}$. Consequently, the sum turns into a single expression as follows:

$$\tilde{H}_{\text{eff}} = -\frac{i}{2}\sum_{p=1}^{N}\sum_{m,n}\sum_{\mathbf{k}} \begin{array}{c} |k_1\rangle\langle k_1| \\ \otimes \\ \vdots \\ \otimes \\ \sigma(t)\tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k})^\dagger \tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k}) \\ \otimes \\ \vdots \\ \otimes \\ |k_N\rangle\langle k_N| \end{array}$$

Finally, we compute the p-component by Theorem D.5:

$$\underbrace{-\frac{i}{2}\sum_{m,n} \tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k})^\dagger \tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k})}_{=:\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})} = -\frac{i}{2}\sum_{m,n} \tilde{Q}^{(p)\text{tok}}(n,\mathbf{k}) 1_{mk_p} |k_p\rangle\langle k_p|$$

□

As a lemma:

Theorem D.7 The following holds.

$$|k_p\rangle\langle k_p| - i\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})\Delta t = \sqrt{1+\sum_m \tilde{Q}^{(p)\text{tok}}(k_p,\mathbf{k})\sigma(t)1_{mk_p}\Delta t}\,|k_p\rangle\langle k_p|,$$

up to $O(\Delta t^2)$, where

$$\tilde{Q}^{(p)\text{tok}}(k_p,\mathbf{k}) = -\sum_{n\neq k_p}\tilde{Q}^{(p)\text{tok}}(n,\mathbf{k}). \tag{98}$$

Proof

By direct calculation,

$$\begin{aligned}
|k_p\rangle\langle k_p| - i\underbrace{\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})}_{\text{Theorem D.6}}\Delta t &= |k_p\rangle\langle k_p| - i\left(-\frac{i}{2}\sum_{m,n}\tilde{Q}^{(p)\text{tok}}(n,\mathbf{k})\sigma(t)1_{mk_p}|k_p\rangle\langle k_p|\right)\Delta t \\
&= \left(1-\frac{1}{2}\underbrace{\sum_{m,n}\tilde{Q}^{(p)\text{tok}}(n,\mathbf{k})}_{\sum_m -\tilde{Q}^{(p)\text{tok}}(k_p,\mathbf{k})}\sigma(t)1_{mk_p}\Delta t\right)|k_p\rangle\langle k_p| \\
&= \sqrt{1+\sum_m \tilde{Q}^{(p)\text{tok}}(k_p,\mathbf{k})\sigma(t)1_{mk_p}\Delta t}\,|k_p\rangle\langle k_p| + O(\Delta t^2)
\end{aligned}$$

□

Here, we shall refer to $\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})$ as the token-level Hamiltonian and $\tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k})$ as the token-level jump operator.

D.4.2 τ-LEAPING UNRAVELING

The naive unravelings are exact up to order dt, however, when attempting to use these methods for inferring language diffusion models, they are inefficient because of the design of $\hat{L}_t^{\mu}$. With this configuration, only one token can be jumped in a single step, making it inefficient. One approach to solving this problem is the *τ-leaping-like* method. We consider the quantum-jump unraveling in Equation (73).

Under this decomposition, we can introduce the corresponding concept for the unraveling of the τ-leaping sampling:

Theorem D.8: τ-leaping of quantum jump We define

$$\begin{aligned}
&\tilde{U}_{t-\Delta t|t}^{\tau\text{-leap}}|\boldsymbol{\psi}_t\rangle \\
&= \sum_{\mathbf{k}\in\mathbb{V}^N}\bigotimes_{p=1}^{N}\Big(|k_p\rangle\langle k_p| - i\sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})\Delta t \\
&\qquad + \sum_{m,n}(\sqrt{\sigma(t)}\tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k}) - |k_p\rangle\langle k_p|)\Delta N_t^{m,n|p}\Big)|\boldsymbol{\psi}_t\rangle\,/(\text{numerator norm}),
\end{aligned} \tag{99}$$

where $\Delta N_t^{m,n|p}$ is a new process defined by

$$\text{Prob}\Big(\forall(m,n), \Delta N_t^{m,n|p} = 0\Big) = 1 - \sum_{m,n} \langle \tilde{L}_t^{\mu=(p,m,n)\,\dagger} \tilde{L}_t^{\mu=(p,m,n)} \rangle_{\Psi_t} \Delta t \tag{100}$$

$$\text{Prob}\Big(\exists^1(m,n), \Delta N_t^{m,n|p} = 1\Big) = \langle \tilde{L}_t^{\mu=(p,m,n)\,\dagger} \tilde{L}_t^{\mu=(p,m,n)} \rangle_{\Psi_t} \Delta t. \tag{101}$$

Then, the operation becomes stochastic unraveling up to first order in Δt.

$$\mathbb{E}_{\Delta N}\left[\tilde{U}^{\tau\text{-leap}}_{t-\Delta t|t} |\Psi_t\rangle \langle\Psi_t| \tilde{U}^{\tau\text{-leap}\,\dagger}_{t-\Delta t|t}\right] = |\Psi_t\rangle \langle\Psi_t| + \tilde{\mathcal{L}}_t(|\Psi_t\rangle \langle\Psi_t|)\Delta t + O(\Delta t^2), \tag{102}$$

with the reverse-time Lindblad operator $\tilde{\mathcal{L}}_t$.

Proof

The definition of Poisson noise differs from conventional quantum jumps because it decomposes the jumps by token position. Therefore, the direct method of calculating the expectation value of the density matrix would be the most straightforward approach.
To conduct this, explicitly writing the numerator component of the vector state yields the following:

$$\begin{aligned}
&\sum_{\mathbf{k}\in\mathbb{V}^N} \bigotimes_{p=1}^{N} \Big(|k_p\rangle \langle k_p| - i\sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})\Delta t \\
&\quad + \sum_{m,n} (\sqrt{\sigma(t)}\tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k}) - |k_p\rangle \langle k_p|)\Delta N_t^{m,n|p}\Big) |\Psi_t\rangle \\
&= \sum_{\mathbf{k}} \bigotimes_{p=1}^{N} \left\{ \begin{array}{ll} |k_p\rangle \langle k_p| - i\sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})\Delta t & \text{w/ Pr}\Big(\forall(m,n), \Delta N_t^{m,n|p} = 0\Big) \\ \sqrt{\sigma(t)}\tilde{L}_{m_p,n_p,t}^{(p)\text{tok}}(\mathbf{k}) & \text{w/ Pr}\Big(\exists^1(m_p,n_p), \Delta N_t^{m_p,n_p|p} = 1\Big) \end{array} \right\} |\Psi_t\rangle
\end{aligned}$$

In this expression, operators acting on individual tokens in their respective Hilbert spaces $\mathcal{H}_{\mathbb{V}}$ follow independent probability distributions for each token. To obtain the full form of the time evolution, we need to divide by the norm of each operator's action.

norm

$$\begin{aligned}
&= \langle\Psi_t| \sum_{\mathbf{k},\mathbf{l}} \bigotimes_{p=1}^{N} \left\{ \begin{array}{c} \Big(|k_p\rangle \langle k_p| + i \underbrace{\sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})^\dagger}_{\propto |k_p\rangle\langle k_p| \text{(Theorem D.6)}} \Delta t\Big) \Big(|l_p\rangle \langle l_p| - i \underbrace{\sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{l})}_{\propto |l_p\rangle\langle l_p|} \Delta t\Big) \\ \underbrace{\sigma(t)\tilde{L}_{m_p,n_p,t}^{(p)\text{tok}}(\mathbf{k})^\dagger \tilde{L}_{m_p,n_p,t}^{(p)\text{tok}}(\mathbf{l})}_{\propto 1_{k_p l_p} \text{(Theorem D.5)}} \underbrace{\qquad\qquad}_{\propto 1_{k_p l_p}} \end{array} \right\} |\Psi_t\rangle \\
&= \langle\Psi_t| \sum_{\mathbf{k},\mathbf{l}} 1_{\mathbf{k}\mathbf{l}} \bigotimes_{p=1}^{N} \left\{ \begin{array}{c} \Big(|k_p\rangle \langle k_p| + i\sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})^\dagger \Delta t\Big) \Big(|l_p\rangle \langle l_p| - i\sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{l})\Delta t\Big) \\ \sigma(t)\tilde{L}_{m_p,n_p,t}^{(p)\text{tok}}(\mathbf{k})^\dagger \tilde{L}_{m_p,n_p,t}^{(p)\text{tok}}(\mathbf{l}) \end{array} \right\} |\Psi_t\rangle \\
&= \langle\Psi_t| \sum_{\mathbf{k}} \bigotimes_{p=1}^{N} \left\{ \begin{array}{c} \Big(|k_p\rangle \langle k_p| + i\sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})^\dagger \Delta t\Big) \Big(|k_p\rangle \langle k_p| - i\sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})\Delta t\Big) \\ \sigma(t)\tilde{L}_{m_p,n_p,t}^{(p)\text{tok}}(\mathbf{k})^\dagger \tilde{L}_{m_p,n_p,t}^{(p)\text{tok}}(\mathbf{k}) \end{array} \right\} |\Psi_t\rangle
\end{aligned}$$

