On the Error-Correcting Effects of Stochasticity in Discrete Diffusion

William Yuan
Georgia Institute of Technology
wyuan65@gatech.edu

Sungwon Jeong
Georgia Institute of Technology
sjeong304@gatech.edu

Amirali Aghazadeh
Georgia Institute of Technology
aaghazadeh3@gatech.edu

Abstract

Discrete diffusion models achieve strong performance in text and image generation, but their inference remains slow and must inherently balance sampling efficiency and sample quality. In this work, we present a systematic study of how the *degree of stochasticity* in Markov transitions governs the sampling tradeoff. We show that highly deterministic transitions converge rapidly but suffer from error accumulation, while more stochastic transitions converge more slowly yet can achieve higher final sample quality. Using an information-theoretic analysis, we identify the underlying mechanism as an error-correcting effect induced by *redundant transitions* that symmetrically exchange mass between states, and show that these transitions can provably contract sampling errors. To demonstrate the practical implications of our theoretical findings, we propose *Discrete Churn and Restart Sampling* (DCRS), a sampling procedure that injects controlled stochasticity by alternating between forward and reverse diffusion processes. Experiments on synthetic datasets and large-scale benchmarks show that DCRS improves the speed-quality tradeoff in the low number of function evaluations regime. On image datasets, DCRS achieves up to a $10\times$ reduction in sampling steps compared to standard samplers while maintaining competitive sample quality, whereas on language benchmarks, we observe more nuanced behavior depending on the corruption process and sampling procedure.

1 Introduction

Discrete diffusion models, a class of generative models that iteratively refine samples via discrete-space Markov chains [1–3], have achieved competitive performance in natural language [4, 5] and image [6, 7] generation. A central challenge in discrete diffusion is understanding the tradeoff between sample quality and sampling speed. Despite its practical importance, this tradeoff is poorly studied and is widely believed to depend on the degree of stochasticity of the generation process.

Existing intuitions are largely heuristic. Masking-based transitions are irreversible during sampling and are therefore thought to be susceptible to error accumulation [8, 9], while highly stochastic uniform transitions are thought to exhibit self-correcting behavior [10, 11]. However, comparisons across transition types are inherently confounded by differences in learned denoising networks. Even under a fixed denoising model, prior work has treated stochasticity as a hyperparameter for improving sample quality, yet its underlying role in error propagation remains poorly understood [12–16].

In this work, we show that stochasticity plays a fundamental and previously underexplored role in discrete diffusion inference. Specifically, we demonstrate a tradeoff: highly deterministic transitions

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

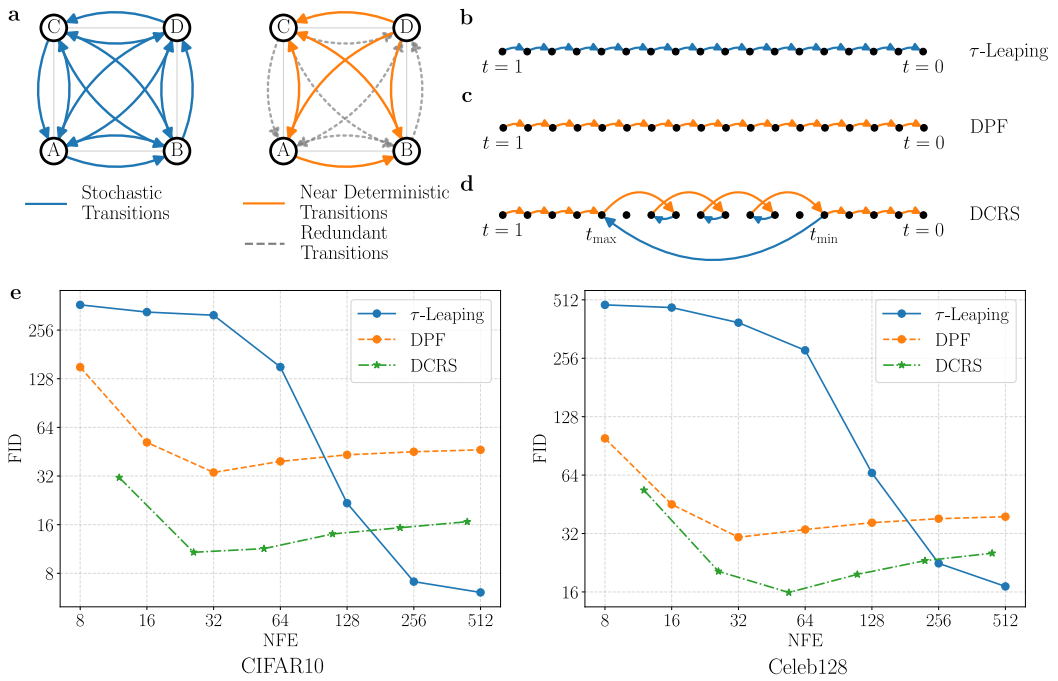


Figure 1: **Stochasticity induces error correction in discrete diffusion, enabling an improved speed-quality tradeoff. (a)** Near-deterministic transitions (orange) obtained by removing redundant transitions from the stochastic kernel (blue), resulting in faster but less error-corrective dynamics. **(b)** Sampling with fully stochastic transitions, where redundant transitions redistribute probability mass between states and help correct accumulated errors. **(c)** Discrete Probability Flow (DPF), which removes redundant transitions to produce nearly deterministic updates that accelerate convergence but are more sensitive to model errors. **(d)** Our proposed Discrete Churn and Restart Sampling (DCRS), which alternates between near-deterministic reverse updates and injected forward-process noise. Restarting introduces stochasticity using the forward process without additional neural network evaluations, while churning provides local corrections. **(e)** Empirical results on CIFAR10 and CelebA illustrating the speed-quality tradeoff: DCRS achieves improved sample quality in the low number of function evaluations (NFE) regime.

enable faster convergence but achieve lower sample quality, whereas more stochastic transitions converge more slowly to higher sample quality. Crucially, we show that this behavior arises from an *error-correcting mechanism* induced by *redundant transitions* between state pairs (Figure 1a), which symmetrically exchange probability mass without altering the time marginal distributions. Using an information-theoretic analysis, we show that these transitions can provably contract errors in the KL divergence. This perspective provides a principled explanation for when and why stochasticity improves generation quality, beyond prior heuristic arguments.

