STRATIFIED HAZARD SAMPLING: MINIMAL-VARIANCE EVENT SCHEDULING FOR CTMC/DTMC DISCRETE DIFFUSION AND FLOW MODELS

Seunghwan Jang, SooJean Han
Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea
{jsh991124, soojean}@kaist.ac.kr

ABSTRACT

Uniform-noise discrete diffusion and flow models (e.g., D3PM, SEDD, UDLM, DFM) generate sequences non-autoregressively by iteratively refining randomly initialized vocabulary tokens through multiple context-dependent replacements. These models are typically formulated as time-inhomogeneous CTMC/DTMC processes and sampled using independent Bernoulli change decisions at each discretization step. This induces Poisson-binomial variance in per-position jump counts that grows with the number of required edits, leading to the characteristic under-editing (residual noise) and over-editing (cascading substitutions) failure modes that degrade sample quality, especially under tight discretization budgets. In contrast, absorbing-state (mask-start) models avoid this instability by allowing each position to jump at most once.

We propose *Stratified Hazard Sampling* (SHS), a training-free, drop-in, and hyperparameter-free inference principle for any sampler that admits a stay-vs.-replace decomposition. SHS models per-token edits as events driven by cumulative hazard (CTMC) or cumulative jump mass (DTMC) and places events by stratifying this cumulative quantity: with a single random phase per position, a token is updated whenever its accumulated hazard crosses unit-spaced thresholds. This preserves the expected number of jumps while achieving the minimum possible conditional variance among unbiased integer estimators (bounded by $1/4$ for any fixed cumulative mass), without altering per-jump destination sampling and thus retaining multimodality. Experiments on uniform-noise discrete diffusion language models show that SHS consistently improves sample quality. We further show that SHS improves robustness under token-level blacklist filtering, with benefits increasing as lexical constraints grow more severe.

1 INTRODUCTION

Non-autoregressive generation for high-dimensional discrete data (e.g., text, code) is a promising approach that can significantly reduce inference latency by updating all tokens in parallel. A growing family of methods recasts discrete generation as a time-inhomogeneous Markov process (CTMC or DTMC) that transports an easy prior p_0 to the data distribution p_1, including discrete diffusion models (D3PM, SEDD, UDLM) and discrete flow matching (DFM). Among initialization strategies, absorbing-state (mask-start) methods (Austin et al., 2021; Campbell et al., 2022)—where each position unmasks exactly once—yield simple, low-variance trajectories but reduce generation to any-order parallel unmasking without iterative refinement. In contrast, uniform-noise initialization (Austin et al., 2021; Lou et al., 2024; Schiff et al., 2025) allows each position to undergo multiple context-dependent jumps, providing genuine self-correction capability at the cost of increased trajectory complexity.

However, this multi-jump flexibility comes at a practical cost in the standard step-based sampler. At each discretization step, every position independently decides whether to jump via a Bernoulli (or categorical) draw. The total number of edits per position is therefore a Poisson-binomial random variable whose variance $\sum_k p_{ik}(1-p_{ik})$ can grow linearly with the cumulative jump mass—precisely the regime where uniform-noise models operate, since meaningful generation typically requires

multiple self-correction edits per position. This sampler-induced variance produces two characteristic failure modes at the tails of the jump-count distribution. *Under-editing* (too few jumps) occurs when insufficient substitutions leave residual noise or local inconsistencies, preventing full integration of contextual information before the process terminates. Conversely, *over-editing* (too many jumps) occurs when excessive substitutions cascade, disrupting even coherent segments and producing repetitions or distortions. These failure modes are not inherent to the uniform-noise formulation itself but are artifacts of the independent per-step sampling mechanism, and they worsen as the discretization budget (NFE) decreases.

The goal of this paper is to preserve the expressive multi-jump dynamics of uniform-noise-start discrete generative models while structurally eliminating this unnecessary sampler variance. To this end, we propose *Stratified Hazard Sampling (SHS)*, which requires no retraining, no additional hyperparameters, and no architectural modification—only a change to the inference-time sampling rule. SHS leverages the cumulative hazard of a non-homogeneous Poisson process (NHPP) and uses the cumulative hazard $S(t) = \int_0^t \lambda(s)\, ds$ to stratify event placement in this space. SHS triggers jumps when the cumulative hazard $S(t)$ crosses integer boundaries offset by a random $\theta \sim \mathrm{Uniform}(0, 1)$. Unlike prior works (Austin et al., 2021; Campbell et al., 2022; Lou et al., 2024; Schiff et al., 2025; Gat et al., 2024), which used probabilistic coin flips at fixed time steps to trigger the jumps, SHS preserves the expected jump count while bounding the jump count variance to a theoretical minimum (at most $1/4$; see Appendix C). Consequently, SHS mitigates the long-tail waiting times and "sampling luck" that are especially pronounced in uniform-noise starts, simultaneously suppressing under- and over-editing, and enabling more stable, reproducible sampling even with fewer neural function evaluations (NFEs).

Token-level blacklists provide a simple and transparent form of lexical control: at each replacement, forbidden tokens are filtered out and the remaining distribution is renormalized. While widely used, this filtering makes sampling trajectories more brittle: a single position that remains unedited can dominate lexical metrics, and this brittleness worsens as the blacklist becomes more restrictive. In this work, we keep the filtering rule fixed and isolate the effect of event scheduling, showing that SHS yields substantially improved robustness.

2 PRELIMINARIES

2.1 CTMC/DTMC-BASED DISCRETE GENERATIVE MODELS

Let $\mathcal{V}$ be a vocabulary, N the sequence length, and denote a sequence as $x = (x_1, \ldots, x_N) \in \mathcal{V}^N$. We consider a continuous-time generative process $(X_t)_{t\in[0,1]}$ with terminal distribution $p_1(x)$ and an initial distribution p_0 (often easy to sample, e.g., uniform or masked tokens).

Many non-autoregressive discrete generative models can be expressed as a time-inhomogeneous Markov process on $\mathcal{V}^N$. We focus on the common *single-site replacement* structure (Austin et al., 2021; Campbell et al., 2022; Gat et al., 2024) where, at each (continuous or discrete) time, each position either stays unchanged or is replaced by a new token.

CTMC formulation. A time-inhomogeneous continuous-time Markov chain (CTMC) is specified by a generator Q_t. For each position i we define an *escape rate* $\lambda_i(t, x) \geq 0$ and a *conditional destination distribution* $q_{t,i}(\cdot \mid x) \in \Delta(\mathcal{V})$ satisfying $q_{t,i}(x_i \mid x) = 0$ (i.e., $q_{t,i}(\cdot \mid x) \in \Delta(\mathcal{V} \setminus \{x_i\})$), where $\Delta(\mathcal{V})$ denotes the probability simplex over $\mathcal{V}$. Let $x^{(i\leftarrow v)}$ denote the sequence obtained from x by replacing x_i with v. We parameterize off-diagonal rates as

$$Q_t\Big(x^{(i\leftarrow v)} \mid x\Big) = \lambda_i(t, x)\, q_{t,i}(v \mid x), \qquad v \neq x_i, \tag{1}$$

and set $Q_t(x \mid x) = -\sum_i \lambda_i(t, x)$. This local CTMC view subsumes continuous-time discrete diffusion models (e.g., CTDD) and flow-based formulations (e.g., DFM) whenever their one-step update admits a mixture-of-(stay vs. replace) form.

DTMC formulation (discrete diffusion). A large class of discrete diffusion models is defined on a discrete time grid and specifies a time-inhomogeneous discrete-time Markov chain (DTMC) with a learned reverse kernel $P_{t_{k+1}|t_k}(\cdot \mid x)$. For single-site updates, we write the per-position kernel

as a categorical distribution $P_{k,i}(\cdot \mid x) \in \Delta(\mathcal{V})$. Any categorical kernel admits a (stay vs. replace) decomposition:

$$\begin{aligned} p_{ik}(x) &= 1 - P_{k,i}(x_i \mid x), \\ q_{k,i}(v \mid x) &= \frac{P_{k,i}(v \mid x)}{p_{ik}(x)} \quad (v \neq x_i), \end{aligned} \tag{2}$$

where $p_{ik}(x) \in [0,1]$ is the probability of changing token i at step k and $q_{k,i}$ is the destination distribution conditional on changing.