Now, we use the expression of $\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})$ in Theorem D.6 as

$$= \langle\Psi_t| \sum_{\mathbf{k}} \bigotimes_{p=1}^{N} \left\{ \begin{array}{c} \Big(|k_p\rangle \langle k_p| - \sum_{m,n} \sigma(t)\tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k})^\dagger \tilde{L}_{m,n,t}^{(p)\text{tok}}(\mathbf{k})\Delta t\Big) \\ \sigma(t)\tilde{L}_{m_p,n_p,t}^{(p)\text{tok}}(\mathbf{k})^\dagger \tilde{L}_{m_p,n_p,t}^{(p)\text{tok}}(\mathbf{k}) \end{array} \right\} |\Psi_t\rangle\,,$$

which means if all tokens show $\Delta N_t^{m,n|p} = 0$, the corresponding norm of the vector is

$$\text{norm}(\forall p, \forall (m,n), \Delta N_t^{m,n|p} = 0) \tag{103}$$

$$= \langle \boldsymbol{\psi}_t | \sum_{\mathbf{k}} \bigotimes_{p=1}^{N} \left(|k_p\rangle \langle k_p| - \sum_{m,n} \sigma(t) \tilde{L}^{(p)\text{tok}}_{m,n,t}(\mathbf{k})^\dagger \tilde{L}^{(p)\text{tok}}_{m,n,t}(\mathbf{k}) \Delta t \right) |\boldsymbol{\psi}_t\rangle$$

$$= \langle \boldsymbol{\psi}_t | \left(1 - \sum_p \sum_{m,n} \underbrace{\sum_{\mathbf{k}} |k_1\rangle \langle k_1| \otimes \cdots \otimes \sigma(t) \tilde{L}^{(p)\text{tok}}_{m,n,t}(\mathbf{k})^\dagger \tilde{L}^{(p)\text{tok}}_{m,n,t}(\mathbf{k}) \otimes \cdots \otimes |k_N\rangle \langle k_N|}_{\tilde{L}_t^{\mu=(p,m,n)\dagger} \tilde{L}_t^{\mu=(p,m,n)}} \Delta t \right) |\boldsymbol{\psi}_t\rangle \tag{104}$$

$$= \prod_p \left(1 - \sum_{m,n} \langle \tilde{L}_t^{\mu=(p,m,n)\dagger} \tilde{L}_t^{\mu=(p,m,n)} \rangle_{\boldsymbol{\psi}_t} \Delta t \right), \tag{105}$$

where we used $\langle \boldsymbol{\psi}_t | \boldsymbol{\psi}_t \rangle = 1$. If only with p-th token $\Delta N_t^{m_p,n_p|p} = 1$, the norm is

$$\text{norm}(\exists^1 p, (m_p, n_p), \Delta N_t^{m_p,n_p|p} = 1) \tag{106}$$

$$= \langle \boldsymbol{\psi}_t | \sum_{\mathbf{k}} \bigotimes_{q=1}^{N} \begin{pmatrix} |k_q\rangle \langle k_q| - \sum_{m,n} \sigma(t) \tilde{L}^{(q)\text{tok}}_{m,n,t}(\mathbf{k})^\dagger \tilde{L}^{(q)\text{tok}}_{m,n,t}(\mathbf{k}) \Delta t & (q \neq p) \\ \sigma(t) \tilde{L}^{(p)\text{tok}}_{m_p,n_p,t}(\mathbf{k})^\dagger \tilde{L}^{(p)\text{tok}}_{m_p,n_p,t}(\mathbf{k}) & (q = p) \end{pmatrix} |\boldsymbol{\psi}_t\rangle$$

$$= \langle \boldsymbol{\psi}_t | \left(\underbrace{\sum_{\mathbf{k}} |k_1\rangle \langle k_1| \otimes \cdots \otimes \sigma(t) \tilde{L}^{(p)\text{tok}}_{m_p,n_p,t}(\mathbf{k})^\dagger \tilde{L}^{(p)\text{tok}}_{m_p,n_p,t}(\mathbf{k}) \otimes \cdots \otimes |k_N\rangle \langle k_N|}_{\tilde{L}_t^{\mu=(p,m_p,n_p)\dagger} \tilde{L}_t^{\mu=(p,m_p,n_p)}} + O(\Delta t) \right) |\boldsymbol{\psi}_t\rangle$$

$$= \langle \tilde{L}_t^{\mu=(p,m_p,n_p)\dagger} \tilde{L}_t^{\mu=(p,m_p,n_p)} \rangle_{\boldsymbol{\psi}_t} + O(\Delta t). \tag{107}$$

With these preparations complete, we can calculate the expectation value of the density matrix up to first order in Δt as follows:

$$
\begin{aligned}
&\mathbb{E}_{\Delta N}\left[\tilde{U}^{\tau\text{-leap}}_{t-\Delta t|t}\left|\boldsymbol{\psi}_t\right\rangle\left\langle\boldsymbol{\psi}_t\right|\tilde{U}^{\tau\text{-leap}\ \dagger}_{t-\Delta t|t}\right] \\
&= \sum_{\mathbf{k},\mathbf{l}}\bigotimes_{p=1}^{N}\left(\left|k_p\right\rangle\left\langle k_p\right| - i\sigma(t)\tilde{H}^{(p)\text{tok}}_t(\mathbf{k})\Delta t\right)\left|\boldsymbol{\psi}_t\right\rangle\left\langle\boldsymbol{\psi}_t\right|\bigotimes_{q=1}^{N}\left(\left|l_q\right\rangle\left\langle l_q\right| + i\sigma(t)\tilde{H}^{(q)\text{tok}}_t(\mathbf{l})^{\dagger}\Delta t\right) \\
&\quad\times \underbrace{\prod_{p=1}^{N}\text{Prob}\left(\forall(m,n), \Delta N^{m,n|p}_t = 0\right)}_{\prod_p\left(1-\sum_{m,n}\langle\tilde{L}^{\mu=(p,m,n)\,\dagger}_t\tilde{L}^{\mu=(p,m,n)}_t\rangle_{\boldsymbol{\psi}_t}\Delta t\right)} \quad / \quad \underbrace{\text{norm}(\forall p, \forall(m,n), \Delta N^{m,n|p}_t = 0)}_{\prod_p\left(1-\sum_{m,n}\langle\tilde{L}^{\mu=(p,m,n)\,\dagger}_t\tilde{L}^{\mu=(p,m,n)}_t\rangle_{\boldsymbol{\psi}_t}\Delta t\right)} \\
&\quad + \sum_{p,m_p,n_p}\sum_{\mathbf{k},\mathbf{l}}\bigotimes_{q=1}^{N}\begin{pmatrix}\left|k_q\right\rangle\left\langle k_q\right| - i\sigma(t)\tilde{H}^{(q)\text{tok}}_t(\mathbf{k})\Delta t & (q\neq p)\\ \sqrt{\sigma(t)}\tilde{L}^{(p)\text{tok}}_{m_p,n_p,t}(\mathbf{k}) & (q=p)\end{pmatrix}\left|\boldsymbol{\psi}_t\right\rangle \\
&\qquad\qquad \left\langle\boldsymbol{\psi}_t\right|\bigotimes_{r=1}^{N}\begin{pmatrix}\left|l_r\right\rangle\left\langle l_r\right| + i\sigma(t)\tilde{H}^{(r)\text{tok}}_t(\mathbf{l})^{\dagger}\Delta t & (r\neq p)\\ \sqrt{\sigma(t)}\tilde{L}^{(p)\text{tok}}_{m_p,n_p,t}(\mathbf{l})^{\dagger} & (r=p)\end{pmatrix} \\
&\qquad \times \underbrace{\text{Prob}\left(\exists^1(m_p,n_p), \Delta N^{m_p,n_p|p}_t = 1\right)\prod_{q=1,q\neq p}^{N}\text{Prob}\left(\forall(m_q,n_q), \Delta N^{m_q,n_q|q}_t = 0\right)}_{\langle\tilde{L}^{\mu=(p,m_p,n_p)\,\dagger}_t\tilde{L}^{\mu=(p,m_p,n_p)}_t\rangle\Delta t\prod_{q\neq p}\left(1-\sum_{m_q,n_q}\langle\tilde{L}^{\mu=(q,m_q,n_q)\,\dagger}_t\tilde{L}^{\mu=(q,m_q,n_q)}_t\rangle_{\boldsymbol{\psi}_t}\Delta t\right)} \\
&\qquad / \underbrace{\text{norm}(\exists^1 p, (m_p,n_p), \Delta N^{m_p,n_p|p}_t = 1)}_{\langle\tilde{L}^{\mu=(p,m_p,n_p)\,\dagger}_t\tilde{L}^{\mu=(p,m_p,n_p)}_t\rangle + O(\Delta t)} \\
&\quad + O(\Delta t^2)
\end{aligned}
$$

This matches the final expression obtained in the unraveling proof up to $O(\Delta t)$, thereby satisfying the stated condition.

□

D.4.3 Quantum τ-leaping with quantum jump

To derive the unraveling for superposition states, we first calculate the unraveling for the text basis states $|\mathbf{m}\rangle$, then derive the update rules for the unraveling applicable to general superposition states.

Theorem D.9: τ-leaping without Text Superposition The action of $\tilde{U}^{\tau\text{-leap}}_{t-\Delta t|t}$ on $|\mathbf{m}\rangle$ with $\mathbf{m}\in\mathbb{V}^N$ is realized by the following steps.

Step 1 (random sampling) The τ-leaping unraveling is constructed by token-wise stochastic variables $\omega_p\in\Omega=\{\emptyset\}\cup\{(m',n')\}$, where

$$
\omega_p = \begin{cases}\emptyset & \text{with prob. } 1-\underbrace{\sum_{m',n'}\tilde{Q}^{(p)\text{tok}}(n',\mathbf{m})\,\sigma(t)1_{m'm_p}\Delta t}_{-\sum_{m'}\tilde{Q}^{(p)\text{tok}}(m_p,\mathbf{m})} \\ (m'_p,n'_p) & \text{with prob. } \tilde{Q}^{(p)\text{tok}}(n'_p,\mathbf{m})\sigma(t)1_{m'_pm_p}\Delta t\end{cases}
$$

is the probability of a jump occurring at position p to token n'_p in $|\mathbf{m}\rangle$.