These results unify a broader principle. Stochasticity improves generation by mixing away errors, but at the cost of slower sampling. While similar tradeoffs have been observed in continuous diffusion models [12, 17–20], we show that in the discrete setting, the strength of error correction depends critically on the *type and degree of stochasticity* feasible under the chosen corruption process.

Motivated by this understanding, we propose *Discrete Churn and Restart Sampling* (DCRS), a training-free inference procedure that injects stochasticity in a controlled manner. DCRS alternates between near-deterministic reverse steps and strategically introduced forward-process noise through two mechanisms: (1) *churning*, which performs local forward-reverse perturbations, and (2) *restarting*, which periodically resets the sampling trajectory using forward transitions (Figure 1d). Notably, the restart steps introduce stochasticity via the forward process *without additional neural network evaluations*, enabling efficient global error correction across the trajectory.

We evaluate DCRS on discrete diffusion models trained on large-scale image and language datasets. We find that DCRS substantially improves the speed-quality tradeoff in the low number of function evaluations (NFE) regime on image benchmarks, achieving comparable sample quality with significantly fewer sampling steps (Figure 1e). On language tasks, we observe more nuanced behavior,

indicating that the strength of error-correcting stochasticity may depend on the domain and biases in the sampling procedure.

Our contributions in this work are as follows:

- We provide a systematic characterization of how stochasticity governs the sampling speed-quality tradeoff in discrete diffusion models. Furthermore, we show that stochasticity can correct for *model approximation errors* at the cost of slower sampling, while reducing stochasticity accelerates sampling but weakens error correction.
- We identify an error-correcting mechanism induced by redundant transitions and show, via an information-theoretic analysis, that adding such transitions can provably contract sampling errors.
- We develop DCRS, a sampling procedure that exploits the error-correcting effects of stochasticity by alternating the forward and reverse processes, improving the speed-quality tradeoff in the low-NFE regime.

2 Background

Continuous-Time Discrete Diffusion Models. Discrete diffusion models operate on spaces $\mathcal{S}^D$, where $\mathcal{S} = \{1, \dots, S\}$ is a finite state space and D denotes the sequence dimension. These models define a forward Markov process over time $t \in [0, 1]$ that evolves a data distribution p_0 toward a limiting distribution p_1. The time marginals $\{p_t\}_{t\in[0,1]}$ are governed by a time-inhomogeneous continuous-time Markov chain (CTMC) with rate matrix $R_t \in \mathbb{R}^{S^D \times S^D}$ satisfying $\frac{dp_t}{dt} = R_t^T p_t$. A generative model is obtained by simulating the corresponding reverse-time process starting from p_1.

The reverse rate matrix $\hat{R}_t$ must satisfy the time-reversal relation, $\hat{R}_t(x, y) = \frac{p_t(y)}{p_t(x)} R_t(y, x)$, which ensures the reversed process follows the time marginal distributions p_t [21]. The probability ratio $\frac{p_t(y)}{p_t(x)}$ is commonly referred to as the discrete score function and can be learned either directly by estimating $s_t^\theta(x)_y \approx \frac{p_t(y)}{p_t(x)}$ [4, 22] or indirectly by learning the posterior distribution $p_{0|t}^\theta \approx q_{0|t}$ and then recovering the discrete score via $\frac{p_t(y)}{p_t(x)} = \sum_{x_0} \frac{q_{t|0}(y|x_0)}{q_{t|0}(x|x_0)} q_{0|t}(x_0 \mid x)$ [2, 3]. Sampling from the CTMC reversal can be performed by discretizing time, the details of which we leave to Appendix A. Since both the estimate of the discrete score function and the method of sampling are inexact, sampling accuracy depends on two distinct sources of error. Understanding the role of each source of error is the focus of our analysis.

Choice of corruption process. Discrete diffusion models admit flexibility in the choice of R_t; however, for computational tractability over large state spaces, rate matrices with low-rank structure are typically used. Specifically, we focus on rate matrices defined for single coordinates $R_t = \beta_t(\mathbf{1}\boldsymbol{\pi}^T - I)$, where $\boldsymbol{\pi} \in \mathbb{R}^S$ denotes the stationary distribution of the corruption process and β_t denotes the noise schedule. Two commonly used stationary distributions we study in this work are the uniform and masking distributions, $\boldsymbol{\pi} = \frac{1}{S}\mathbf{1}$ and $\boldsymbol{\pi} = \boldsymbol{e}_M$.

Adjusting stochasticity at inference time. Although fully deterministic sampling is generally infeasible in discrete settings, the reverse process can be made nearly deterministic [14–16]. The Discrete Probability Flow (DPF) formulation removes *redundant transitions* between state pairs:

$$\hat{R}_{t,\mathrm{DPF}}(y, x) = \left(\frac{p_t(x)}{p_t(y)} R_t(x, y) - R_t(y, x)\right)_+, \tag{1}$$

where $(\cdot)_+ = \max\{\cdot, 0\}$. Importantly, the learned discrete score function $\frac{p_t(y)}{p_t(x)}$ can be reused without any retraining. This construction eliminates symmetric transitions that exchange probability mass between states, resulting in faster but more deterministic dynamics. Although these redundant transitions do not affect the marginal evolution of p_t, we show that they play a critical role in correcting sampling errors. More generally, stochasticity can be reintroduced by adding any valid rate matrix $\hat{R}_{t,\mathrm{DB}}$ that satisfies the detailed-balance relation $p_t(x)\hat{R}_{t,\mathrm{DB}}(x, y) = p_t(y)\hat{R}_{t,\mathrm{DB}}(y, x)$:

$$\hat{R}_{t,\nu_t} = \hat{R}_{t,\mathrm{DPF}} + \nu_t \hat{R}_{t,\mathrm{DB}}, \tag{2}$$

where $\nu_t \geq 0$ controls the level of additional stochasticity [15].