Unified jump-mass view. Both CTMC τ-leaping and DTMC categorical updates can be written in the same schematic form: at step k, draw a change indicator $B_{ik} \sim \mathrm{Bernoulli}(p_{ik})$ and, if $B_{ik} = 1$, sample a new token from $q_{k,i}(\cdot \mid x)$. For CTMC Euler/τ-leaping, $p_{ik} = h\,\lambda_i(t_k, x)$ (assuming $p_{ik} \leq 1$); for DTMC, p_{ik} is given by equation 2. Define the cumulative hazard/jump mass

$$S_{i,k} \;=\; \sum_{j=0}^{k-1} p_{ij}, \qquad S_{i,n} \approx \int_0^1 \lambda_i(t, X_t)\, dt \tag{3}$$

Under the standard step-based sampler, the total number of jumps at position i is $J_i = \sum_{k=0}^{n-1} B_{ik}$, a Poisson-binomial random variable with $\mathbb{E}[J_i] = \sum_k p_{ik}$ and $\mathrm{Var}(J_i) = \sum_k p_{ik}(1 - p_{ik})$. This Poisson-binomial variance, amplified under uniform-noise initialization where multiple self-correction edits are required, is the sampler-induced variance targeted by SHS. For more general background on CTMCs, DTMCs, and their relationship, the reader is referred to standard probability references such as Ross (2023) and Grimmett & Stirzaker (2020).

Absorbing-state (mask-start) vs. uniform-noise. The sampler-induced jump-count variance above primarily matters in *multi-jump* (self-correction) regimes. In the standard absorbing-state (mask-start) setting, the reverse dynamics is unmask-only: each position transitions monotonically from `[MASK]` to a vocabulary token at most once, so the per-position jump count satisfies $J_i \in \{0, 1\}$. See Appendix D.4 for a formal statement.

2.2 Lexical constraints via blacklist filtering

We use token-level blacklists as a simple lexical constraint and as a robustness stress test for step-based inference under *uniform-noise* initialization. Fix a blacklist ratio $\rho \in [0, 1)$ and sample a forbidden set $\mathcal{B}_\rho \subset \mathcal{V}$ uniformly at random with $|\mathcal{B}_\rho| = \lfloor \rho |\mathcal{V}| \rfloor$. Let the allowed vocabulary be $\mathcal{V}_\rho := \mathcal{V} \setminus \mathcal{B}_\rho$ and the allowed set of sequences $\mathcal{A}_\rho := \{x \in \mathcal{V}^N : \forall i,\ x_i \in \mathcal{V}_\rho\}$.

Safe-vocabulary initialization. To avoid conflating lexical filtering with explicit "token removal" dynamics, we initialize

$$X_0 \sim \mathrm{Unif}(\mathcal{V}_\rho)^N, \tag{4}$$

i.e., each position starts from a uniformly random token in the *allowed* vocabulary.

Mass-preserving destination filtering (renormalization). At inference time, we keep the model-predicted *change mass* (DTMC) $p_{ik}(x)$ in equation 2 (or the *escape rate* (CTMC) $\lambda_i(t, x)$ in equation 1) unchanged, and enforce the blacklist only by filtering and renormalizing the *destination* distribution:

$$q_{t,i}^{(\rho)}(v \mid x) \;=\; \frac{q_{t,i}(v \mid x)\, \mathbf{1}[v \in \mathcal{V}_\rho]}{\sum_{u \in \mathcal{V}_\rho} q_{t,i}(u \mid x)}. \tag{5}$$

This ensures that lexical filtering does not *trivially* reduce the expected number of effective edits by shrinking the total jump mass; it only restricts which tokens can be proposed when a jump occurs. In all blacklist experiments (Section 5.2), we apply the same filtering rule equation 5 to both the Standard sampler and SHS, and isolate the effect of *event scheduling*.

Conditioning lens (background). One may view $\mathcal{A}_\rho$ as a terminal constraint and ask for the conditional distribution $p_1(\cdot \mid \mathcal{A}_\rho)$. In general, the *exact* conditioned CTMC/DTMC dynamics corresponds to a Doob h-transform and can reweight not only destinations but also effective jump intensities, so masking-only destination filtering is generally biased as a conditional sampler. We use this only as a diagnostic lens and do not attempt to estimate h_t; see Appendix E for details.

3 Related Works

3.1 Discrete Diffusion and CTMC Generative Models

Diffusion-style generative modeling has been extended to discrete spaces in several ways. Multinomial diffusion and related categorical constructions were studied by Hoogeboom et al. (2021). D3PM (Austin et al., 2021) generalizes discrete diffusion by allowing structured transition matrices, including absorbing-state and nearest-neighbor corruptions, and demonstrates strong results on text and images. A fully continuous-time perspective that explicitly formulates both forward corruption and reverse generation as CTMCs was developed by Campbell et al. (2022), connecting discrete diffusion sampling to Markov jump process simulation techniques. More recently, Score Entropy Discrete Diffusion (SEDD) (Lou et al., 2024) learns the reverse jump rates by estimating probability ratios (concrete scores) with a scalable score-entropy objective. In parallel, Schiff et al. (2025) revisited uniform-noise diffusion language models (UDLM), deriving discrete classifier-free/classifier-based guidance and improved variational training bounds, which are particularly relevant for controllable generation settings where multiple edits per position are common. Relatedly, von Rütte et al. (2025a) propose Generalized Interpolating Discrete Diffusion (GIDD), which generalizes masked diffusion by interpolating between data and a mixing distribution and explores hybrid masking/uniform-noise noising schemes. A systematic study of scaling behavior across noise types by von Rütte et al. (2025b) finds that uniform diffusion requires more parameters but less data for compute-efficient training, and scales a uniform diffusion model to 3B and 10B parameters.

Beyond diffusion, flow-based viewpoints for discrete data have recently emerged. Discrete Flow Matching (DFM) (Gat et al., 2024) proposes a discrete analogue of flow matching for high-dimensional categorical data and reports strong performance in language and code generation. Other contemporaneous work explores geometric and manifold-aware discrete flow formulations.

3.2 Iterative Masked-Token Refinement for Non-Autoregressive Generation

A parallel line of work studies iterative refinement with masked language models. Mask-Predict (Ghazvininejad et al., 2019) performs parallel decoding by repeatedly masking and regenerating low-confidence tokens. MaskGIT (Chang et al., 2022) introduces a confidence-based masking schedule for fast non-autoregressive synthesis, popularizing keep-the-most-confident refinement. For language modeling, several diffusion-inspired or simplex/continuous relaxations have been proposed, including DiffusionBERT (He et al., 2023), SSD-LM (Han et al., 2022), and continuous-time/continuous-space categorical diffusion formulations (Dieleman et al., 2022).

3.3 Simulation of Inhomogeneous Poisson Processes and Markov Jump Processes

Simulating CTMCs and non-homogeneous Poisson processes (NHPPs) is a classical topic. Thinning is a standard exact technique for NHPP simulation (Lewis & Shedler, 1979). Uniformization is widely used in Markov jump process inference and sampling, enabling algorithms that avoid expensive matrix exponentials while preserving correctness (Rao & Teh, 2013). These classical tools inform modern samplers for high-dimensional discrete diffusion and CTMC generative models, and motivate variance-reduction schemes tailored to discrete generative inference.

3.4 Constrained generation and token-level lexical filtering

Lexical constraints have a long history as decoding-time control mechanisms in sequence generation. For autoregressive models, lexically constrained decoding has been studied through exact or approximate search procedures such as Grid Beam Search (Hokamp & Liu, 2017) and Dynamic Beam Allocation (Post & Vilar, 2018), with later work improving practicality and throughput (Hu et al., 2019). In safety-oriented settings, a widely used baseline is hard filtering (banning) of disallowed tokens or phrases. However, empirical evidence suggests that simple banned-word strategies can be insufficient as a complete safety solution, highlighting the distributional mismatch introduced by naive filtering (Gehman et al., 2020).

Our work does not propose a new constraint satisfaction algorithm. Instead, we study how common destination-side filtering heuristics interact with *step-based* CTMC/DTMC samplers. In particular,

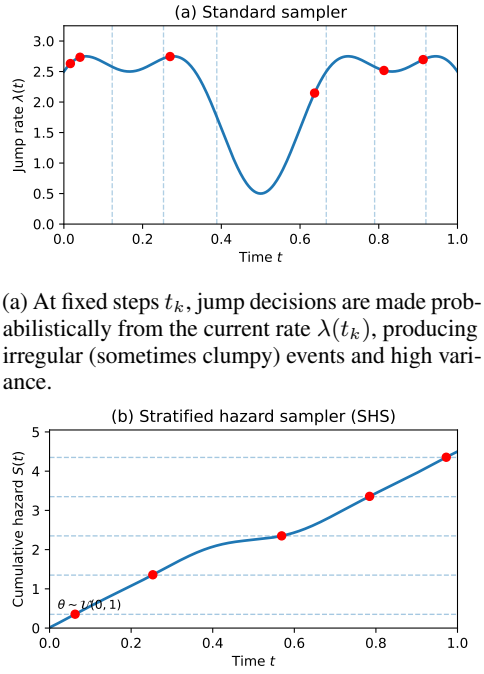
when lexical filtering is applied but the underlying stay-vs.-replace masses (or escape rates) remain unchanged, sampling quality can become sensitive to under-/over-edit trajectory tails. SHS addresses this orthogonal instability by reducing sampler-induced variance in edit counts and timing, while remaining a drop-in replacement that can be composed with destination filtering (Section 2.2 and Section 5.2).