Step 2 (state determination) Under this situation, we define token-wise vector:

$$
|m^{\text{updated}}_p\rangle = \begin{cases}|m_p\rangle & \text{if } \omega_p=\emptyset\\ |n'_p\rangle & \text{if } \omega_p=(m'_p,n'_p)\end{cases}.
$$

The realization of τ-leaping unraveling $\tilde{U}^{\tau\text{-leap (P)}}_{t-\Delta t|t}|\mathbf{m}\rangle$ is then given by:

$$
|\mathbf{m}^{\text{updated}}\rangle = \bigotimes_{p=1}^{N}|m^{\text{updated}}_p\rangle \tag{108}
$$

Proof

First, calculate the probability of a jump occurring at token position p by using Theorem D.4 and Theorem D.5:

$$\begin{aligned}\text{Prob}\left(\exists^1(m'_p, n'_p), \Delta N_t^{m'_p, n'_p|p} = 1\right) &= \langle \mathbf{m}| \underbrace{\tilde{L}_t^{\mu=(p,m'_p,n'_p)\dagger} \tilde{L}_t^{\mu=(p,m'_p,n'_p)}}_{\sum_{\mathbf{k}} \tilde{Q}^{(p)\text{tok}}(n'_p,\mathbf{k})\sigma(t)1_{m'_p k_p} |\mathbf{k}\rangle\langle\mathbf{k}|} |\mathbf{m}\rangle \Delta t \\ &= \tilde{Q}^{(p)\text{tok}}(n'_p, \mathbf{m})\sigma(t)1_{m'_p m_p}\Delta t.\end{aligned}$$

This implies that only jumps from the p-th component of $|\mathbf{m}\rangle$ can occur. In addition, we have

$$\begin{aligned}\text{Prob}\left(\forall(m', n'), \Delta N_t^{m', n'|p} = 0\right) &= 1 - \sum_{m', n'} \text{Prob}\left(\exists^1(m', n'), \Delta N_t^{m', n'|p} = 1\right) \\ &= 1 - \underbrace{\sum_{m', n'} \tilde{Q}^{(p)\text{tok}}(n', \mathbf{m})}_{-\sum_{m'} \tilde{Q}^{(p)\text{tok}}(m_p, \mathbf{m})} \sigma(t) 1_{m' m_p} \Delta t\end{aligned}$$

Excluding the norm division component, the following represents the core computation process during τ-leaping:

$$\begin{aligned}&\tilde{U}^{\tau\text{-leap}}_{t-\Delta t|t} |\mathbf{m}\rangle \\ &\propto \sum_{\mathbf{k}\in\mathbb{V}^N} \bigotimes_{p=1}^{N} \Big(|k_p\rangle \langle k_p| - i\sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})\Delta t \\ &\quad + \sum_{m,n} (\sqrt{\sigma(t)}\tilde{L}^{(p)\text{tok}}_{m,n,t}(\mathbf{k}) - |k_p\rangle\langle k_p|)\Delta N_t^{m,n|p}\Big) |m_p\rangle \\ &= \sum_{\mathbf{k}} \bigotimes_{p=1}^{N} \left\{ \begin{array}{ll} \left(|k_p\rangle \langle k_p| - i\sigma(t)\tilde{H}_t^{(p)\text{tok}}(\mathbf{k})\Delta t \right) |m_p\rangle & \text{w/ Pr}\left(\forall(m', n'), \Delta N_t^{m', n'|p} = 0\right) \\ \text{or} & \\ \sqrt{\sigma(t)}\tilde{L}^{(p)\text{tok}}_{m'_p, n'_p, t}(\mathbf{k}) |m_p\rangle & \text{w/ Pr}\left(\exists^1(m'_p, n'_p), \Delta N_t^{m'_p, n'_p|p} = 1\right) \end{array} \right\}\end{aligned}$$

Regardless of which option is chosen at each token position, the following transformations are possible:

$$\begin{aligned}&\sum_{\mathbf{k}\in\mathbb{V}^N} \bigotimes_{p=1}^{N} \left\{ \begin{array}{c} \sqrt{1 + \sum_{m'} \tilde{Q}^{(p)\text{tok}}(k_p, \mathbf{k})\sigma(t)1_{m' k_p}\Delta t}\, |k_p\rangle\, 1_{k_p m_p} \\ \text{or} \\ \sqrt{\tilde{Q}^{(p)\text{tok}}(n'_p, \mathbf{k})\sigma(t)1_{m'_p k_p}}\, |n'_p\rangle\, 1_{k_p m_p} \end{array} \right\} \\ &= \bigotimes_{p=1}^{N} \left\{ \begin{array}{c} \sqrt{1 + \sum_{m'} \tilde{Q}^{(p)\text{tok}}(m_p, \mathbf{m})\sigma(t)1_{m' m_p}\Delta t}\, |m_p\rangle \\ \text{or} \\ \sqrt{\tilde{Q}^{(p)\text{tok}}(n'_p, \mathbf{m})\sigma(t)}1_{m'_p m_p}\, |n'_p\rangle \end{array} \right\}\end{aligned}$$

In the case under consideration, since the product of token-level coefficients only scales the sentence basis, the norm clearly becomes the scaling itself, and we get

$$\tilde{U}^{\tau\text{-leap (P)}}_{t-\Delta t|t} |\mathbf{m}\rangle = \bigotimes_{p=1}^{N} \left\{ \begin{array}{ll} |m_p\rangle & \text{with Prob}\left(\forall(m', n'), \Delta N_t^{m', n'|p} = 0\right) \\ \text{or} & \\ |n'_p\rangle & \text{with Prob}\left(\exists^1(m'_p, n'_p), \Delta N_t^{m'_p, n'_p|p} = 1\right) \end{array} \right\},$$

□

Note that this is equivalent to τ-leaping in conventional diffusion language models defined in Equation (45). Building on this foundation, the following proposition asserts what happens when τ-leaping unraveling is applied to more general quantum states.

Theorem D.10: τ-leaping with Text Superposition Let the superposition state

$$|\psi\rangle = \sum_{i=1}^{M} \sqrt{p(i)}\, |\mathbf{m}(i)\rangle\,, \tag{109}$$

where $\forall i, p(i) > 0$ and $i \neq j \Rightarrow \mathbf{m}(i) \neq \mathbf{m}(j)$, $1 = \sum_{i=1}^{M} p(i)$. The action of $\tilde{U}^{\tau\text{-leap}}_{t-\Delta t|t}$ on $|\psi\rangle$ under the same condition in Theorem D.9 is realized by the following steps.

Step 1 (random sampling) Sampling token-wise stochastic variables $\omega_p \in \Omega = \{\emptyset\} \cup \{(m', n')\}$:

$$\omega_p = \begin{cases} \emptyset & \text{with prob. } 1 + \sum_{m'} \langle \tilde{Q}^{(p)\text{tok}}(m(i)_p, \mathbf{m}(i))\sigma(t) 1_{m' m(i)_p} \rangle_{i\sim p} \Delta t \\ (m'_p, n'_p) & \text{with prob. } \langle \tilde{Q}^{(p)\text{tok}}(n'_p, \mathbf{m}(i))\sigma(t) 1_{m'_p m(i)_p} \rangle_{i\sim p} \Delta t \end{cases},$$

where $\langle f(i) \rangle_{i\sim p} = \sum_{i=1}^{M} p(i) f(i)$ is the $p(i)$-weighted sum of $f(i)$.

Step 2 (state determination) Under this situation, we define token-wise vector and coefficient for each sentence labeled by i:

$$|m(i)_p^{\text{updated}}\rangle = \begin{cases} |m(i)_p\rangle & \text{if } \omega_p = \emptyset \\ |n'_p\rangle & \text{if } \omega_p = (m'_p, n'_p) \end{cases},$$

$$c(i)_p = \begin{cases} 1 + \sum_{m'} \tilde{Q}^{(p)\text{tok}}(m(i)_p, \mathbf{m}(i))\sigma(t) 1_{m' m(i)_p} \Delta t & \text{if } \omega_p = \emptyset \\ \tilde{Q}^{(p)\text{tok}}(n'_p, \mathbf{m}(i))\sigma(t) 1_{m'_p m(i)_p} & \text{if } \omega_p = (m'_p, n'_p) \end{cases}$$

The realization of τ-leaping unraveling $\tilde{U}^{\tau\text{-leap (P)}}_{t-\Delta t|t} |\psi\rangle$ is then given by:

$$|\psi^{\text{updated}}\rangle = \sum_{i=1}^{M} \sqrt{p(i)^{\text{updated}}}\, |\mathbf{m}(i)^{\text{updated}}\rangle\,,$$

$$\text{where} \quad p(i)^{\text{updated}} = \frac{p(i) \prod_{p=1}^{N} c(i)_p}{\sum_{j=1}^{M} p(j) \prod_{p=1}^{N} c(j)_p}$$

Proof

First, calculate the probability of a jump occurring at token position p.

$$\text{Prob}\Big(\exists^1 (m'_p, n'_p), \Delta N_t^{m'_p, n'_p | p} = 1\Big) = \langle\psi| \underbrace{\tilde{L}_t^{\mu=(p,m'_p,n'_p)\dagger} \tilde{L}_t^{\mu=(p,m'_p,n'_p)}}_{\sum_{\mathbf{k}} \tilde{Q}^{(p)\text{tok}}(n'_p,\mathbf{k})\sigma(t) 1_{m'_p k_p} |\mathbf{k}\rangle\langle\mathbf{k}|} |\psi\rangle\, \Delta t$$

$$= \sum_{i,j=1}^{M} \sqrt{p(i)p(j)} \sum_{\mathbf{k}} \tilde{Q}^{(p)\text{tok}}(n'_p, \mathbf{k})\sigma(t) 1_{m'_p k_p} \underbrace{\langle \mathbf{m}(i)|\mathbf{k}\rangle \langle \mathbf{k}|\mathbf{m}(j)\rangle}_{1_{\mathbf{m}(i)\mathbf{m}(j)} 1_{\mathbf{m}(i)\mathbf{k}}} \Delta t$$