Under an exact discrete score, all such choices of ν_t produce the same marginal distributions. In practice, model approximation errors break this equivalence and hence ν_t must be tuned to balance sample quality and sampling efficiency.

3 Explaining Error Correction

A central question in discrete diffusion is: *why does stochasticity improve sample quality, despite slowing down sampling?* Empirically, we observe a consistent tradeoff: less stochastic samplers converge faster but are sensitive to accumulated errors, while more stochastic samplers converge more slowly yet often achieve lower final error. This suggests that stochasticity plays a dual role, simultaneously affecting both convergence speed and error propagation.

We show that this behavior is a consequence of how stochasticity interacts with different forms of sampling error across NFE regimes: (i) **Discretization errors (low-NFE)**: Increasing stochasticity increases the number of transitions per unit time, leading to larger discretization errors when using finite step sizes. (ii) **Model approximation errors (high-NFE)**: Stochastic transitions mix probability mass across states, contracting distributional errors and mitigating sampling errors. This section formalizes this tradeoff and characterizes the error-correcting role of stochasticity.

3.1 Information-Theoretic Analysis

We first address how stochasticity affects sampling in the absence of model approximation error. In the low-NFE regime, increasing stochasticity leads to larger discretization error: Euler steps using rate R are known to incur $O(\Delta t^2 \|R\|^2)$ error locally. Intuitively, higher transition rates induce more frequent state changes, making finite-step approximations less accurate. This reasoning is consistent with prior work [3, 23], which provides global convergence guarantees in distribution with discretization errors that scale with the largest entries of $\hat{R}_t$.

In the high-NFE regime, discretization errors are negligible, so improvements in sample quality due to stochasticity must arise from differences in how *model approximation errors* accumulate. This suggests that stochasticity must play some role in enabling robustness to general errors.

To analyze error propagation, we consider q_s, a perturbed distribution relative to the true marginal p_s. A common intuition is that a noisy channel $q_{t|s}$ can only destroy distinguishability between inputs p_s and q_s. This is captured through the data processing inequality, $D_{KL}(q_{t|s}q_s \| q_{t|s}p_s) \leq D_{KL}(q_s \| p_s)$, which implies that errors *cannot become worse*. More importantly, under specific inputs and channels, the divergence can *strictly contract* [24, 25]. This motivates studying the tightest possible contraction coefficient

$$\eta_{KL}(p_s, q_{t|s}) = \sup_{q_s : D_{KL}(q_s \| p_s) < \infty} \frac{D_{KL}(q_{t|s}q_s \| q_{t|s}p_s)}{D_{KL}(q_s \| p_s)},$$

which quantifies the worst-case error propagation through the transition kernel, where $\eta_{KL}(q_{t|s}) = \sup_{p_s} \eta_{KL}(p_s, q_{t|s})$ takes the worst case over all input distributions. A natural question under this theoretical formulation is whether some corruption processes are more contractive. We show that this is indeed the case in Lemma 3.1:

Lemma 3.1 (Forward Process Contraction)**.** *Let p_s be the ground-truth time marginal distribution and q_s be a perturbed distribution. Then*

$$D_{KL}(q_{t|s}q_s \| q_{t|s}p_s) \leq \eta_{KL}(p_s, q_{t|s}) D_{KL}(q_s \| p_s),$$

where for masking diffusion we have $\eta_{KL}(p_s, q^M_{t|s}) = \frac{\alpha_t}{\alpha_s}$, whereas for uniform diffusion we have $\eta_{KL}(p_s, q^U_{t|s}) \leq \frac{S(\frac{\alpha_t}{\alpha_s})^2}{(S-2)\frac{\alpha_t}{\alpha_s}+2}$. In particular, for forward processes of the form $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + (1-\alpha_t)\boldsymbol{\pi}$, masking achieves the worst-case contraction rate for all forward processes we consider.

The lemma shows that the error-contracting effects of stochasticity depend on the choice of transition kernel. Especially when the size of the state space S is small, uniform diffusion exhibits stronger contraction than masking diffusion, where $\eta_{KL}(p_s, q^U_{t|s}) < \eta_{KL}(p_s, q^M_{t|s})$. We further demonstrate that masking diffusion achieves the worst-case contraction rate for all inputs and the forward processes considered here (see Appendix B for proof).

While the forward process is contractive toward a stationary distribution by design, it is less obvious whether similar guarantees hold for the reverse process used in sampling. Surprisingly, we show that reverse-time transitions also exhibit error contraction in Lemma 3.2, which states:

Lemma 3.2 (Reverse Process Contraction)**.** *Consider a diffusion process with noise schedule α_t and define $\sigma_{s|t} := \frac{1-\alpha_s}{1-\alpha_t}$. Let $q^{\mathrm{DPF}}_{s|t}$ denote the reverse transition kernel corresponding to the DDIM/DPF update in Equation (13). Then DPF sampling satisfies $\eta_{KL}(p_t, q^{\mathrm{DPF}}_{s|t}) \leq \sigma_{s|t} + (1-\sigma_{s|t})\eta_{KL}(p_t, p_{0|t})$. For masking diffusion, $\eta_{KL}(p_t, q_{s|t}) = 1$ and the bound is tight, whereas for uniform diffusion $\eta_{KL}(p_t, q_{s|t}) < 1$. Now consider a stochasticity schedule ν_t with corresponding reverse kernel $q^{\nu_t}_{s|t}$. This kernel can be expressed as a composition of $q^b_{s|t}$, the DPF step followed by the backward corrector step, and $q^f_{t|s}$, the forward step. Consequently, $\eta_{KL}(p_t, q^{\nu_t}_{s|t}) \leq \eta_{KL}(p_t, q^b_{s|t})\eta_{KL}(q^f_{t|s})$.*

The result is twofold. First, it demonstrates that near the data distribution where the posterior is nearly deterministic, error contraction is significantly worse. Second, it shows that increasing stochasticity can reduce the contraction coefficient, correcting for perturbations from previous iterations. However, as we show next through an analysis in TV distance, this comes at the cost of amplifying model approximation errors in later iterations.