4 Stratified Hazard Sampling

We propose *Stratified Hazard Sampling (SHS)*, adapting stratified event simulation to the *step-based* samplers used in CTMC/DTMC discrete generative models. SHS replaces the independent per-step change decisions $B_{ik} \sim \text{Bernoulli}(p_{ik})$ (Section 2) with *single-phase* stratification in cumulative hazard/jump-mass space. Concretely, SHS maintains the cumulative mass $S_{i,k}$ from equation 3 and triggers a jump whenever $S_{i,k}$ crosses unit-spaced thresholds $\{\theta_i + m\}_{m \in \mathbb{Z}_{\geq 0}}$ for a single random phase $\theta_i \sim \text{Uniform}(0, 1)$.

One boundary per step. The complete pseudocode of SHS is shown in Algorithm 1. Since $p_{ik} \leq 1$ holds by definition for DTMC kernels and is a prerequisite for valid CTMC τ-leaping, at most one unit boundary can be crossed per step, so the if statement on Line 16 is sufficient.

Note that SHS changes only the *timing* of jump events by coupling the Bernoulli decisions across steps through a single phase θ_i. Importantly, SHS leaves the *destination sampling* $q_{k,i}$ unchanged, so the model's categorical multi-modality is preserved.



(a) At fixed steps t_k, jump decisions are made probabilistically from the current rate $\lambda(t_k)$, producing irregular (sometimes clumpy) events and high variance.

(b) Cumulative hazard $S(t) = \int_0^t \lambda(s)\, ds$ and unit-spaced boundaries placed with a single offset $\theta \sim U(0, 1)$. A jump occurs deterministically when $S(t)$ crosses $\theta + k$.

Figure 1: **Standard sampler vs. SHS (ours).**

Composability with lexical filtering. SHS modifies only the stay-vs-replace event scheduling and leaves the per-jump destination sampling unchanged. Therefore, it can be composed with common destination-side modifications such as blacklist filtering by simply replacing $q_{t,i}$ with the filtered destination $q_{t,i}^{(\rho)}$ in equation 5 while keeping the same scheduler.

Jump-count concentration. Let $S_i^{\text{tot}} := S_{i,n} = \sum_{k=0}^{n-1} p_{ik}$ denote the total cumulative hazard/jump mass accumulated at position i. Write $S_i^{\text{tot}} = I_i + f_i$ with $I_i = \lfloor S_i^{\text{tot}} \rfloor$ and $f_i \in [0, 1)$. Since SHS counts how many thresholds $\theta_i + m$ are crossed by S_i^{tot}, we have

$$J_i \;=\; I_i + \mathbf{1}[\theta_i < f_i], \tag{6}$$

so $J_i \in \{I_i, I_i + 1\}$ *conditional on the realized cumulative mass* S_i^{tot}, and the long-tail jump-count fluctuations of step-wise Bernoulli sampling are eliminated. Note that in the full model the masses $p_{ik}(x)$ are state-dependent, so S_i^{tot} itself is a random variable whose distribution may differ between SHS and standard sampling (see Appendix D.3 for discussion). As a result, SHS suppresses the characteristic under-edit (too few substitutions) and over-edit (cascading substitutions) failure modes that arise under uniform-noise initialization.

4.1 Theoretical Properties

SHS is built on a simple primitive: *randomized rounding* of a nonnegative real mass S into an integer count $J \in \mathbb{Z}_{\geq 0}$ using a *single* uniform random variable. Writing $S = I + f$ with $I := \lfloor S \rfloor$ and $f := S - I \in [0, 1)$ (a decomposition we use throughout), the rounding rule is $J = I + \mathbf{1}[\theta < f]$ for $\theta \sim \text{Uniform}(0, 1)$. In SHS, this mass is the total cumulative hazard/jump mass S_i^{tot} at each position (equation 6).

Algorithm 1 Stratified Hazard Sampling (SHS) for CTMC/DTMC discrete generative models

Input:
- Initial state $X \sim p_0$
- Time grid $t_k = kh$ for $k = 0, \ldots, n$
- A routine MODELSTEP(t_k, X_{t_k}, i) that returns:
  - change mass $p_{ik} \in [0, 1]$
  - destination $q_{k,i}(\cdot \mid X_{t_k})$
  - (cf. equation 1, equation 2)

Output: Final sample X at t_n (i.e., $t = 1$)
for $i = 1$ **to** N **do**
  $S_i \leftarrow 0$; $m_i \leftarrow 0$; **Draw** $\theta_i \sim \mathrm{Uniform}(0, 1)$
end for
for $k = 0, 1, \ldots, n-1$ **do**
  for $i = 1$ **to** N **do**
    $(p_{ik},\, q_{k,i}) \leftarrow$ MODELSTEP(t_k, X_{t_k}, i)
    $S_i \leftarrow S_i + p_{ik}$
    if $S_i \geq \theta_i + m_i$ **then** {/*crossed the next hazard boundary*/}
      $X^i_{t_{k+1}} \sim q_{k,i}(\cdot \mid X_{t_k})$
      $m_i \leftarrow m_i + 1$
    else
      $X^i_{t_{k+1}} \leftarrow X^i_{t_k}$
    end if
  end for
end for
return X_{t_n}

Proposition 1 (**Unbiasedness**). *Fix $S \geq 0$ and draw $\theta \sim \mathrm{Uniform}(0, 1)$. Write $S = I + f$ with $I := \lfloor S \rfloor$ and $f := S - I \in [0, 1)$. Define $J := I + \mathbf{1}[\theta < f]$. Then $\mathbb{E}[J] = S$.*

Proposition 2 (**Minimal variance**). *With J as in Proposition 1 (i.e., $S = I + f$ with $f := S - \lfloor S \rfloor$), we have*

$$\mathrm{Var}(J) = f(1 - f) \leq \frac{1}{4}. \tag{7}$$

Moreover, among all integer-valued unbiased estimators of S supported on $\{\lfloor S \rfloor, \lceil S \rceil\}$, this variance is the minimum possible.

Proof sketch. Write $S = I + f$ with $I = \lfloor S \rfloor$ and $f = S - I \in [0, 1)$. Then $J = I + B$ where $B = \mathbf{1}[\theta < f] \sim \mathrm{Bernoulli}(f)$. Hence $\mathbb{E}[J] = I + f = S$ and $\mathrm{Var}(J) = f(1 - f) \leq 1/4$. See Appendix C for the full proof (and Appendix B for Proposition 1).

Interpretation. For a fixed mass S, standard step-wise simulation yields a Poisson-binomial count with variance $\sum_k p_k(1 - p_k)$, which can scale linearly with S in the many-edit regime. In contrast, SHS concentrates the count to two adjacent integers ($\lfloor S \rfloor$ or $\lceil S \rceil$), eliminating long-tail "sampler luck" in the number of edits while keeping the model's categorical choices intact.

A tail event relevant to blacklist robustness. Beyond variance reduction, SHS also optimally suppresses the "no-edit" tail event under a fixed realized cumulative mass: among integer-valued random variables with $\mathbb{E}[J] = S$, SHS attains the minimum possible $\mathbb{P}(J = 0)$. This provides a useful lens for lexical filtering robustness, where performance can be dominated by under-edited positions. See Appendix E.3 (Proposition 4).

4.2 On Preserving Multi-modality: Decomposing Sampler Variance

A natural concern with our minimal-variance claim is whether reducing $\mathrm{Var}(J_i)$ harms the multi-modality (generation diversity) that stochastic sampling aims to provide. We argue that SHS does not; instead, it selectively minimizes spurious variance from sampler instability while preserving meaningful variance from model expressiveness.