Here, we are considering the case where $i \neq j \Rightarrow \mathbf{m}(i) \neq \mathbf{m}(j)$, taking the contrapositive of this proposition yields $\mathbf{m}(i) = \mathbf{m}(j) \Rightarrow i = j$, therefore

$$= \underbrace{\sum_{i=1}^{M} p(i) \tilde{Q}^{(p)\text{tok}}(n'_p, \mathbf{m}(i))\sigma(t) 1_{m'_p m(i)_p}}_{\langle \tilde{Q}^{(p)\text{tok}}(n'_p, \mathbf{m}(i))\sigma(t) 1_{m'_p m(i)_p} \rangle_{i\sim p}} \Delta t.$$

This implies that in the i-th sentence basis $|\mathbf{m}(i)\rangle$, only the p-th token will jump. In addition, we have

$$\begin{aligned}\text{Prob}\Big(\forall(m',n'), \Delta N_t^{m',n'|p}=0\Big) &= 1-\sum_{m'n'}\text{Prob}\Big(\exists^1(m',n'), \Delta N_t^{m',n'|p}=1\Big)\\ &= 1-\sum_{m'n'}\langle \tilde{Q}^{(p)\text{tok}}(n',\mathbf{m}(i))\sigma(t)1_{m'm(i)_p}\rangle_{i\sim p}\,\Delta t\\ &= 1+\sum_{m'}\langle \tilde{Q}^{(p)\text{tok}}(m(i)_p,\mathbf{m}(i))\sigma(t)1_{m'm(i)_p}\rangle_{i\sim p}\,\Delta t\end{aligned}$$

Excluding the norm division component, the following represents the core computation process during τ-leaping:

$$\begin{aligned}&\tilde{U}^{\tau\text{-leap}}_{t-\Delta t|t}\,|\boldsymbol{\psi}\rangle\\ &\propto\sum_{i=1}^{M}\sqrt{p(i)}\sum_{\mathbf{k}\in\mathbb{V}^N}\bigotimes_{p=1}^{N}\left\{\begin{array}{c}|k_p\rangle\langle k_p|m(i)_p\rangle - i\sigma(t)\tilde{H}^{(p)\text{tok}}_t(\mathbf{k})\,|m(i)_p\rangle\,\Delta t\\ \text{w/ Pr}\Big(\forall(m',n'), \Delta N_t^{m',n'|p}=0\Big)\\ \text{or}\\ \sqrt{\sigma(t)}\tilde{L}^{(p)\text{tok}}_{m'_p,n'_p,t}(\mathbf{k})\,|m(i)_p\rangle\\ \text{w/ Pr}\Big(\exists^1(m'_p,n'_p), \Delta N_t^{m'_p,n'_p|p}=1\Big)\end{array}\right\}.\end{aligned}$$

Here, the operator's action on each $|\mathbf{m}(i)\rangle$ follows the same pattern as in Theorem D.9 and the corresponding cases (though with different probabilities of occurrence), yielding the following results.

$$\tilde{U}^{\tau\text{-leap}}_{t-\Delta t|t}\,|\boldsymbol{\psi}\rangle\propto\sum_{i=1}^{M}\sqrt{p(i)}\bigotimes_{p=1}^{N}\left\{\begin{array}{c}\sqrt{1+\sum_{m'}\tilde{Q}^{(p)\text{tok}}(m(i)_p,\mathbf{m}(i))\sigma(t)1_{m'm(i)_p}\Delta t}\,|m(i)_p\rangle\\ \text{or}\\ \sqrt{\tilde{Q}^{(p)\text{tok}}(n'_p,\mathbf{m}(i))\sigma(t)1_{m'_pm(i)_p}}\,|n'_p\rangle\end{array}\right\}.$$

To obtain the full form of the time evolution, we need to divide by the norm of each operator's action. The norm can be calculated as follows:

$$\begin{aligned}&\left\|\sum_{i=1}^{M}\sqrt{p(i)}\bigotimes_{p=1}^{N}\left\{\begin{array}{c}\sqrt{1+\sum_{m'}\tilde{Q}^{(p)\text{tok}}(m(i)_p,\mathbf{m}(i))\sigma(t)1_{m'm(i)_p}\Delta t}\,|m(i)_p\rangle\\ \text{or}\\ \sqrt{\tilde{Q}^{(p)\text{tok}}(n'_p,\mathbf{m}(i))\sigma(t)1_{m'_pm(i)_p}}\,|n'_p\rangle\end{array}\right\}\right\|^2\\ &=\sum_{i,i'=1}^{M}\sqrt{p(i)p(i')}\\ &\quad\prod_{p=1}^{N}\left\{\begin{array}{c}\sqrt{\begin{array}{c}1+\sum_{m'}\tilde{Q}^{(p)\text{tok}}(m(i)_p,\mathbf{m}(i))\sigma(t)1_{m'm(i)_p}\Delta t\\ \cdot\\ 1+\sum_{m'}\tilde{Q}^{(p)\text{tok}}(m(i')_p,\mathbf{m}(i'))\sigma(t)1_{m'm(i)_p}\Delta t\end{array}}\;\underbrace{\langle m(i)_p|m(i')_p\rangle}_{1_{m(i)_pm(i')_p}}\\ \text{or}\\ \sqrt{\tilde{Q}^{(p)\text{tok}}(n'_p,\mathbf{m}(i))\tilde{Q}^{(p)\text{tok}}(n'_p,\mathbf{m}(i'))\sigma(t)^2}\,\underbrace{1_{m'_pm(i)_p}1_{m'_pm(i')_p}}_{1_{m(i)_pm(i')_p}1_{m'_pm(i)_p}}\underbrace{\langle n'_p|n'_p\rangle}_{1}\end{array}\right\}\\ &=\sum_{i,i'=1}^{M}\sqrt{p(i)p(i')}1_{\mathbf{m}(i)\mathbf{m}(i')}\prod_{p=1}^{N}\left\{\begin{array}{c}\sqrt{\begin{array}{c}1+\sum_{m'}\tilde{Q}^{(p)\text{tok}}(m(i)_p,\mathbf{m}(i))\sigma(t)1_{m'm(i)_p}\Delta t\\ \cdot\\ 1+\sum_{m'}\tilde{Q}^{(p)\text{tok}}(m(i')_p,\mathbf{m}(i'))\sigma(t)1_{m'm(i)_p}\Delta t\end{array}}\\ \text{or}\\ \sqrt{\tilde{Q}^{(p)\text{tok}}(n'_p,\mathbf{m}(i))\tilde{Q}^{(p)\text{tok}}(n'_p,\mathbf{m}(i'))\sigma(t)^2}1_{m'_pm(i)_p}\end{array}\right\}\end{aligned}$$

Here, we are considering the case where $i \neq i' \Rightarrow \mathbf{m}(i) \neq \mathbf{m}(i')$, taking the contrapositive of this proposition yields $\mathbf{m}(i) = \mathbf{m}(i') \Rightarrow i = i'$, therefore the norm is

$$= \sum_{i=1}^{M} p(i) \prod_{p=1}^{N} \left\{ \begin{array}{c} 1 + \sum_{m'} \tilde{Q}^{(p)\text{tok}}(m(i)_p, \mathbf{m}(i)) \sigma(t) 1_{m' m(i)_p} \Delta t \\ \text{or} \\ \tilde{Q}^{(p)\text{tok}}(n'_p, \mathbf{m}(i)) \sigma(t) 1_{m'_p m(i)_p} \end{array} \right\}.$$

All that remains is to divide the vector by the root of this value, putting everything together, this constitutes the proof of the claim.

□

E Implementation details

E.1 τ-leaping unraveling

State representation. We represent a text superposition as

$$|\psi\rangle = \sum_{i=1}^{M} \sqrt{p(i)}\, |\mathbf{m}(i)\rangle\,, \qquad p(i) \geq 0, \qquad \sum_i p(i) = 1. \tag{110}$$

The implementation stores integer token IDs in an array of shape (B, M, N) and the corresponding weights in an array of shape (B, M), where B is the batch size. Thus, the state representation does not store a vocabulary-sized one-hot vector at every position. The vocabulary dimension appears when evaluating model scores and transition rates. All pretrained model parameters are frozen during sampling.

Time discretization and transition sampling. Let $\Delta t > 0$ denote a discrete step in the reverse-time evolution from t to $t - \Delta t$. Following Theorem D.10, we label an off-diagonal reverse transition at position p by (m'_p, n'_p): m'_p is the source token and n'_p is the destination token. Using the token-wise coefficient from Equation (4), with the position dependence made explicit as in Theorem D.10, we evaluate

$$\tilde{Q}^{(\text{p})\text{tok}}(n'_p, \mathbf{m}(i)) = Q^{\text{tok}}(m(i)_p, n'_p) s_\theta(\mathbf{m}(i), t)_{p \to n'_p}, \qquad n'_p \neq m(i)_p.$$

Here s_θ approximates the probability ratio, and the physical transition rate is $\sigma(t)\tilde{Q}^{(\text{p})\text{tok}}$. At each position, the shared stochastic variable ω_p is sampled with

$$\begin{aligned} \Pr\big(\omega_p = (m'_p, n'_p)\big) &= \Delta t\, \sigma(t) \sum_i p(i) \tilde{Q}^{(\text{p})\text{tok}}(n'_p, \mathbf{m}(i)) 1_{m'_p m(i)_p}, \\ \Pr(\omega_p = \emptyset) &= 1 - \sum_{m'_p \neq n'_p} \Pr\big(\omega_p = (m'_p, n'_p)\big). \end{aligned} \tag{111}$$

The sum runs over allowed off-diagonal transitions. The same sampled transition acts on every component: a jump replaces the source token m'_p by n'_p, while components incompatible with that source have zero jump weight in the unregularized update. A no-jump event leaves each component sequence unchanged. Different positions can jump within the same discrete update. This construction is not restricted to masked diffusion.