Theorem 3.3 (Contraction Under Model Error)**.** *Consider reverse-time sampling with a learned posterior estimator $p^\theta_{0|t}$ and discretization $1 = t_1 > t_2 > \cdots > t_{K+1} = 0$. Let p^θ_t denote the generated marginal distribution and p_t the ground-truth marginal distribution. Define the posterior estimation error at step k as $\epsilon_k := TV\left(p^\theta_{0|t_k}, p_{0|t_k} \middle| p^\theta_{t_k}\right)$. Then the sampling error satisfies the recursion $TV(p^\theta_{t_{k+1}}, p_{t_{k+1}}) \leq A_k TV(p^\theta_{t_k}, p_{t_k}) + B_k\epsilon_k$, where A_k controls contraction of accumulated sampling error, B_k controls amplification of posterior estimation error. Consequently,*

$$TV(p^\theta_0, p_0) \leq \sum_{k=1}^{K} \left(\prod_{j=k+1}^{K} A_j \right) B_k \epsilon_k.$$

For DPF sampling and sampling with stochasticity schedule $\nu_{t_k} = \frac{\alpha_{t_k}(1-\alpha_{t_{k+1}})}{\alpha_{t_{k+1}} - \alpha_{t_k}}$, these coefficients are

$$A^{\mathrm{DPF}}_k = \sigma_{t_{k+1}|t_k} + \left(1 - \sigma_{t_{k+1}|t_k}\right)\eta_{TV}(p_{0|t_k}), \qquad B^{\mathrm{DPF}}_k = 1 - \sigma_{t_{k+1}|t_k},$$
$$A^{\nu_t}_k = \alpha_{t_{k+1}}\eta_{TV}(p_{0|t_k}), \qquad B^{\nu_t}_k = \alpha_{t_{k+1}}.$$

In particular, if $\eta_{TV}(p_{0|t_k}) = 1$, we have $A^{\nu_t}_k \leq A^{\mathrm{DPF}}_k$ and $B^{\nu_t}_k \geq B^{\mathrm{DPF}}_k$.

The bound reveals two competing terms: A_k, which contracts accumulated errors and decreases with increased stochasticity, and B_k, which amplifies model approximation error and grows with stochasticity. This formalizes the tradeoff observed in practice: stochasticity improves sample quality up to an optimal level, beyond which model error dominates. Finally, although our analysis focuses on the one-dimensional setting, Appendix B extends the error bounds to general dimensions, recovering them in terms of KL contraction coefficients and showing that contraction is most effective near the noise distribution when input dimensions are nearly independent.

3.2 1D Experimental Validation

To validate the theoretical predictions, we analyze a controlled 1D discrete diffusion model with 15 states under a uniform corruption process. Using an analytically derived posterior $p_{0|t}$, this setting isolates sampling dynamics from model approximation error, allowing us to directly study the effect of stochasticity [26]. We compare τ-leaping using the default stochastic rate matrix and the DPF rate matrix, which we refer to as τ-leaping and DPF sampling, respectively. Figures 2a and 2b show that, in the absence of model error, DPF achieves lower error faster than τ-leaping, consistent with reduced stochasticity leading to fewer transitions and lower discretization error.

To study error propagation, we introduce model approximation errors by perturbing the score function with multiplicative noise (see Appendix D.3.1 for details). In this setting, we observe a reversal in behavior: DPF sampling becomes more sensitive to errors and ultimately performs worse than τ-leaping. This supports our theoretical claim that stochastic transitions provide an error-correcting effect that mitigates accumulated errors over time.

We further examine the role of the noise schedule in light of our analysis in Theorem 3.3. Under a geometric schedule α_t, we expect the DPF contraction factor $\sigma_{s|t} \approx 1$ near the noise distribution,

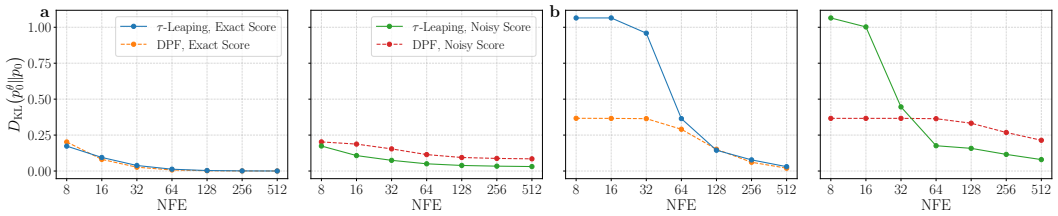


Figure 2: **Controlled 1D experiments illustrating the speed-quality tradeoff and error-correcting effect of stochasticity.** We report empirical KL divergence between the ground-truth distribution p_0 and generated distribution p_0^θ under a uniform corruption process. Results are shown for different noise schedules β_t: (**a**) linear and (**b**) geometric, and each schedule is evaluated over exact scores (left) and perturbed scores (right). With exact scores, near-deterministic sampling (DPF) converges faster due to reduced stochasticity. Under perturbed scores, stochastic sampling (τ-leaping) achieves lower final error, demonstrating its ability to correct accumulated model errors. The gap is more pronounced under the geometric schedule, which drives $\sigma_{s|t}$ closer to 1 near the noise distribution, weakening DPF error correction while making the benefits of stochastic mixing more apparent.

consequently leading to greater error accumulation under DPF sampling. As a result, the additional error correction from stochasticity in τ-leaping becomes more pronounced, leading to the observed divergence between τ-leaping and DPF.

Overall, these results illustrate the two competing effects identified in Section 3.1: reduced stochasticity improves convergence speed but increases sensitivity to model error, while higher stochasticity increases discretization errors but performs robustly at high NFE due to error-correcting effects.