This can be formalized via the Law of Total Variance. Let $Q(X_1)$ denote the quality (e.g., likelihood or task metric) of a final sample X_1 at $t = 1$. The total variance in quality, $\mathrm{Var}[Q(X_1)]$, arises from two sources of randomness: the sampling trajectory $\mathcal{T}$ (controlled by SHS) and the model's categorical choices $\mathcal{U}$ at each jump (preserved by SHS). The decomposition is:

$$\mathrm{Var}[Q(X_1)] = \underbrace{\mathbb{E}_{\mathcal{T}}[\mathrm{Var}_{\mathcal{U}}(Q(X_1) \mid \mathcal{T})]}_{\text{(Term 1: Model Expressiveness)}} + \underbrace{\mathrm{Var}_{\mathcal{T}}[\mathbb{E}_{\mathcal{U}}(Q(X_1) \mid \mathcal{T})]}_{\text{(Term 2: Sampler Instability)}}.$$

Term 1 represents the "good" variance: the model's inherent multi-modality in selecting diverse, high-quality tokens given a stable trajectory $\mathcal{T}$. SHS preserves this, as it does not modify the categorical sampling (Algorithm 1).

Term 2 represents the "bad" variance: instability from sampler "luck" in $\mathcal{T}$, leading to stuck (low J_i) or overshot (high J_i) paths with poor average quality $\mathbb{E}_{\mathcal{U}}[Q]$. This is not true diversity but unreliability.

SHS contributes by minimizing Term 2: bounding $\mathrm{Var}(J_i) \leq 1/4$ conditional on S_i^{tot} ensures near-optimal paths ($J_i \approx I$ or $I + 1$), substantially reducing $\mathrm{Var}_{\mathcal{T}}[\mathbb{E}_{\mathcal{U}}(\cdot)]$. Thus, SHS enhances reliability without compromising the model's expressive power.

Crucially, while strictly separating $\mathcal{T}$ and $\mathcal{U}$ is an approximation due to state-dependent rates, this decomposition clarifies our design intent: to remove sampler-induced noise (Term 2) while leaving the model's expressive stochasticity (Term 1) intact.

5 Experiments

5.1 Text Generation

We evaluate SHS on two uniform-noise discrete diffusion language models—UDLM (Schiff et al., 2025) and GIDD (von Rütte et al., 2025a)—to test whether the variance-reduction benefit generalizes across model families. In both cases, we keep the trained model (rate/velocity predictor) *fixed* and only swap the inference procedure between the standard sampler and SHS.

Protocol. For UDLM we use $\mathrm{NFE} \in \{4, 8, 16, 32, 64\}$; for GIDD, $\mathrm{NFE} \in \{16, 32, 64, 128, 256\}$. UDLM generates 1024 sentences per run over 5 random seeds for each configuration, while GIDD generates 1024 sentences with a single seed. We report sample entropy and **generative perplexity (Gen. PPL)** scored by a fixed `GPT2-large` evaluator (lower is better). We additionally evaluate on the 3B-parameter uniform diffusion model of von Rütte et al. (2025b) with $\mathrm{NFE} \in \{4, 8, 16, 32, 64, 128\}$, 1000 sentences, and a single seed; Gen. PPL is scored by `Qwen2.5-7B`.

Results. Tables 1–3 and Figure 2 summarize the results. Across all models, SHS consistently improves Gen. PPL, with the largest gains in the *few-step* regime: on UDLM, SHS reduces Gen. PPL by 6.3% at NFE=4, narrowing to 1.4% at NFE=64; on GIDD, improvements range from 17.3% (NFE=16) to 15.7% (NFE=256); on the 3B model (von Rütte et al., 2025b), SHS reduces Gen. PPL by up to 7.4%, while the confidence-based adaptive sampler yields substantially higher Gen. PPL than either parallel sampler despite comparable position-update budgets. These gains persist across three independently trained models (110M–3B), confirming that the benefit stems from the sampler, not from model-specific artifacts.

NFE	Standard		SHS	
	Entropy	Gen. PPL↓	Entropy	Gen. PPL↓
4	6.184 ± 0.047	551.5 ± 26.9	$\mathbf{6.163 \pm 0.037}$	$\mathbf{516.8 \pm 28.3}$
8	6.282 ± 0.011	340.1 ± 18.5	$\mathbf{6.259 \pm 0.033}$	$\mathbf{322.3 \pm 11.6}$
16	6.352 ± 0.039	248.2 ± 6.3	$\mathbf{6.295 \pm 0.022}$	$\mathbf{233.9 \pm 10.6}$
32	6.375 ± 0.030	203.5 ± 12.5	$\mathbf{6.340 \pm 0.034}$	$\mathbf{195.8 \pm 11.8}$
64	6.377 ± 0.026	183.6 ± 8.0	$\mathbf{6.370 \pm 0.035}$	$\mathbf{181.0 \pm 7.3}$

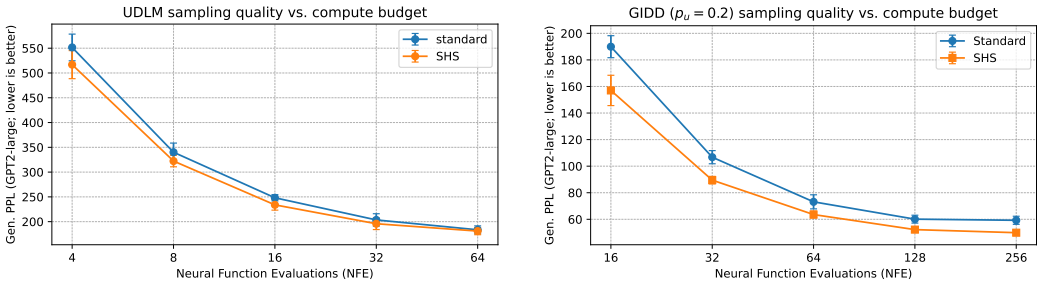
Table 1: **UDLM text generation under NFE budgets.** Mean±std over 5 seeds (1024 sentences per run). Gen. PPL is scored by `GPT2-large` (lower is better).

NFE	Standard			SHS		
	Gen. PPL↓	NLL↓	Accuracy↑	Gen. PPL↓	NLL↓	Accuracy↑
16	189.95 ± 8.02	5.246 ± 0.043	0.223 ± 0.004	$\mathbf{157.02 \pm 11.05}$	$\mathbf{5.054 \pm 0.069}$	$\mathbf{0.231 \pm 0.005}$
32	106.76 ± 4.79	4.670 ± 0.045	0.266 ± 0.004	$\mathbf{89.49 \pm 2.74}$	$\mathbf{4.494 \pm 0.031}$	$\mathbf{0.272 \pm 0.003}$
64	73.15 ± 5.13	4.290 ± 0.071	0.292 ± 0.005	$\mathbf{63.57 \pm 2.54}$	$\mathbf{4.151 \pm 0.040}$	$\mathbf{0.299 \pm 0.004}$
128	60.10 ± 2.88	4.095 ± 0.048	0.310 ± 0.005	$\mathbf{52.23 \pm 2.03}$	$\mathbf{3.955 \pm 0.039}$	$\mathbf{0.313 \pm 0.005}$
256	59.26 ± 2.95	4.081 ± 0.050	0.310 ± 0.005	$\mathbf{49.92 \pm 1.53}$	$\mathbf{3.910 \pm 0.031}$	$\mathbf{0.317 \pm 0.004}$

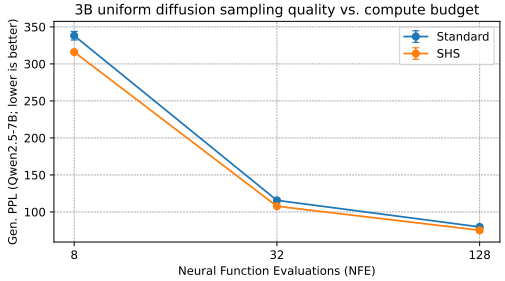
Table 2: **GIDD text generation under NFE budgets.** Mean±std over 1024 sentences. Gen. PPL is scored by `GPT2-large` (lower is better).

NFE	Standard		Adaptive (von Rütte et al., 2025b)		SHS	
	Entropy	Gen. PPL↓	Entropy	Gen. PPL↓	Entropy	Gen. PPL↓
8	5.823 ± 0.673	338.1 ± 58.5	7.164 ± 0.432	1292.0 ± 1.6	$\mathbf{5.756 \pm 0.613}$	$\mathbf{316.1 \pm 1.1}$
32	4.751 ± 0.686	115.7 ± 0.5	6.477 ± 0.813	649.8 ± 1.2	$\mathbf{4.680 \pm 0.695}$	$\mathbf{107.7 \pm 0.2}$
128	4.379 ± 0.696	79.8 ± 0.5	6.160 ± 1.249	473.3 ± 1.6	$\mathbf{4.321 \pm 0.797}$	$\mathbf{75.2 \pm 2.3}$

Table 3: **3B uniform diffusion model (von Rütte et al., 2025b): text generation under NFE budgets.** 1000 sentences, single seed. Gen. PPL is scored by `Qwen2.5-7B` (lower is better). Adaptive uses the confidence-based decoding of von Rütte et al. (2025b), updating $\lceil N_{\text{seq}}/\text{NFE} \rceil$ positions per step so that the total position-update budget is comparable to the parallel samplers.