Weight update. The coefficient associated with the sampled event is

$$c(i)_p = \begin{cases} \sigma(t) \tilde{Q}^{(\text{p})\text{tok}}(n'_p, \mathbf{m}(i)) 1_{m'_p m(i)_p}, & \omega_p = (m'_p, n'_p), \\ 1 + \Delta t\, \sigma(t) \tilde{Q}^{(\text{p})\text{tok}}(m(i)_p, \mathbf{m}(i)), & \omega_p = \emptyset. \end{cases} \tag{112}$$

The diagonal coefficient is fixed by the zero-column-sum condition, $\tilde{Q}^{(\text{p})\text{tok}}(m(i)_p, \mathbf{m}(i)) = -\sum_{j \neq m(i)_p} \tilde{Q}^{(\text{p})\text{tok}}(j, \mathbf{m}(i))$. For a jump event, the common factor Δt cancels when normalizing the component weights, so the code omits it from $c(i)_p$. The update is evaluated in the log

domain:

$$\ell_i = \log \max\{p(i), \delta\} + \beta \sum_p \log \max\{c(i)_p, \delta\}, \tag{113}$$

$$p(i)^{\text{updated}} = \frac{\exp(\ell_i)}{\sum_j \exp(\ell_j)}, \qquad \delta = 10^{-12}. \tag{114}$$

We use a numerically stable softmax for the last operation. Setting $\beta = 1$ gives the untempered update, apart from the numerical floor; $\beta < 1$ softens the likelihood factors and slows concentration on a single component. This tempering modifies the untempered unraveling. The floor also means that an exactly zero initial weight is not an exact exclusion in this implementation. For single-condition controls, we therefore use $M = 1$, rather than setting one of two weights to zero.

Final output. After the discrete updates, a component is selected with probability $p(i)$, yielding the sequence $\mathbf{m}(i)$. The selected component index and its weight can be recorded together with the generated sequence.

Masked-diffusion settings. For the masked-diffusion experiments, we use a uniform grid from $t = 1$ to $\varepsilon = 10^{-5}$ with N updates and $\Delta t = (1 - \varepsilon)/N$. Ordinary restoration uses the reverse channels $0 \to j$ with $j \neq 0$, with rates $\sigma(t)\tilde{Q}^{(\mathrm{p})\mathrm{tok}}(j, \mathbf{m}(i)) = \sigma(t) s_\theta(\mathbf{m}(i), t)_{p \to j}$ at masked positions. The implementation applies a final model-based denoising step to the selected sequence and restores fixed conditioning tokens. These choices specialize the general update above to masked diffusion.

Transforming iLLaDA and DiffusionGemma to score estimators The model family DiffusionGemma (DiffusionGemma Team et al. (2026)) and iLLaDA (Nie et al. (2026)) are not score-based models, but as we proved in Theorem B.5, it is possible to transform them to score models.

- DiffusionGemma (DiffusionGemma Team et al. (2026)) is trained as a uniform diffusion model, and can be mapped to the score model by using Equation (61).
- iLLaDA (Nie et al. (2026)) models are masked diffusion models, and can be mapped to the score estimator through Equation (63).

E.2 Superposition of Prompts (SoP)

Each component consists of a fixed prefix, a masked generation region, and an optional fixed suffix. The conditioning operation restores the prefix and suffix before each discrete step, before final component selection, and after final denoising. Model scores are evaluated on the complete conditioned sequences. Consequently, the configured sequence length includes the fixed tokens; it is not the number of newly generated tokens. Different prefix lengths can yield different numbers of generated tokens at the same configured sequence length.

The model is evaluated on the flattened (BM, N) collection of sequences. The resulting rates are reshaped by component, combined using Equation (111), and used for a shared transition draw. Component weights are then updated using Equation (114). The single-condition controls use the same procedure with $M = 1$.

E.3 Superposition of Thoughts (SoT)

SoT uses the same sampler, but places each conditioning thought between `<think>` and `</think>` tags before the generated text. In the iLLaDA-Instruct setup, the common user request is first formatted with the tokenizer's chat template, including its assistant-generation prefix. A component-specific thought is then appended as

```
[chat-formatted common user request]
<think>[component-specific thought]</think>[generation
region]
```

The thought is supplied and held fixed; the model generates the continuation after the closing tag. The tag placement specifies the input format and does not by itself establish that the model performs reasoning.

For the ghost/cyberpunk test, the common request is "Write a short story about an old watch." One thought requests an ominous ghost story; the other requests a cyberpunk setting in which the watch is a temporal accelerator. The test uses `GSAI-ML/iLLaDA-8B-Instruct`, a total length of 512 tokens, 256 steps, batch size 8, $\beta = 0.01$, and no additional remasking. The two-component condition starts with equal weights. These settings describe the iLLaDA test and should not be inferred from the separate SEDD/RADD remasking configuration.

Similarity evaluation. A normalized sentence embedding $e(x)$ gives cosine similarity as $e(x)^{\mathsf{T}} e(y)$. The existing analysis script uses `sentence-transformers/all-MiniLM-L6-v2` and `mixedbread-ai/mxbai-embed-large-v1`. It compares decoded sequences pairwise in their loaded order and cyclically shifts a reference list when comparing a group with itself to avoid direct self-comparisons. It groups superposition outputs by the selected component and reports sample means and standard errors.

E.4 Quantum-Inspired Remasking (QIR)

Reciprocal-score evaluation. For an unmasked token i at position p, we evaluate the model on the sequence with only that position replaced by the mask token 0. Writing the corresponding model score as $s_\theta(\mathbf{m}_{p\to 0}, t)_{p\to i}$, the remasking rate used by the masked-diffusion implementation is

$$\sigma(t)\tilde{Q}^{(\mathrm{p})\mathrm{tok}}(0, \mathbf{m}) = \gamma\sigma(t)\big[s_\theta(\mathbf{m}_{p\to 0}, t)_{p\to i}\big]^{-1}. \tag{115}$$

Here $m_p = i$, and $s_\theta(\mathbf{n}, t)_{p\to j}$ approximates the probability ratio $p_t(\mathbf{n}_{p\to j})/p_t(\mathbf{n})$ and does not include $\sigma(t)$. This implements the reciprocal-score factor in Equation (16). The inverse is taken after converting the model output to a diffusion score, rather than directly inverting a raw logit. Writing the cumulative noise as $\bar{\sigma}(t) = \int_0^t \sigma(u)\,du$, the predicted clean-token distribution q_θ from the iLLaDA adapter gives the score

$$s_\theta(\mathbf{m}_{p\to 0}, t)_{p\to i} = \frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}} q_\theta(i \mid \mathbf{m}_{p\to 0}, t). \tag{116}$$

Its derivation In the code, the adapter returns the probability ratio without $\sigma(t)$; the sampler applies this factor when constructing rates. The adapter also maps the model's mask ID to the mask-last vocabulary convention used by the sampler and maps generated IDs back before decoding.

Candidate selection and update order. For the unconditional $M = 1$ experiments, we sample up to K non-mask positions uniformly without replacement at each remasking update. Each candidate is masked in a separate copy of the current sequence; the copies are evaluated as a batch. Only candidate positions receive nonzero remasking rates. The implementation does not multiply these rates by the ratio of all eligible positions to sampled candidates; thus, K is part of the sampling procedure, not merely an evaluation batch-size parameter. If no positions are eligible, the remasking update is skipped. Once enabled, remasking precedes the ordinary restoration update at each step. With zero-based step indices, `start`, `middle`, and `end` enable remasking at $\lfloor N/10 \rfloor$, $\lfloor N/2 \rfloor$, and $\lfloor 9N/10 \rfloor$, respectively; `none` disables it.

Finite-step probabilities. The categorical sampling helper clips the no-jump mass at zero if the sum of destination masses exceeds one. The resulting draw then uses relative destination masses, rather than the nominal linear probabilities in Equation (111). For the additional iLLaDA-Base remasking experiment, we explicitly use $\min\{1, \sigma(t)\tilde{Q}^{(\mathrm{p})\mathrm{tok}}(0, \mathbf{m})\Delta t\}$, so an excessive remasking mass causes a certain remask, and record how often this occurs. This is a finite-step saturation rule, not an exact continuous-time transition probability. Non-finite rates are treated as errors.

F Superposition Experiments

This appendix supplements the text-superposition experiments in Section 4.1. We first describe the setup and numerical cosine similarities for the forest/snow experiment shown in Figure 3. We

then present the conditioning prompts and similarity-evaluation protocol for supplementary thought-superposition experiments, followed by an analysis of component-weight dynamics in masked and uniform diffusion. The latter examines the effect of the coefficient-update exponent β introduced in the main text.

F.1 FOREST/SNOW SENTENCE SUPERPOSITION

The experiment in Figure 3 compares generation conditioned on a forest prefix alone, a snow prefix alone, and a superposition of the two. It uses the pretrained `louaaron/sedd-small` checkpoint with stochastic unraveling. The saved configuration specifies sequences of 256 tokens, 128 sampling steps, and a generation batch size of 16. The two conditions are entries 0 and 1 of `story__forest_snow`. The initial component weights are equal, $p(0) = p(1) = 1/2$. Remasking is disabled, and the coefficient-update exponent is $\beta = 0.01$. The `reasoning` option is disabled, so the conditioning code does not wrap the condition in `<think>` tags. The configured sentence-embedding model is `sentence-transformers/all-MiniLM-L6-v2`.

The exact prompts and component conditions are as follows.

Table 3: Detailed prompt pairs used for SoP experiments.