4 Discrete Churn and Restart Sampling (DCRS)

Building on the analysis in Section 3, we now design a sampling algorithm that explicitly exploits the error-correcting role of stochasticity. Our results show that stochastic transitions contract errors but slow convergence, while near-deterministic transitions accelerate sampling but are prone to error accumulation. This indicates that an effective sampler should combine both behaviors: primarily deterministic updates for efficiency, together with strategically injected stochasticity for error correction. We propose *Discrete Churn and Restart Sampling* (DCRS), an inference algorithm that interleaves near-deterministic reverse updates with controlled injections of forward-process noise. DCRS consists of two complementary mechanisms: *restarting*, which introduces global stochasticity across the trajectory, and *churning*, which provides local error correction within a time window.

Restarting. DCRS initiates the generative process by simulating a reverse trajectory from the stationary distribution at $t = 1$ to a predefined time $t_{\min}$ to compute an initial state $x^0_{t_{\min}} = \mathrm{DPF}_\theta(x_1, 1 \to t_{\min})$, where $\mathrm{DPF}_\theta(x, t \to s)$ denotes running the DPF sampler from state x over the interval $[t, s]$. The algorithm then enters a *restart* phase, alternating between forward-time corruption and reverse-time updates for $i \in \{0, \dots, K-1\}$ iterations:

$$x^{i+1}_{t_{\max}} \sim q_{t_{\max}|t_{\min}}(\cdot \mid x^i_{t_{\min}}), \text{ and } x^{i+1}_{t_{\min}} = \mathrm{Churn}_\theta(x^{i+1}_{t_{\max}}, t_{\max} \to t_{\min}).$$

Here, $\mathrm{Churn}_\theta(x, t \to s)$ refers to the corresponding churning update at state x over the interval from t to s (see below). Because the forward transition $q_{t_{\max}|t_{\min}}$ admits a closed-form expression, these restart steps introduce stochasticity without additional neural network evaluations or model approximation error, providing an efficient mechanism for error correction across the trajectory.

Churning. While restarting introduces stochasticity to correct errors before the restart, it does not address error accumulation within the restart window. To address this, we introduce a finer-grained mechanism termed *churning*, which injects local stochasticity within each restart window. Given a time discretization $\{t_i\}_{i=1}^N$, the forward process advances to an intermediate time $(1+\gamma)t_i$ for some $\gamma \in (0, 1)$:

$$x_{(1+\gamma)t_i} \sim q_{(1+\gamma)t_i|t_i}(\cdot \mid x_{t_i}).$$

This injected stochasticity is followed by a single DPF sampling step from $(1+\gamma)t_i$ to t_{i+1}. In contrast to full restart iterations, which may involve multiple forward and reverse steps, the churning operation uses only one reverse update with a larger step size to compensate for the temporary forward movement in time. Churning allows for additional correction within the restart window, especially

when the error accumulation under DPF sampling is significant. Generation finishes by running $x_0 = \text{DPF}_\theta(x^K_{t_{\text{min}}}, t_{\text{min}} \to 0)$. Together, restarting and churning allow DCRS to balance efficiency and error correction. Deterministic reverse steps provide fast convergence, while forward-process perturbations periodically reintroduce stochasticity to contract accumulated errors. Further details about the sampling algorithm can be found in Algorithm 6 of Appendix C.

Impact of hyperparameters. DCRS introduces three main hyperparameters: the restart interval $[t_{\min}, t_{\max}]$, the number of restart iterations K, and the churning parameter γ. These parameters control where and how stochasticity is injected, and therefore directly govern the tradeoff between error correction and discretization error. In particular, we find that hyperparameter selection strategies from continuous diffusion samplers generally do not transfer well to the discrete setting, due in part to parallel decoding errors across dimensions that are unique to discrete diffusion transitions.

Our theoretical analysis suggests that restarting should be most effective in regions where errors accumulate and DPF contraction is otherwise weak. Empirically, we find that optimal restart windows align with regions where discretization steps are coarsest, suggesting that the added stochasticity is effective at mitigating parallel decoding errors across dimensions.

By the same reasoning, increasing the number of restart iterations exhibits diminishing returns. While additional restarts initially improve sample quality, excessive restarts introduce additional discretization error and can ultimately degrade performance. In practice, we find that the largest marginal improvement is typically achieved after the first restart iteration.

Finally, the churning parameter γ controls the magnitude of forward-reverse perturbations within the restart window. We observe that small values of $\gamma \approx 10^{-3}$ consistently yield the best performance, suggesting that beyond a certain scale, the discretization error induced by larger DPF steps outweighs the error-correcting benefits of increased stochasticity.

Applying higher-order solvers. While higher-order solvers reduce discretization error, they do not address model approximation error and can exacerbate error accumulation under near-deterministic dynamics. To improve sample quality in large-scale datasets, we employ a higher-order solver for the DPF equation (Equation (1)) only within the restart window. Specifically, we use the trapezoidal solver outlined in Ren et al. [26], which uses a weighted combination of two reverse rate matrix evaluations to reduce discretization errors. In practice, using higher-order solvers outside the restart window often leads to catastrophic compounding of discretization errors (see Appendix D.4.3). To address this, we leverage the error-correcting effects of restarting and noise churning within the restart window. For additional details, we refer to Appendix C.

5 Experiments

To validate the theoretical predictions in Section 3, we evaluate how stochasticity affects sampling behavior across controlled and large-scale settings. We focus on two questions: (1) whether stochasticity improves sample quality through error correction, and (2) whether DCRS can leverage this effect to improve the speed-quality tradeoff in practice.