(a) UDLM (b) GIDD



(c) 3B uniform diffusion (von Rütte et al., 2025b)

Figure 2: **Gen. PPL vs. NFE.** SHS consistently improves generation quality across models and scales, with the largest gains in the few-step regime.

5.2 Robustness under Lexical Constraints

We next evaluate robustness under *random vocabulary truncation*, which provides a controlled lexical stress test. We follow the same UDLM evaluation and GPT2-large scoring pipeline as Section 5.1, but fix the discretization budget to $\text{NFE} = 64$ and vary the blacklist severity.

Protocol. For each blacklist ratio $\rho \in \{0.0, 0.1, 0.3, 0.5, 0.7\}$, we sample a forbidden set $\mathcal{B}_\rho \subset \mathcal{V}$ uniformly at random with $|\mathcal{B}_\rho| = \lfloor \rho |\mathcal{V}| \rfloor$, and define the allowed vocabulary $\mathcal{V}_\rho = \mathcal{V} \setminus \mathcal{B}_\rho$. We initialize from the restricted uniform noise $X_0 \sim \text{Unif}(\mathcal{V}_\rho)^N$ (Eq. equation 4). At each edit, we

enforce the blacklist by renormalizing the destination distribution to $\mathcal{V}_\rho$ (Eq. equation 5), while keeping the model-predicted change masses unchanged. We compare the Standard step-based sampler and SHS under the *same* sampled blacklist $\mathcal{B}_\rho$ per run.

We generate 1024 sentences per run and repeat over 5 random seeds. As in Section 5.1, we report (i) the sample entropy logged by the UDLM evaluation and (ii) Gen. PPL computed by a fixed GPT2-large evaluator (lower is better).

Results. Random vocabulary truncation degrades generation quality as ρ increases, as expected due to a reduced feasible token set. However, SHS consistently yields lower Gen. PPL than Standard across all blacklist ratios, and the degradation with ρ is more gradual under SHS (Table 4 and Figure 3). Since each blacklist $\mathcal{B}_\rho$ is sampled independently per ρ (not nested), the meaningful comparison is the Standard–SHS gap at each fixed ρ, not the absolute trend across ρ. These robustness trends are consistent with SHS's optimal suppression of the "no-edit" tail under a fixed cumulative mass (Appendix E, Proposition 4).

ρ	Standard		SHS	
	Entropy	Gen. PPL↓	Entropy	Gen. PPL↓
0.0	6.377 ± 0.026	183.6 ± 8.0	$\mathbf{6.370 \pm 0.035}$	$\mathbf{181.0 \pm 7.3}$
0.1	6.715 ± 0.008	193.7 ± 3.1	$\mathbf{6.667 \pm 0.009}$	$\mathbf{181.8 \pm 3.7}$
0.3	6.481 ± 0.007	231.4 ± 4.3	$\mathbf{6.438 \pm 0.018}$	$\mathbf{217.1 \pm 1.7}$
0.5	6.017 ± 0.018	207.8 ± 2.8	$\mathbf{5.974 \pm 0.017}$	$\mathbf{197.5 \pm 1.3}$
0.7	4.634 ± 0.030	261.8 ± 1.6	$\mathbf{4.590 \pm 0.026}$	$\mathbf{244.2 \pm 4.4}$

Table 4: UDLM generation under random vocabulary truncation (NFE=64). Mean±std over 5 seeds (1024 sentences per run). Gen. PPL is scored by GPT2-large (lower is better).

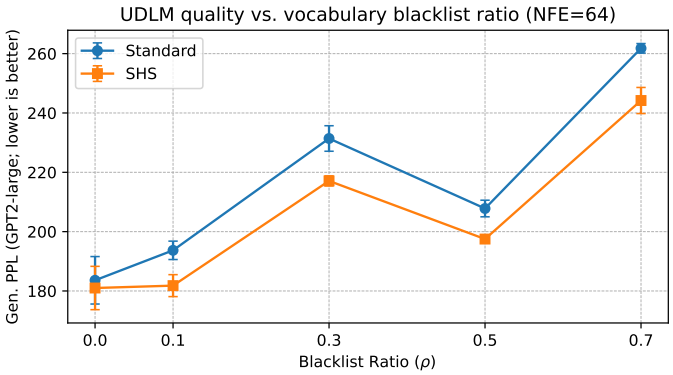


Figure 3: Robustness under random vocabulary truncation on UDLM (Gen. PPL vs. blacklist ratio ρ; lower is better). Blacklists are sampled independently per ρ; the relevant comparison is the Standard–SHS gap at each ρ.

6 Conclusion

We proposed Stratified Hazard Sampling (SHS), a drop-in, hyperparameter-free variance-reduction rule for step-based inference in CTMC/DTMC discrete diffusion and flow models. By stratifying cumulative hazard/jump mass with a single random phase per position, SHS preserves the expected number of edits while achieving minimal possible jump-count variance ($\leq 1/4$) at virtually no computational cost, without changing destination sampling.

Experiments on UDLM, GIDD, and a 3B-parameter uniform diffusion model (von Rütte et al., 2025b) show consistent sample-quality gains, especially in the few-step regime, and improved robustness under vocabulary truncation; MNIST diagnostics further confirm the intended regularization of event timing (Appendix A). Future work will extend experiments to larger code benchmarks and discrete flow matching models, and compare SHS with orthogonal sampling strategies across a broader NFE range.

ACKNOWLEDGEMENTS

We thank Yair Schiff (Cornell University) and Albert No (Yonsei University) for valuable feedback and discussions during the course of this research.

REFERENCES

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*, 2021.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, 2022.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://neurips.cc/virtual/2024/poster/95902.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3356–3369, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.301. URL https://aclanthology.org/2020.findings-emnlp.301/.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*, 2019.

Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes*. Oxford University Press, 4 edition, 2020. ISBN 9780198847601.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. *arXiv preprint arXiv:2210.17432*, 2022.

Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1535–1546, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1141. URL https://aclanthology.org/P17-1141/.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, 2021.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. Improved lexically constrained decoding for translation and monolingual rewriting. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long and Short Papers), pp. 839–850, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1090. URL https://aclanthology.org/N19-1090/.

P. A. W. Lewis and G. S. Shedler. Simulation of nonhomogeneous poisson processes by thinning. *Naval Research Logistics Quarterly*, 26(3):403–413, 1979. doi: 10.1002/nav.3800260304.

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 (ICML)*, 2024.

Matt Post and David Vilar. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1314–1324, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1119. URL https://aclanthology.org/N18-1119/.

Vinayak Rao and Yee Whye Teh. Fast MCMC sampling for markov jump processes and continuous-time bayesian networks. *Journal of Machine Learning Research*, 14:3295–3320, 2013.

Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, 13 edition, 2023.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Alexander Rush, Volodymyr Kuleshov, Hugo Dalla-Torre, Sam Boshar, Bernardo P. de Almeida, and Thomas Pierrot. Simple guidance mechanisms for discrete diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

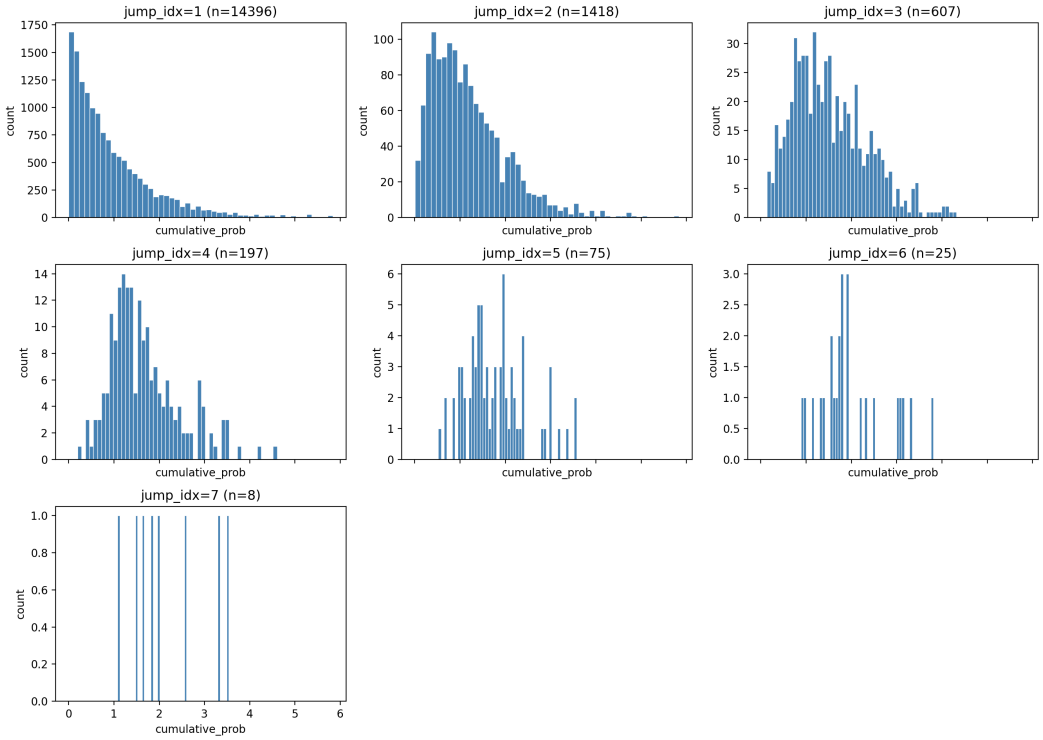
Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=rvZv7sDPV9.