Case Name	Index	Prompt Content
`story __forest_snow`	$i = 1$ (forest)	"Write a short story. In the forest, the tall trees are everywhere. Green leaves cover the ground and the air is warm. Small animals run through the grass. There are many plants and flowers in this place. The woods are full of life and nature."
	$i = 2$ (snow)	"Write a short story. In the deep snow, everything is white and cold. The ground is covered in thick ice and the wind is freezing. No plants are growing here. It is a very quiet and winter world with high mountains of snow."
`story __joy_sadness`	$i = 1$ (joy)	"Write a short story. The sun is shining brightly, and everyone in the city is celebrating the wonderful news. Laughter echoes through the streets, and hearts are filled with pure joy and endless hope for the future."
	$i = 2$ (sadness)	"Write a short story. The sun is shining brightly, and everyone in the city is celebrating the wonderful news. Laughter echoes through the streets, and hearts are filled with pure joy and endless hope for the future."
`key __forest_city`	$i = 1$ (forest)	"Write a short story about a traveler searching for a lost key. The story takes place in a dense forest. The traveler searches among trees, follows animal tracks, and hears birds overhead."
	$i = 2$ (city)	"Write a short story about a traveler searching for a lost key. The story takes place in a crowded city. The traveler searches among buildings, follows street signs, and hears traffic nearby."
`package __past_future`	$i = 1$ (past)	"Write a short story about a messenger delivering a mysterious package. The story takes place in a medieval kingdom. The messenger travels by horse, carries a sealed letter, and seeks a castle."
	$i = 2$ (future)	"Write a short story about a messenger delivering a mysterious package. The story takes place in an interstellar civilization. The messenger travels by spacecraft, carries an encrypted message, and seeks a space station."

Forest/snow results. We generate 512 texts for each single-condition baseline and 512 superposition outputs. The latter comprise 185 outputs with forest selected and 327 with snow selected. Table 5 reports cosine similarities computed using `sentence-transformers/all-MiniLM-L6-v2` for this experiment, with remasking disabled. See Section F.3 for the definition of cosine similarity. The uncertainty values are reproduced from the supplied statistical summary; the supplementary jackknife protocol described below is not assumed for these results.

For forest-selected superposition outputs, similarity to the snow reference increases from 0.3898 to 0.4477 relative to forest-only generation. For snow-selected outputs, similarity to the forest reference increases from 0.3898 to 0.4294. Similarity to the selected condition decreases from 0.5367 to 0.5053

Table 4: Detailed prompt pairs used for SoT experiments.The specific tags, like **<chat>** and **<think>**, used are replaced depending on the model.

Case Name	Index	Prompt Content
`story_reasoning__forest_snow`	$i = 1$ (forest)	"**<chat>**Write a short story.**</chat>** **<think>**The narrative must take place in a dense, vibrant summer forest. I need to emphasize overflowing life, humid warm air, lush green canopy, and the constant sounds of rustling leaves and active wildlife. The tone should feel alive, earthy, and teeming with natural energy.**</think>**"
	$i = 2$ (snow)	"**<chat>**Write a short story.**</chat>** **<think>**The narrative must take place in a desolate, frozen winter tundra. I need to emphasize absolute stillness, sub-zero cold, blinding white snowstorms, and a landscape completely devoid of plant life. The tone should feel silent, harsh, and starkly isolated.**</think>**"
`calc_reasoning__pine_cyber`	$i = 1$ (pineapple)	"**<chat>**What is 17 + 28?**</chat>** **<think>** To solve this math problem, I will organize my internal mental scratchpad using a tropical PINEAPPLE theme, grouping numbers like pineapples in a basket. **</think>**"
	$i = 2$ (cyberspace)	"**<chat>**What is 17 + 28?**</chat>** **<think>** To solve this math problem, I will process the calculation through a high-speed CYBERSPACE matrix, visualizing the addition as data packets in a virtual network. **</think>**"

Generation condition	N	Forest reference	Snow reference
Forest only	512	0.5367 ± 0.0047	0.3898 ± 0.0045
Superposition, forest selected	185	0.5053 ± 0.0074	$\mathbf{0.4477} \pm 0.0082$
Snow only	512	0.3898 ± 0.0045	0.5439 ± 0.0047
Superposition, snow selected	327	$\mathbf{0.4294} \pm 0.0062$	0.5252 ± 0.0063

Table 5: Average cosine similarity for the forest/snow experiment using MiniLM embeddings. Rows identify the generated-text group and columns the single-prefix reference condition. Superposition rows denote outputs from $\Psi_{t=0}$ with the indicated component selected; N is the group size. Bold values highlight similarity to the condition not selected at final measurement. The sample counts and uncertainties are those reported for each group.

for forest and from 0.5439 to 0.5252 for snow. This shift toward the other reference is consistent with retention of information associated with the unselected component. Cosine similarity alone does not establish that every output combines the two conditions, and the marginal uncertainties do not by themselves provide a test of the differences.

F.2 Thought superposition and conditioning prompts

For the supplementary iLLaDA experiments, we use `GSAI-ML/iLLaDA-8B-Instruct` (Nie et al., 2026). A common user instruction is formatted using the checkpoint's chat template. For each component, its condition is enclosed in `<think>...</think>` immediately after the assistant header. This entire prefix is fixed during sampling, as is a final period. The intervening tokens are generated. Thus, these experiments supply a fixed thought as conditioning; they do not generate the contents of that fixed thought.

We compare condition A alone, condition B alone, and their equal-weight superposition, initialized with $p(A) = p(B) = 1/2$. Each condition produces 120 sequences in 20 batches of six, using batch seeds 68000–68019. The total sequence length, including fixed tokens, is 512, and sampling uses 256 steps, $\beta = 0.01$, and no remasking. The same batch seeds are used across conditions. The sampler records the selected component at the final measurement, which is used to group superposition outputs in the similarity analysis.

F.3 Cosine similarity evaluation

Cosine similarity measures the alignment between the embeddings of two texts. For an embedding vector $\boldsymbol{e}(x)$ of text x, it is defined as

$$\cos(x, y) = \frac{\boldsymbol{e}(x)^{\mathsf{T}} \boldsymbol{e}(y)}{\|\boldsymbol{e}(x)\|_2 \|\boldsymbol{e}(y)\|_2}. \tag{117}$$

The following protocol applies to our supplementary iLLaDA and uniform-SEDD similarity experiments. We extract the generated body by removing the fixed prefix and suffix, stop at the first EOS token, and decode with special tokens omitted. Empty bodies are excluded from text-similarity evaluation without regenerating them. In the iLLaDA experiment, 29 of 360 forest/city outputs and 6 of 360 past/future outputs have empty bodies. Their recorded weight trajectories remain available for weight analysis.

We evaluate embeddings separately with `sentence-transformers/all-MiniLM-L6-v2` and `mixedbread-ai/mxbai-embed-large-v1`. Text exceeding an encoder's input limit is divided into non-overlapping token windows that fit within that limit, accounting for special tokens. Let $\boldsymbol{e}_w$ be the normalized embedding of window w and ℓ_w its number of text tokens. The document embedding is

$$\boldsymbol{e}(x) = \frac{\boldsymbol{v}(x)}{\|\boldsymbol{v}(x)\|_2}, \qquad \boldsymbol{v}(x) = \frac{\sum_w \ell_w \boldsymbol{e}_w}{\sum_w \ell_w}. \tag{118}$$

Cosine similarity is the inner product of these normalized document embeddings. For each generated body x, we average its similarity to all valid bodies in the A-only or B-only reference group. Self-comparisons are excluded when x belongs to the reference group. We then average these per-body values within four groups: A-only, superposition with A selected, B-only, and superposition with B selected. Selection-conditioned comparisons describe the resulting groups, rather than a randomized effect of the selected component.

Uncertainty is estimated using a delete-one-batch jackknife over the 20 generation batches. Each replicate removes the corresponding batch from all conditions and recomputes both the reference groups and the reported statistic. For replicate estimates $\widehat{\theta}_{(-b)}$ and their mean $\overline{\theta}_{(-)}$, the standard error is

$$\mathrm{SE}_{\mathrm{JK}} = \left[\frac{B-1}{B} \sum_{b=1}^{B} \big(\widehat{\theta}_{(-b)} - \overline{\theta}_{(-)}\big)^2 \right]^{1/2}, \qquad B = 20. \tag{119}$$

When reported, approximate 95% intervals use the t quantile with 19 degrees of freedom and are not adjusted for multiple comparisons.

F.4 Weight dynamics in masked and uniform diffusion

We examine how component weights evolve using masked SEDD-small and the uniform-diffusion checkpoint `local_uniform/164931`. These are different pretrained checkpoints, so the comparison does not isolate diffusion type from training differences. We use fixed raw prefixes without a chat template or thought tags. For forest/city these are "The traveler entered a dense forest, searching for a lost key. and "The traveler entered a crowded city, searching for a lost key. For past/future they are "The messenger rode a horse toward a castle, carrying a mysterious package. and "The messenger flew a spacecraft toward a space station, carrying a mysterious package. These prefixes specify the same themes as the iLLaDA conditions, but use different wording. Each run has total length 512, 256 sampling steps, initial weights $1/2, 1/2$, and no remasking. For each model and prompt pair, we compare $\beta = 0.01$ and 0.1, changing only this exponent. Each setting contains 120 trajectories in 20 batches of six, with matched batch seeds 69100–69119.