Image datasets. We use pretrained model checkpoints from Campbell et al. [3] and Zhang et al. [14] on CIFAR10 and CelebA, respectively, which employ a discretized Gaussian corruption process. We evaluate the quality of generated samples in terms of Fréchet Inception Distance (FID) over 50K samples (Figure 1e). For both datasets, we use a restart window of $[t_{\text{min}}, t_{\text{max}}] = [0.7, 0.8]$ and test a single restart iteration over varying NFE used outside of the restart window. To demonstrate how DCRS scales with NFE, we proportionally increase the number of discretization steps in the restart window relative to the total number of steps outside the window. We also use τ-leaping outside of the restart window, where a higher-order solver is used inside the window. We also observe that replacing the DPF rate matrix with the modified matrix $\hat{R}_{t,\nu_t}$ with $\nu_t = 0.005$ tends to slightly boost performance (Equation (2)). In practice, we observe that higher-order solvers are ineffective without the DPF rate matrix, and hence we do not provide such samplers as baselines in our results (Appendix D.4.5).

Overall, DCRS substantially improves the speed-quality tradeoff in the low-NFE regime, achieving comparable sample quality with significantly fewer model evaluations (20-30 steps). On CelebA, DCRS matches the best performance of τ-leaping with fewer evaluations, while on CIFAR10, we observe a small gap at high NFE. These results are consistent with our theoretical analysis: reducing

Method	MAUVE (↑)			Gen PPL. (↓)			Entropy (↑)		
	T=256	*T=512*	*T=1024*	*T=256*	*T=512*	*T=1024*	*T=256*	*T=512*	*T=1024*
ReMDM-Cap	0.00635	0.00618	0.00628	175.44	226.54	356.62	5.749	5.797	5.875
ReMDM-Loop	0.00697	0.00785	0.00714	135.31	141.15	175.02	5.710	5.719	5.758
MDLM	0.00762	0.00770	0.00822	112.20	106.93	105.07	5.653	5.640	5.635
	T=32	*T=64*	*T=128*	*T=32*	*T=64*	*T=128*	*T=32*	*T=64*	*T=128*
ReMDM-Cap	0.00489	0.00567	0.00603	224.55	172.19	160.28	5.762	5.734	5.728
ReMDM-Loop	0.00453	0.00501	0.00615	349.25	198.74	148.71	5.814	5.758	5.722
MDLM	0.00499	0.00556	0.00745	196.63	143.30	121.67	5.738	5.697	5.669

Table 1: Generative perplexity, entropy, and MAUVE evaluated for a masked diffusion language model (MDLM) trained on OpenWebText over 5000 sequences of length 1024.

stochasticity improves efficiency but increases error accumulation, whereas additional stochasticity can improve robustness to model approximation errors that persist in the high-NFE regime.

Language datasets. We evaluate pre-trained models from UDLM [10] and MDLM [5] on LM1B and OpenWebText, which use uniform and masking corruption processes, respectively. For MDLM, we examine the stochasticity tradeoff from adding stochasticity via ReMDM, which adjusts stochasticity through closed-form transition distributions (see Appendix B.2). We report generative perplexity for the uniform diffusion model over 1024 sequences of length 128, and for the masked diffusion model over 5000 sequences of length 1024.

In contrast to image models, we find that reducing stochasticity has a limited impact on final sample quality. In particular, our results for UDLM (see Appendix D.5, Table 9) show that deterministic and stochastic samplers (e.g., DDIM vs. D3PM) achieve comparable performance. This can be explained by the fact that these samplers avoid constant-rate approximations, further suggesting that parallel decoding errors serve as the major performance bottleneck. As a result, increasing stochasticity alone does not improve convergence in this setting. Broadly, these results suggest that improving sample quality in text diffusion models may require mechanisms that explicitly introduce transitions across dimensions to better correct for parallel decoding errors, rather than solely increasing stochasticity within each dimension.

We further observe that, while less stochastic samplers achieve lower error at low NFE, increased stochasticity does not consistently improve sample quality and can even degrade performance. In Table 1, we show that the error-correcting effects reported in ReMDM disappear without nucleus sampling. We hypothesize that nucleus sampling sharpens the posterior $p_{0|t}$, biasing samples toward high-probability modes and altering the contraction behavior of the diffusion process. Consistent with our theory, this sharpening can reduce the effective error-correcting ability of stochastic transitions by pushing the system toward a worst-case contraction regime.

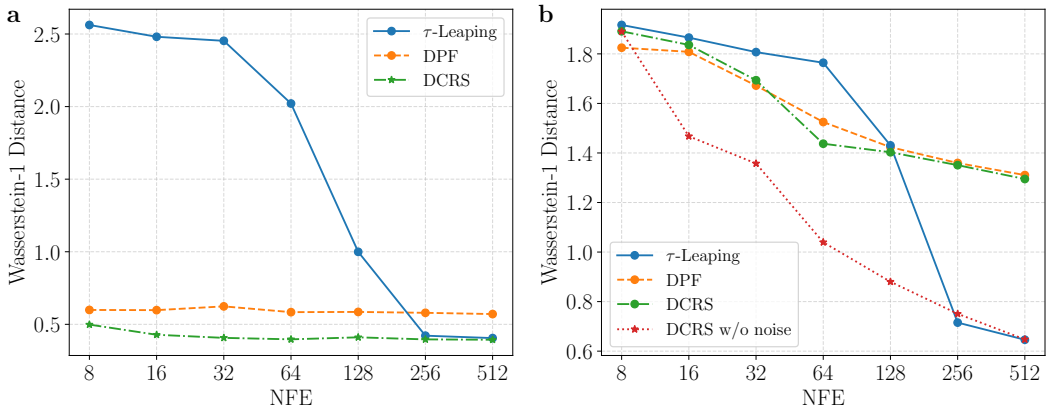


Figure 3: **Sample generation experiments on an 8D projection of a mixture of Gaussians dataset for models trained on (a) uniform and (b) masking corruption process.** Sample quality is evaluated using the Wasserstein-1 distance between 10K generated samples and the ground-truth distribution. DCRS achieves the best speed-quality tradeoff through error correction.