Dimitri von Rütte, Janis Fluri, Omead Pooladzandi, Bernhard Schölkopf, Thomas Hofmann, and Antonio Orvieto. Scaling behavior of discrete diffusion language models, 2025b. URL https://arxiv.org/abs/2512.10858.

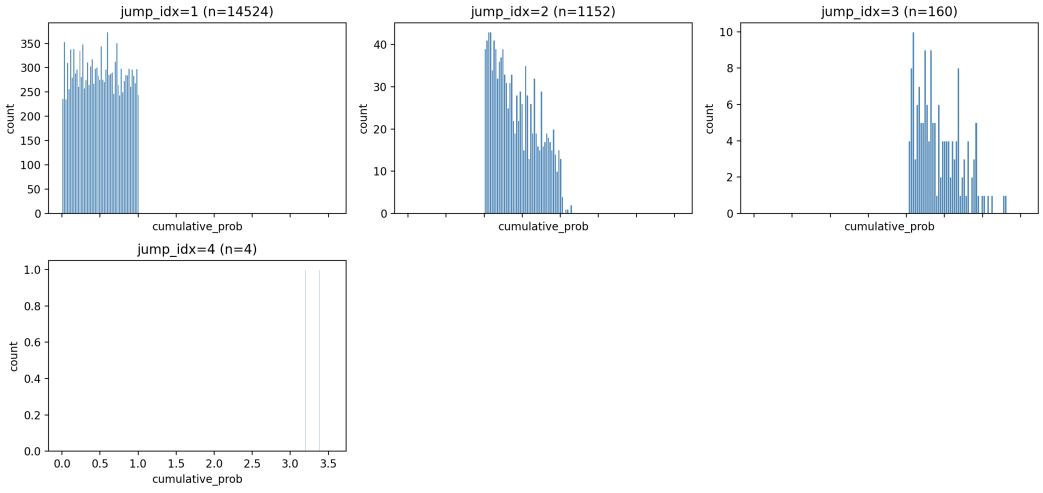
A MNIST TEST

A.1 SETUP

We use quantized MNIST (28×28, 256 states) as a controlled diagnostic: it is the simplest visual discrete generation task that admits direct inspection of under-/over-edit artifacts while allowing exhaustive trajectory logging without confounding factors from tokenizer quality or language model capacity. We fix the pretrained predictor and vary only the sampling procedure: (i) the standard sampler, and (ii) our Stratified Hazard Sampling (SHS).



(a) Standard sampling



(b) SHS

Figure 4: **Hazard-space jump locations.** Histograms of S_k (the cumulative hazard at the k-th jump; shown as `cumulative_prob` on the x-axis in the plots). Standard sampling exhibits Erlang/Gamma-shaped variability, while SHS produces stratified, bounded-support locations, demonstrating reduced trajectory randomness.

A.2 JUMP TIMING DIAGNOSTICS IN HAZARD SPACE

We first validate whether SHS produces the intended *regularization of event times* without altering the per-jump categorical choice.

Let $\lambda(t)$ denote a (token-wise) jump rate along a sampled trajectory, and define the cumulative hazard

$$S(t) = \int_0^t \lambda(s)\, ds. \tag{8}$$

We record the hazard-space jump locations $\{S_k\}_{k\geq 1}$, where S_k is the cumulative hazard value at the k-th jump.

Standard baseline: Erlang/Gamma-shaped locations. Under the standard NHPP view, inter-event increments in hazard space satisfy $\Delta S_k \sim \mathrm{Exp}(1)$ i.i.d., so the k-th jump location becomes

$$S_k = \sum_{i=1}^{k} \Delta S_i \sim \mathrm{Gamma}(k, 1), \tag{9}$$

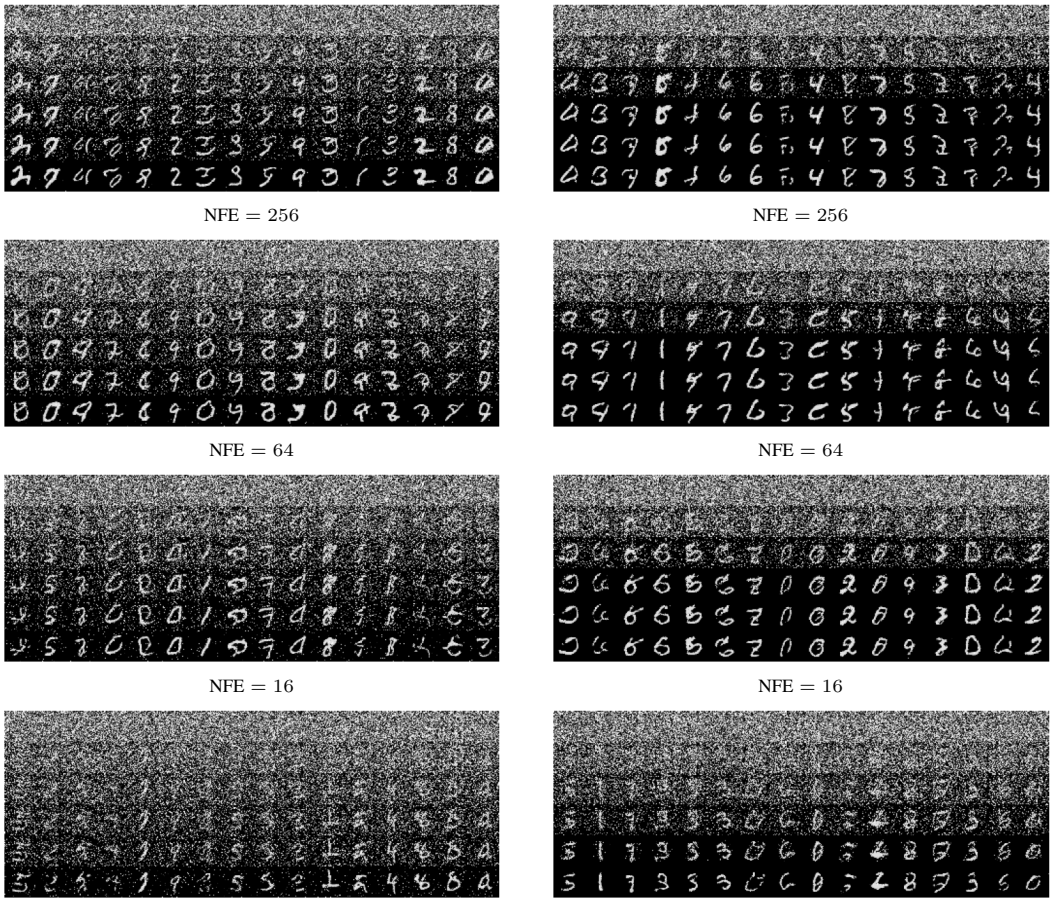
which yields an Erlang-shaped distribution as k increases. Empirically, we observe broad, heavy-tailed variability in early jump locations under standard sampling, consistent with the "under-/over-edit" instability in the noise-start setting.

SHS: stratified (bounded-support) jump locations. SHS triggers an event when the cumulative hazard crosses regularly spaced thresholds with a single random offset, which constrains the k-th event to occur within a narrow hazard interval (stratification). Consequently, the empirical histograms of H_k concentrate with bounded support and avoid the exponential long-tail behavior. This confirms that SHS makes the *timing* of jumps substantially more regular while leaving the per-jump destination sampling unchanged.

A.3 QUANTIZED MNIST RECONSTRUCTIONS: ELIMINATING UNDER-EDIT

We next visualize the practical effect of trajectory stabilization on a simple discrete generation task. Starting from a noise initialization, we run the sampler for NFE $\in \{256, 64, 16, 8\}$ discretization steps and compare the resulting reconstructions.