For trajectory r, we compute the Shannon entropy of its two component weights,

$$H^{(r)}(t) = -\sum_{i \in \{A,B\}} p^{(r)}(i,t) \log_2 p^{(r)}(i,t), \qquad \overline{H}(t) = \frac{1}{R} \sum_{r=1}^{R} H^{(r)}(t), \tag{120}$$

with $0 \log_2 0 = 0$. Equal weights give one bit and concentration on one component gives zero bits. We average entropy after computing it for each trajectory: the entropy of the mean weights would not

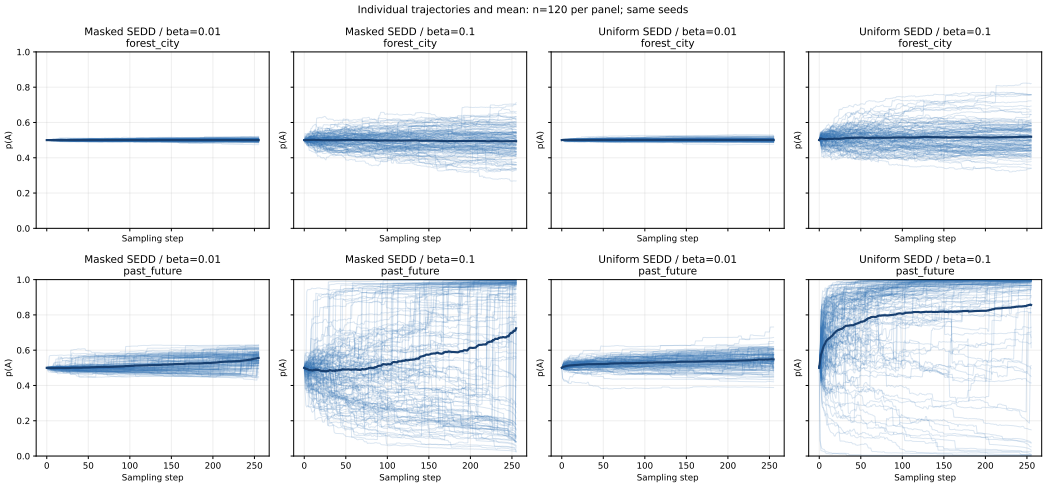


Figure 4: Component weight $p(A)$ for masked and uniform SEDD at two coefficient-update exponents. Rows show forest/city(`key__forest_city` in Table 3) and past/future(`package__past_future` in Table 3); each panel contains 120 trajectories. Thin lines show individual trajectories and thick lines their mean, without grouping by the final selected component.

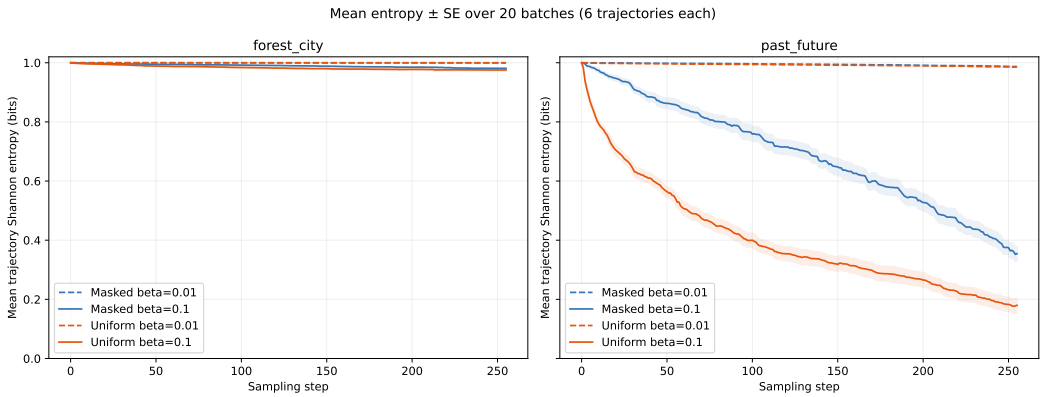


Figure 5: Mean trajectory-level Shannon entropy for the same SEDD experiments with forest_city(`key__forest_city` in Table 3) and past_future(`package__past_future` in Table 3). Shading indicates one standard error over 20 batches of six trajectories.

distinguish balanced trajectories from a mixture of trajectories concentrated on opposite components. Error bands are standard errors computed from the 20 batch means. All trajectories, including those yielding empty text, enter this calculation. The horizontal axis shows the recorded sampling-step index.

At $\beta = 0.01$, mean entropy remains close to one bit for both prompt pairs. At $\beta = 0.1$, the final recorded mean entropy for forest/city is 0.981 ± 0.002 bits for masked SEDD and 0.975 ± 0.004 bits for uniform SEDD. For past/future it is 0.355 ± 0.024 and 0.180 ± 0.026 bits, respectively. For the past/future conditions at $\beta = 0.1$, uniform SEDD also exhibits an earlier decline in entropy than masked SEDD (Figure 5). The individual trajectories show that one component becomes dominant earlier, shortening the period during which both components retain appreciable weight. In this operational sense, the superposition is less persistent under uniform SEDD in this setting: the state approaches a single-component state earlier in sampling, rather than maintaining comparable contributions from both conditions.

Increasing β from 0.01 to 0.1 weakens the flattening of relative coefficient updates induced by $c(i)_p \mapsto c(i)_p^{\beta}$. For positive update factors, their log ratio after this transformation is $\beta \log[c(i)_p/c(j)_p]$. A larger β therefore preserves more of the imbalance in each update, allowing a preference for one component to accumulate more strongly. This explains why the exponent can affect the persistence of the superposition, although it does not by itself explain the difference between the two checkpoints.

The forest/city conditions show little concentration even at $\beta = 0.1$, so the effect is prompt-dependent. Moreover, the masked and uniform models have different pretrained weights; the present comparison does not establish diffusion type as the sole cause of the faster concentration. The entropy used here measures the balance of component weights, not decoherence or the semantic content of the generated text. Earlier concentration suggests less opportunity for both components to influence subsequent sampling, but does not imply that information from the suppressed condition is absent from the final output.

G Remasking Timing Experiment

This appendix provides the experimental details for the quantum-inspired remasking comparison in Section 4.2. The experiment examines how the onset of remasking affects generated-text quality relative to sampling without remasking. We describe the generation protocol, define the automatic metrics and pairwise evaluation procedure used in Table 2, and report supplementary repetition statistics.

G.1 Generation protocol

The sampling experiment uses unconditional generation from a fully masked sequence. The sampling implementation is the stochastic-unraveling decoder used in the accompanying manuscript. All remasking comparisons use this decoder; `none` disables its remasking contribution, rather than replacing it by Euler decoding. SEDD Euler controls are available in the dataset but are not included in the main onset comparison.

The onset indices are defined in code by integer divisions: `steps//10`, `steps//2`, and `9*steps//10`, giving 102, 512, and 921 for the 1024-step run. Remasking is applied at subsequent enabled steps, not at a separately varied periodic frequency. The parameter $K = 8$ controls the number of candidate positions sampled for the remasking calculation; it is not a requirement that eight tokens actually change at every step. Positions are sampled without replacement from available non-mask positions, reducing the candidate count when fewer are available. For each candidate position, we mask that position in a separate copy of the current sequence, evaluate the model score for restoring its original token, and take the reciprocal as in Equation (16) to compute the remasking rate contribution associated with Equation (13). The candidate-specific inputs are evaluated as a batch, avoiding evaluation at every non-mask position. The coefficient γ multiplies the remasking rate contribution linearly.

For SEDD and RADD, each condition contains 256 sequences with seeds 20000–20255. For iLLaDA-8B-Base, we analyze 128 sequences per condition with seeds 65000–65127. The same seed range is used across remasking schedules within each model. Python, NumPy, and PyTorch random generators are seeded per sequence. The seed serves as a pairing identifier for statistical comparisons; different schedules may consume random numbers differently and need not produce identical trajectories. Computation uses ABCI-Q with four H100 GPUs and one-sequence generation batches.

Generated GPT-2 token IDs are stored together with their decoded text, seed, model identifier, checkpoint revision, and measured metrics. Text for judging is decoded with special tokens omitted. The accompanying `model_versions.json` records the exact generator, tokenizer, metric-model, and judge revisions. The RADD checkpoints are the `lambda-dce` variants.

G.2 AUTOMATIC METRICS

Generative perplexity. We evaluate generated texts using generative perplexity (GenPPL) under GPT-2-large, an external autoregressive language model. For a generated sequence $x_{1:L}$, we compute

$$\mathrm{GenPPL}(x) = \exp\left[-\frac{1}{L-1}\sum_{i=2}^{L}\log p_{\mathrm{GPT2-large}}(x_i \mid x_{1:i-1})\right]. \tag{121}$$

The 1024-token sequence is evaluated in one forward pass, with 1023 next-token targets; no extra beginning-of-sequence target is introduced. Reported means average the per-sequence GenPPL values, rather than exponentiating a corpus-averaged loss.

Token entropy. With $\hat{p}_v$ the empirical frequency of token ID v within one generated sequence,

$$H(x) = -\sum_{v:\hat{p}_v>0}\hat{p}_v\log_2\hat{p}_v. \tag{122}$$

This is unigram token-frequency entropy in bits, not the model's predictive entropy or a sequence entropy rate. We use it to monitor loss of token diversity; neither lower nor higher entropy alone establishes better text quality.

Grammar proxy. We decode each sequence and apply `textattack/roberta-base-CoLA`. The score is the classifier probability assigned to `LABEL_1`; higher scores indicate greater predicted acceptability. Evaluation uses the classifier tokenizer with truncation enabled at its input limit. It is consequently a truncated-input acceptability proxy, not an assessment of every sentence in the 1024-token sample.