Mixture of Gaussians. We study a controlled mixture-of-Gaussians setting to directly visualize the effect of stochasticity on mode coverage and error correction. We train MLP-based diffusion models on a 2D dataset with 8 modes and, following prior work [14, 27], quantize the data into bit strings. We add high-dimensional structure by projecting the data into 8 dimensions using a random Gaussian projection, yielding 128-bit sequences (see Appendix D.3.2). To accentuate model errors, we train models with a linear noise schedule and perform inference with a geometric schedule, which increases step sizes near the data distribution and amplifies error accumulation.

In Figure 3a, we observe a clear tradeoff between convergence speed and sample quality for the uniform diffusion model. The τ-leaping sampler converges slowly but produces well-distributed samples across modes, whereas the near-deterministic DPF sampler immediately generates samples near the ground-truth distribution (within $\sim$8 NFE) but fails to improve with NFE, reflecting sensitivity to accumulated errors. In contrast, DCRS restores sample quality by reintroducing stochasticity.

We evaluate sample quality using the Wasserstein-1 distance between 10K generated samples and the ground-truth distribution. Across both uniform and masking corruption processes, DCRS achieves improved speed-quality tradeoffs by balancing fast convergence with error correction. For DCRS, we use restart windows of $[t_{\text{min}}, t_{\text{max}}] = [0.01, 0.05]$ (uniform) and $[0.1, 0.2]$ (masking), with the number of discretization steps inside the restart window matched to the main trajectory. In this controlled setting, we find that restarting alone provides most of the benefit, while additional noise churning or higher-order solvers can introduce unnecessary discretization error.

Impact of temperature sampling. For the masked diffusion experiments in Figure 3b, we sample from the sharpened score $(\frac{p_t(y)}{p_t(x)})^{1/\tau}$ for $\tau = 0.8$. The overall effect is equivalent to scaling marginal logits $\tau^{-1} \log p_t(y)$, analogous to temperature sampling. We observe that stochasticity alone does not yield error-correcting benefits in this setting; instead, these effects emerge only in conjunction with temperature scaling. This can be explained by the structure of masking diffusion: without additional stochasticity, sequences remain highly masked until late in the trajectory ($t \approx 0.1$), limiting opportunities for mixing and error contraction. Temperature scaling sharpens the posterior, further concentrating probability mass and reducing the effectiveness of stochastic transitions in correcting errors. To counteract this, we introduce targeted stochasticity using a schedule $\nu_t = 20 \cdot \mathbf{1}_{t<0.1}$, which improves sample quality at high NFE by encouraging mixing earlier in the trajectory. However, we find that restarting alone, without additional noise, often yields stronger improvements than combining restarting with stochastic perturbations. This suggests that resetting the trajectory to earlier time steps allows the reverse CTMC to revisit higher-entropy regions, where contraction is stronger and mixing between modes is more effective.

6 Related Work

Accelerating Discrete Diffusion Inference. Prior work has achieved strong performance in the low-NFE regime of discrete diffusion models through *distillation methods* [11, 28, 29], which require retraining models using pre-trained samplers as targets. In contrast, our work focuses on improving the efficiency of *training-free* sampling methods. A complementary line of work develops higher-order solvers that approximate transition dynamics using Taylor expansions, including Runge-Kutta and trapezoidal methods [26]. These approaches trade additional neural network evaluations for reduced discretization error. Corrector-based methods [3, 30] further improve sampling by introducing Markov Chain Monte Carlo updates that preserve the time marginals. Unlike these approaches, which primarily address discretization error, our method targets the accumulation of *model approximation errors* through controlled stochasticity. Notably, higher-order solvers and corrector iterations are complementary to DCRS and can be combined to further improve performance.

Understanding Error Correction in Diffusion Models. In continuous diffusion models, stochasticity has been observed to improve sample quality by mitigating error accumulation, for example, through noise injection during sampling [18, 31]. Theoretical work [31] shows that stochasticity can induce contraction in the Wasserstein-1 distance. In discrete settings, prior work [10, 13, 32] has noted that uniform diffusion processes can iteratively correct errors, while masking-based processes involve irreversible transitions. Similar observations have been made in discrete flow-matching, where redundant transitions can improve performance. However, these effects are largely understood heuristically. In contrast, we provide a principled characterization of error correction in discrete diffusion through KL contraction analysis and show how stochasticity can be explicitly controlled to

mitigate the accumulation of *model approximation errors* during inference. We demonstrate that the effectiveness of stochasticity depends on the structure of the corruption process and validate these findings empirically across controlled and large-scale settings.

7 Conclusion

We investigated the role of stochasticity in discrete diffusion models and showed that it induces an error-correcting mechanism that mitigates the accumulation of model approximation errors during sampling. Building on this insight, we proposed Discrete Churn and Restart Sampling (DCRS), a training-free algorithm that balances fast deterministic updates with strategically injected stochasticity to improve the speed-quality tradeoff, particularly in the low-NFE regime. Our theoretical and empirical results highlight that the effectiveness of stochasticity depends on both its magnitude and the structure of the corruption process, suggesting that controlling stochasticity is a fundamental design principle for discrete diffusion inference. A limitation of our approach is the need to tune the level and schedule of stochasticity. An interesting direction for future work is to develop adaptive methods for optimizing this tradeoff, as well as to further characterize optimal sampling rates through information-theoretic analysis.

References

[1] 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.

[2] 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.

[3] 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.

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

[5] Subham Sekhar Sahoo et al. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[6] Shuyang Gu et al. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10696–10706, June 2022.

[7] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[8] Alan Nawzad Amin, Nate Gruver, and Andrew Gordon Wilson. Why masking diffusion works: Condition on the jump schedule for improved discrete diffusion. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[9] Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z. Pan, Hyeji Kim, Sham Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction, 2025.

[10] Yair Schiff et al. Simple guidance mechanisms for discrete diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[11] Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Forty-second International Conference on Machine Learning*, 2025.

[12] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[13] Guanghan Wang, Yair Schiff, Subham Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[14] Pengze Zhang, Hubery Yin, Chen Li, and Xiaohua Xie. Formulating discrete probability flow through optimal transport. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[15] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi S. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *The Forty-First International Conference on Machine Learning*, 2024.