Observation. Standard sampling often exhibits an *under-edit* failure mode in the few-step regime: some positions experience too few jumps, leaving noticeable x_0-derived noise even late in sampling. In contrast, SHS produces visibly cleaner samples at early stages and substantially reduces residual noise in low-NFE runs. Qualitatively, SHS maintains legible digit structure even at NFE $= 16$ and NFE $= 8$, where Standard outputs remain dominated by unresolved noise.



(a) Standard sampling (b) SHS

Figure 5: **Quantized MNIST reconstructions under varying step budgets.** Standard sampling vs. SHS at NFE $\in \{256, 64, 16, 8\}$ discretization steps. SHS reduces the under-edit regime and yields cleaner early-stage reconstructions.

B Full Proof of Proposition 1 (Unbiasedness)

Fix $S \geq 0$ and draw $\theta \sim \mathrm{Uniform}(0,1)$. Let $I = \lfloor S \rfloor$ and $f = S - I \in [0,1)$. Define

$$J \;=\; I + \mathbf{1}[\theta < f]. \tag{10}$$

Then

$$\mathbb{E}[J] \;=\; I + \mathbb{P}(\theta < f) \;=\; I + f \;=\; S, \tag{11}$$

since θ is uniform on $[0,1]$. This proves Proposition 1.

C Full Proof of Proposition 2 (Minimal Variance)

Using the notation from Appendix B, we have $J = I + B$ where $B = \mathbf{1}[\theta < f] \sim \mathrm{Bernoulli}(f)$. Since I is deterministic,

$$\mathrm{Var}(J) = \mathrm{Var}(I + B) = \mathrm{Var}(B) = f(1-f) \leq \frac{1}{4}, \tag{12}$$

where the last inequality follows because the quadratic $f(1-f)$ attains its maximum at $f = 1/2$.

Finally, any integer-valued random variable $\widetilde{J}$ supported on $\{I, I+1\}$ is determined by $p = \mathbb{P}(\widetilde{J} = I+1)$. Unbiasedness $\mathbb{E}[\widetilde{J}] = S = I + f$ forces $p = f$. Therefore $\mathrm{Var}(\widetilde{J}) = p(1-p) = f(1-f)$ for *any* such unbiased estimator, and SHS attains this minimum. This proves Proposition 2.

D Additional Analysis of Stratified Hazard Sampling

This appendix collects additional properties of Stratified Hazard Sampling (SHS) and clarifies its scope for CTMC/DTMC-based samplers.

D.1 SHS as a systematic coupling of Bernoulli trials

Consider a fixed sequence of per-step change masses $\{p_k\}_{k=0}^{n-1} \subset [0,1]$ and define the cumulative sums $S_k = \sum_{j=0}^{k-1} p_j$ (with $S_0 = 0$) and $S_n = \sum_{k=0}^{n-1} p_k$. The standard step-based sampler draws independent indicators $B_k^{\mathrm{iid}} \sim \mathrm{Bernoulli}(p_k)$ and the total number of changes is $J = \sum_k B_k^{\mathrm{iid}}$ (Poisson-binomial).

SHS instead draws a *single* phase $\theta \sim \mathrm{Uniform}(0,1)$. Write $S_k = I_k + f_k$ with $I_k := \lfloor S_k \rfloor$ and $f_k := S_k - I_k \in [0,1)$, and note that $S_{k+1} = S_k + p_k$ (so $S_{k+1} = I_{k+1} + f_{k+1}$ with $I_{k+1} := \lfloor S_{k+1} \rfloor$ and $f_{k+1} := S_{k+1} - I_{k+1}$). We define a coupled indicator sequence by

$$B_k^{\mathrm{shs}} \;=\; \mathbf{1}[\, I_k + \mathbf{1}[\theta < f_k] \;<\; I_{k+1} + \mathbf{1}[\theta < f_{k+1}] \,], \tag{13}$$

which is equivalent to triggering a change whenever the cumulative sum crosses the next unit boundary $\theta + m$. Summing equation 13 over k yields the randomized-rounding form $J^{\mathrm{shs}} = \lfloor S_n \rfloor + \mathbf{1}[\theta < f_n]$ (cf. equation 6).

D.2 Event-time stratification in cumulative hazard space

For a CTMC with an *exogenous* (state-independent) rate $\lambda(t) \geq 0$, define the continuous cumulative hazard $S(t) = \int_0^t \lambda(\tau)\, d\tau$. SHS places event boundaries at $\theta,\ \theta+1,\ \theta+2, \ldots$ in hazard space, so the m-th event time is

$$T_m \;=\; \inf\{t \in [0,1] : S(t) \geq \theta + (m-1)\}. \tag{14}$$

Equivalently, the hazard value at event m is exactly $\theta + (m-1)$. Thus, SHS guarantees *exactly one* event in each unit-length hazard interval, eliminating the exponential-tail inter-arrival variability of Poisson processes in hazard space (where $\Delta S \sim \mathrm{Exp}(1)$).

On a discrete grid (DTMC kernels or CTMC τ-leaping), S_k increases in steps. If each increment satisfies $p_k \leq p_{\max} \leq 1$, then the boundary-crossing event triggered at step k overshoots its hazard boundary by at most $p_{\max}$. Therefore, SHS still yields tightly controlled event locations in hazard/jump-mass space even under discretization.

D.3 SCOPE: INFERENCE-TIME VARIANCE REDUCTION VS. EXACT SIMULATION

The results in Propositions 1–2 are unconditional statements about randomized rounding for a *fixed* mass S (or a fixed sequence $\{p_k\}$). In a full generative model, the per-step masses $p_{ik}(x)$ (or rates $\lambda_i(t, x)$) are typically *state-dependent*. Because SHS couples change decisions over time through the phase θ_i, it introduces additional memory beyond the visible state x. Accordingly, SHS should be viewed as an *inference-time variance reduction integrator* for CTMC/DTMC samplers rather than as an exact simulator of the underlying Markov process.

Despite this, SHS leaves the per-change destination kernels $q_{k,i}$ untouched and dramatically reduces sampler-induced variability in the number and timing of edits. This reduction is especially valuable under uniform-noise initialization, where meaningful generation requires multiple self-correction edits per position and where Poisson-binomial fluctuations can dominate the observed instability.

Composability with destination-side modifications. SHS changes only the *event scheduling* (stay vs. replace decisions) by coupling per-step Bernoulli trials through a single phase, and leaves the per-jump destination sampling unchanged (Algorithm 1). Therefore, SHS can be composed with destination-side modifications such as vocabulary truncation / blacklist filtering by simply replacing $q_{t,i}$ (or $q_{k,i}$) with the filtered destination $q^{(\rho)}$ in equation 5, while keeping the same change masses p_{ik} (or escape rates λ_i).

D.4 ABSORBING-STATE (MASK-START) IMPLIES SINGLE-JUMP TRAJECTORIES

This appendix formalizes why the jump-count variance discussed in Section 2.1 is a *multi-edit* phenomenon and is absent under the standard absorbing-state (mask-start) setting.

Setup. Let V be the vocabulary and let $m =$ `[MASK]` be a dedicated sentinel token. Define the extended alphabet $\bar{V} := V \cup \{m\}$, and consider step-based sampling on a grid $\{t_k\}_{k=0}^{n}$. A state is $x \in \bar{V}^N$. At each step k and position i, assume the sampler admits a (stay vs. replace) decomposition as in Section 2.1, with a change mass $p_{ik}(x) \in [0, 1]$ and a destination distribution $q_{k,i}(\cdot \mid x) \in \Delta(V)$.

Unmask-only (absorbing) assumption. In the absorbing-state (mask-start) sampler, we assume *unmask-only* dynamics: once a position leaves m, it never changes again. Concretely, for any $x \in \bar{V}^N$,

$$p_{ik}(x) = 0 \quad \text{whenever } x_i \in V. \tag{15}$$

When $x_i = m$, the update at step k draws $B_{ik} \sim \text{Bernoulli}(p_{ik}(x))$ and, if $B_{ik} = 1$, sets $x'_i \sim q_{k,i}(\cdot \mid x)$ (thus $x'_i \in V$); otherwise it keeps $x'_i = m$.