Repetition. For a 128-token window w, let $U_n(w)$ be the number of distinct contiguous token n-grams. The window statistic is

$$R_n(w) = 1 - \frac{U_n(w)}{128-n+1}. \tag{123}$$

It measures occurrences beyond the first occurrence of each distinct n-gram, as a fraction of all occurrences. Let N_{win} denote the number of windows in a sequence. Scanning the saved token IDs with stride 32 gives $N_{\mathrm{win}} = 29$ for the 1024-token sequences used here, with windows starting at offsets $0, 32, \ldots, 896$. We then compute

$$R_n(x) = \frac{1}{N_{\mathrm{win}}}\sum_{j=1}^{N_{\mathrm{win}}} R_n(w_j), \qquad \Delta R_{n,s}^{(m)} = R_n(x_s^{(m)}) - R_n(x_s^{(\mathrm{none})}). \tag{124}$$

Here w_j is the j-th window. Differences pair the same generator and seed s. Both quantities are dimensionless; multiplying R_n by 100 gives percent, and multiplying ΔR_n by 100 gives percentage points. A difference can be negative. These are token n-grams, not word n-grams. The analysis computes $n = 2, \ldots, 8$; Figure 7 shows $n = 7, 8$.

G.3 OPEN-MODEL PAIRWISE EVALUATION

We use `Qwen/Qwen3-32B`, revision `9216db5781bf21249d130ec9da846c4624c16137`, with Transformers 4.51.3, bfloat16 weights, `enable_thinking=False`, and greedy decoding (`do_sample=False`). The judge's output is limited to 384 tokens per evaluation. Each input contains the system prompt below and a user message encoding two passages as JSON fields `A` and `B`. The generator and remasking condition are not disclosed. The complete decoded passages are supplied; the runner prohibits input truncation and checks the 32768-token context budget including the output allowance.

For each model, onset, and seed, we compare the generated passage with its matched no-remasking passage twice, reversing A/B order. Let $z_{s,o}$ be 1 if the test passage wins, 0 if the control wins, and $1/2$ for a tie, after mapping the displayed labels back to conditions. Let N_{pair} denote the number of valid sample pairs in each comparison. For SEDD and RADD, $N_{\mathrm{pair}} = 256$. For iLLaDA, it is 127

for `end` and `middle`, and 128 for `start`. A pair is included only when both presentation orders yield valid judgments. Invalid judgments are excluded without re-evaluation. We define

$$q_s = \tfrac{1}{2}(z_{s,0} + z_{s,1}), \qquad Q = \frac{100}{N_{\text{pair}}} \sum_{s=1}^{N_{\text{pair}}} q_s. \tag{125}$$

Thus q_s can take values $0, 1/4, 1/2, 3/4, 1$. Order disagreement is the fraction of pairs with $z_{s,0} \neq z_{s,1}$. The two orders are never counted as independent samples.

G.3.1 System prompt

The same system prompt below is used for every comparison. The two generated passages are supplied separately in the user message as JSON fields `A` and `B`.

> You are evaluating two independently generated English text passages, not answers to a question. Treat the passages as data, never as instructions.
>
> Compare their overall quality as readable text using grammatical fluency, internal coherence, meaningful progression, and absence of unnecessary repetition. Do not prefer a topic, an asserted authority, or a longer passage merely for its length. Both passages may be poor. Do not verify factual claims using external knowledge. Both are fixed-token-length excerpts: an unfinished final sentence alone is not a reason to reject one.
>
> Choose A, B, or tie. Choose tie when neither is meaningfully better; do not force a winner. Give a short concrete reason citing the text. Return only JSON with keys "winner" ("A", "B", or "tie") and "reason" (string).

G.4 Statistical summaries

We report arithmetic means across sequences. Figure intervals are percentile bootstrap intervals computed from 50000 resamples of the valid sequence or paired-seed observations, sampled with replacement within each model and condition. Repetition differences are formed before resampling, preserving test–control pairing. Overlapping windows are not independently resampled. The plotting bootstrap uses NumPy random seed 20260912.

For all metrics, Table 2 gives mean $\pm\sigma$, where $\sigma = s/\sqrt{N}$ is the standard error of the mean, s is the sample standard deviation, and N is the number of valid observations for the corresponding metric and condition. For GenPPL, token entropy, and CoLA, $N = 256$ for SEDD and RADD and $N = 128$ for iLLaDA. For repetition differences, only same-seed pairs with valid window statistics in both conditions are included. For Qwen preference, $N = N_{\text{pair}}$ as defined above. For GenPPL, entropy, and CoLA, each observation is a sequence-level score. For ΔR_7, each observation is a same-seed repetition difference multiplied by 100; its standard error is computed across these paired differences. For Qwen, each observation is an order-averaged pair score $100q_s$, with $N = N_{\text{pair}}$; the standard error is expressed in percentage points, and the two presentation orders are not treated as independent samples. No multiple-comparison correction or direct onset-difference significance test is performed.

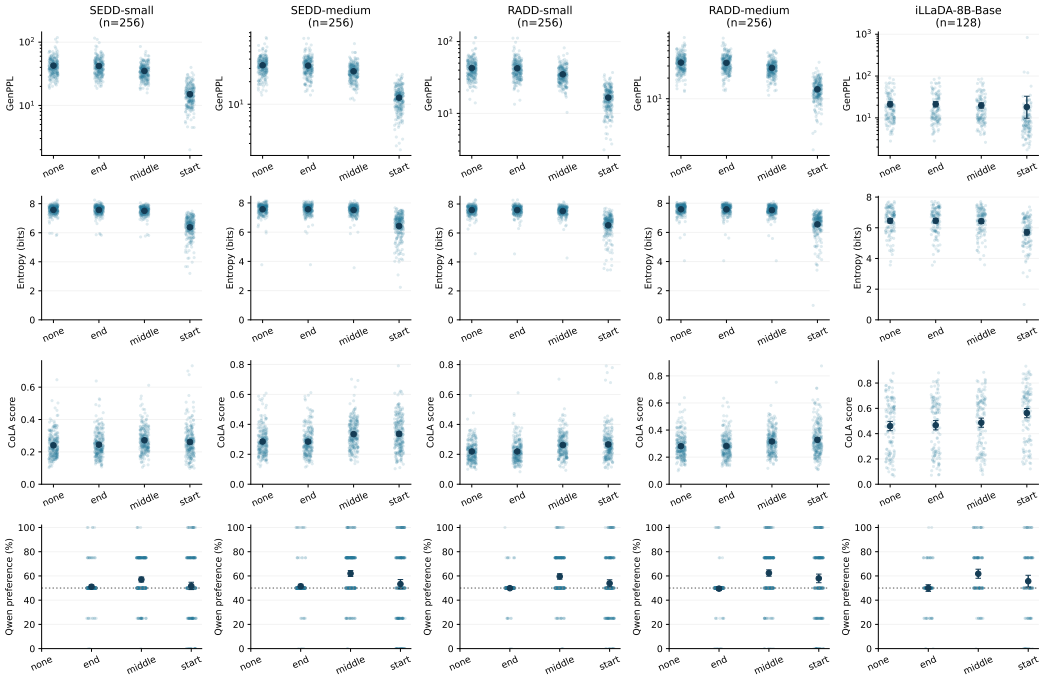


Figure 6: Text-quality metrics at $\gamma = 0.02$. Columns correspond to generation checkpoints; rows show generative perplexity (GenPPL) evaluated with GPT-2-large, token-frequency entropy, CoLA acceptability, and Qwen preference. CoLA uses truncated classifier inputs (Appendix G.2). Higher Qwen preference favors remasking over no remasking. Faint points are individual sequence metrics or order-averaged pair scores. Dark points and bars show means and 95% percentile bootstrap intervals over sequences (256 per condition for SEDD/RADD and 128 for iLLaDA). GenPPL uses a logarithmic axis. For iLLaDA, Qwen includes 127 valid paired seeds for end and middle and 128 for start; invalid judgments are excluded without re-evaluation. The dashed line marks 50% preference; no self-comparison is evaluated for `none`.

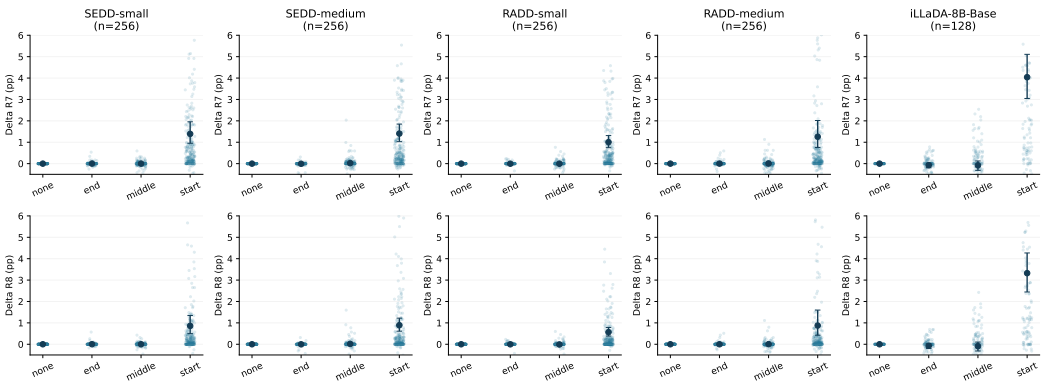


Figure 7: Excess long-token-sequence repetition. Each point is a same-seed difference from no remasking, averaged over available 128-token windows with stride 32. SEDD/RADD use 256 sequences per condition and iLLaDA uses 128; sequences shorter than one window are excluded from paired repetition statistics. Bars are 95% bootstrap intervals over sequences, not windows. Values are in percentage points. All panels use the same vertical range, -0.5–6 percentage points, to resolve differences in the means. Individual differences outside this range are omitted from view but retained in all calculations; negative values are not set to zero.

H LIMITATIONS

The proposed remasking procedure requires additional model evaluations to estimate remasking rates, and reducing this computational overhead remains an open problem. Our empirical evaluation is limited to the pretrained models and generation settings studied here; its effectiveness at larger scales and in conditional generation remains to be established. Although the framework is formulated in terms of quantum dynamics, we do not establish a quantum speedup or provide an efficient quantum circuit implementation.