Define the (non-trivial) edit indicator and total edit count

$$B_{ik} := 1[x'_i \neq x_i], \qquad J_i := \sum_{k=0}^{n-1} B_{ik}. \tag{16}$$

Proposition 3 (Single-edit property under absorbing-state dynamics)**.** *Under equation 15, each position can undergo at most one edit: for every i, $J_i \in \{0, 1\}$ almost surely.*

Proof. If $J_i = 0$ there is nothing to prove. Otherwise, let $k^\star$ be the first step where $B_{ik^\star} = 1$. By construction, x_i changes only when $x_i = m$, hence $x_i^{(k^\star)} = m$ and after the update we have $x_i^{(k^\star+1)} \in V$. Then equation 15 implies that for all later steps $\ell > k^\star$, $p_{i\ell}(x^{(\ell)}) = 0$ and thus $B_{i\ell} = 0$. Therefore exactly one edit can occur, so $J_i = 1$. □

Variance implication. Since $J_i \in \{0, 1\}$, it is a Bernoulli random variable and

$$\text{Var}(J_i) \leq \tfrac{1}{4}. \tag{17}$$

Moreover, many mask-start implementations explicitly force any remaining `[MASK]` tokens to be resolved by the terminal time (e.g., by setting $p_{i,n-1}(x) \equiv 1$ for $x_i = m$ at the final step), in which case $J_i = 1$ deterministically and $\text{Var}(J_i) = 0$.

Interpretation (why SHS is mainly needed for uniform-noise start). Under mask-start, the state explicitly reveals whether a position has been generated via the sentinel $m = \texttt{[MASK]}$, and the unmask-only constraint equation 15 enforces a single-edit trajectory per position. In contrast, uniform-noise start typically requires multiple self-correction edits per position, where sampler randomness in the number and timing of edits becomes substantial (Section 2.1), motivating SHS as a minimal-variance event scheduler.

E BLACKLISTS: CONDITIONING LENS AND MASS-PRESERVING FILTERING

This appendix collects background and technical details for the random-blacklist experiments in Section 5.2. We emphasize that our main paper studies SHS as an *inference-time variance reduction rule* (Appendix D, Section D.3), and we use blacklists primarily as a robustness stress test rather than as an exact conditional sampler.

E.1 CONDITIONING VIEWPOINT AND DOOB h-TRANSFORM (BACKGROUND)

Let $\mathcal{B} \subset \mathcal{V}$ be a blacklist and define the allowed event $\mathcal{A} = \{x \in \mathcal{V}^N : \forall i,\ x_i \notin \mathcal{B}\}$. If the goal is the *conditional* terminal distribution,

$$p_1(x \mid \mathcal{A}) \;=\; \frac{p_1(x)\,\mathbf{1}[x \in \mathcal{A}]}{\mathbb{P}_{p_1}(X_1 \in \mathcal{A})}, \tag{18}$$

then conditioning generally changes the dynamics at intermediate times.

CTMC Doob transform. For a time-inhomogeneous CTMC with generator Q_t, define the harmonic function

$$h_t(x) \;=\; \mathbb{P}(X_1 \in \mathcal{A} \mid X_t = x). \tag{19}$$

A classical result (Doob's h-transform) gives the conditioned generator

$$Q_t^{(\mathcal{A})}(x, y) \;=\; Q_t(x, y)\,\frac{h_t(y)}{h_t(x)} \qquad (y \neq x), \tag{20}$$

with diagonal entries chosen so rows sum to zero. Crucially, equation 20 reweights not only *which* transitions are taken but also their effective intensities, through the lookahead ratio $h_t(y)/h_t(x)$.

DTMC analogue. An analogous h-transform exists for discrete-time Markov chains and similarly reweights the transition kernel by a lookahead ratio. The key point is the same: the correct conditional dynamics depends on future survival.

Why destination-only masking is not exact conditioning. A common blacklist heuristic modifies only the destination distribution via renormalization (cf. equation 5):

$$q_{t,i}^{(\rho)}(v \mid x) \;\propto\; q_{t,i}(v \mid x)\,\mathbf{1}[v \notin \mathcal{B}_\rho], \tag{21}$$

while leaving the base escape rates $\lambda_i(t, x)$ (or DTMC stay-vs.-replace masses $p_{ik}(x)$) unchanged. This generally defines a different dynamics than the Doob-transformed one equation 20, since the latter incorporates the global lookahead h_t in both destinations and intensities. In this paper, we do *not* attempt to estimate h_t; we use this viewpoint only as a diagnostic lens.

E.2 MASS-PRESERVING DESTINATION FILTERING USED IN OUR EXPERIMENTS

In Section 5.2, we use *random vocabulary truncation* rather than explicit token-removal conditioning. Fix ρ and let $\mathcal{V}_\rho = \mathcal{V} \setminus \mathcal{B}_\rho$. We start from $X_0 \sim \mathrm{Unif}(\mathcal{V}_\rho)^N$ (Eq. equation 4) and enforce the blacklist by *renormalizing only the destination distribution* (Eq. equation 5), keeping the change mass unchanged.

DTMC kernel view. Let $P_{k,i}(\cdot \mid x)$ be the model's per-position categorical kernel, and let $p_{ik}(x) = 1 - P_{k,i}(x_i \mid x)$ and $q_{k,i}(\cdot \mid x)$ be the stay-vs.-replace decomposition (Eq. equation 2). Define the filtered destination

$$q_{k,i}^{(\rho)}(v \mid x) \;=\; \frac{q_{k,i}(v \mid x)\,\mathbf{1}[v \in \mathcal{V}_\rho]}{\sum_{u \in \mathcal{V}_\rho} q_{k,i}(u \mid x)}. \tag{22}$$

Then the filtered kernel is

$$P_{k,i}^{(\rho)}(x_i \mid x) = P_{k,i}(x_i \mid x), \qquad P_{k,i}^{(\rho)}(v \mid x) = p_{ik}(x)\, q_{k,i}^{(\rho)}(v \mid x) \quad (v \neq x_i), \tag{23}$$

so the total probability of changing the token remains exactly $p_{ik}(x)$.

CTMC generator view. Similarly, for a CTMC parameterization $Q_t(x^{(i \leftarrow v)} \mid x) = \lambda_i(t, x) q_{t,i}(v \mid x)$, we keep $\lambda_i(t, x)$ unchanged and replace $q_{t,i}$ by $q_{t,i}^{(\rho)}$ in Eq. equation 5, redistributing rate mass only over the allowed vocabulary.

Relation to "compensating" for removed mass. An equivalent implementation of sampling $v \sim q_{t,i}^{(\rho)}(\cdot \mid x)$ is rejection sampling: draw $v \sim q_{t,i}(\cdot \mid x)$ repeatedly until $v \in \mathcal{V}_\rho$. This can be interpreted as "compensating" for the removed blacklist mass by renormalization; it preserves the effective change probability once a change event is triggered.

E.3 A SIMPLE SHS PROPERTY RELEVANT TO BLACKLIST ROBUSTNESS

The blacklist experiments typically become more brittle as ρ increases because generation must succeed under a smaller feasible action space. In this regime, *under-editing tails* (positions that receive too few edits) are especially harmful. The following elementary proposition formalizes an optimal suppression of the zero-edit event under a fixed realized cumulative mass.

Proposition 4 (SHS minimizes the probability of zero edits under fixed cumulative mass)**.** *Fix $S \geq 0$ and let $J \in \mathbb{Z}_{\geq 0}$ be any random variable such that $\mathbb{E}[J] = S$. Then*

$$\mathbb{P}(J = 0) \;\geq\; (1 - S)_+ \;:=\; \max\{0, 1 - S\}. \tag{24}$$

Moreover, SHS attains equality: if $\theta \sim \mathrm{Unif}(0, 1)$ and $J_{\mathrm{shs}} = \lfloor S \rfloor + \mathbf{1}[\theta < S - \lfloor S \rfloor]$, then $\mathbb{P}(J_{\mathrm{shs}} = 0) = (1 - S)_+$.

Proof. Since $J \geq \mathbf{1}[J \geq 1]$, we have $\mathbb{E}[J] \geq \mathbb{P}(J \geq 1)$. Thus $\mathbb{P}(J = 0) = 1 - \mathbb{P}(J \geq 1) \geq 1 - \mathbb{E}[J] = 1 - S$. Truncating at 0 yields $(1 - S)_+$. For SHS, if $S \geq 1$ then $J_{\mathrm{shs}} \geq 1$ almost surely. If $S < 1$ then $J_{\mathrm{shs}} = \mathbf{1}[\theta < S]$ is Bernoulli(S), hence $\mathbb{P}(J_{\mathrm{shs}} = 0) = 1 - S$. □

How to read Proposition 4 in this paper. Proposition 4 is a statement about randomized rounding for a *fixed* realized mass S. In the full model, the per-step masses $p_{ik}(x)$ are state-dependent, so SHS should be viewed as an inference-time variance reduction rule rather than an exact simulator (Appendix D, Section D.3). Nevertheless, the proposition provides a useful lens: among unbiased integer edit-count constructions with the same expected total mass, SHS is optimal in suppressing the "no-edit" tail, which is consistent with the robustness trends observed under increasing blacklist severity.