[16] Itai Gat et al. Discrete flow matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[17] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[18] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

[19] Yu Cao, Jingrun Chen, Yixin Luo, and Xiang Zhou. Exploring the optimal choice for generative processes in diffusion models: Ordinary vs stochastic differential equations. In *Advances in Neural Information Processing Systems*, volume 36, pages 33420–33468. Curran Associates, Inc., 2023.

[20] Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ODE is provably fast. In *Advances in Neural Information Processing Systems*, volume 36, pages 68552–68575. Curran Associates, Inc., 2023.

[21] J R Norris. Markov chains. In *Markov Chains*. Cambridge University Press, Cambridge, February 1997.

[22] Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. In *Advances in Neural Information Processing Systems*, 2025.

[23] Yinuo Ren, Haoxuan Chen, Grant M. Rotskoff, and Lexing Ying. How discrete and continuous diffusion meet: Comprehensive analysis of discrete diffusion models via a stochastic integral framework. In *NeurIPS 2024 Workshop on Mathematics of Modern Machine Learning*, 2024.

[24] Y. Polyanskiy and Y. Wu. Strong data-processing inequalities for channels and bayesian networks. *arXiv preprint arXiv:1508.06025*, 2016.

[25] M. Raginsky. Strong data processing inequalities and-sobolev inequalities for discrete channels. *arXiv preprint arXiv:1411.3575*, 2016.

[26] Yinuo Ren et al. Fast solvers for discrete diffusion models: Theory and applications of high-order algorithms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[27] Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023.

[28] Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Distillation of discrete diffusion through dimensional correlations. In *Forty-second International Conference on Machine Learning*, 2025.

[29] Justin Deschenaux and Caglar Gulcehre. Beyond autoregression: Fast LLMs via self-distillation through time. In *The Thirteenth International Conference on Learning Representations*, 2025.

[30] Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[31] Yilun Xu, Mingyang Deng, Xiang Cheng, Yonglong Tian, Ziming Liu, and Tommi S. Jaakkola. Restart sampling for improving generative processes. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[32] 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*, 2025.

[33] Neta Shaul et al. Flow matching with general discrete paths: A kinetic-optimal perspective. In *The Thirteenth International Conference on Learning Representations*, 2025.

[34] Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2025.

[35] Y. Gu and Y. Polyanskiy. Non-linear log-sobolev inequalities for the potts semigroup and applications to reconstruction problems. *Communications in Mathematical Physics*, 404(2): 769–831, December 2023. doi: 10.1007/s00220-023-04785-3.

A CTMC Background

A.1 CTMC Simulation Details

Given R_t, one can approximate the forward process of the CTMC by discretizing time with step size Δt, yielding the infinitesimal transition distribution:

$$q_{t|t-\Delta t}(y \mid x) = \delta_{x,y} + R_t(y, x)\Delta t + O(\Delta t^2), \tag{3}$$

where $\delta_{x,y}$ denotes the indicator function over $x = y$. For computational convenience, the rate matrix is often restricted to the form $R_t = \beta_t R_b$, where $R_b \in \mathbb{R}^{S^D \times S^D}$ is a fixed base rate matrix and β_t defines the noise schedule. Under this assumption, the forward transition distribution admits the following closed-form expression:

$$q_{t|s}(y \mid x) = \left[\exp\Big(\int_s^t R_\tau \, d\tau\Big)\right](y, x). \tag{4}$$

Because the state space $\mathcal{S}^D$ grows exponentially with dimension, the full rate matrix is often intractable in practice. Consequently, the reverse rate matrix is typically structured to factor across dimensions:

$$\hat{R}_t(y, x) = \sum_{d=1}^{D} \frac{p_t(y^d \mid x^{\setminus d})}{p_t(x^d \mid x^{\setminus d})} R_t^d(y^d, x^d), \tag{5}$$

where $x^{\setminus d}$ denotes all coordinates except d, and $R_t^d \in S \times S$ is a single-site rate matrix acting on dimension d. Under this factorization, the reverse process is a composition of conditionally independent processes governed by R_t^d. Sampling can therefore be performed independently for each dimension, incurring at most second-order discretization error:

$$q_{t|t-\Delta t}(y^{1:D} \mid x^{1:D}) = \prod_{d=1}^{D} q_{t|t-\Delta t}(y^d \mid x^{1:D}) + O(\Delta t^2). \tag{6}$$

Herein, we divide the main algorithms for sampling into discretization samplers, such as τ-leaping and Euler steps that assume a constant rate matrix over each sampling step, and analytic samplers, such as Tweedie τ-leaping and the D3PM posterior.

A.1.1 Discretization Samplers

In principle, samples can be generated via discretizing the infinitesimal transition distribution defined in Equation (3), using a first-order linearization that assumes a constant rate matrix over some interval of length Δt.

When the state space lacks ordering, such as in language modeling, the naive approach of simulating the CTMC through sampling from the categorical distribution defined by $\delta_{x_t,y} + \Delta t R_t(x_t, y)$ corresponds to the current state x_t. If step sizes h are not sufficiently small, the computed distribution may not be a valid distribution, and care must be taken to adaptively select interval sizes to avoid this [16].

When the state space is well-ordered, such as the pixel values of an image, τ-leaping can improve upon the efficiency of Euler steps by simulating transitions via independent Poisson arrivals across dimensions. Specifically, samples are updated according to:

$$x_{t-\Delta t} = x_t + \sum_{y \in B_1(x_t)} P_{t,y}(y - x_t),$$

where $B_1(x_t)$ denotes the Hamming-1 neighborhood of x_t and $P_{t,y} \sim \text{Poisson}(\Delta t\, \hat{R}_t(y, x_t))$. Crucially, τ-leaping sums multiple jumps together, allowing for more efficient traversal of the state space.

A.1.2 Analytic Samplers

D3PM uses Bayes rule to derive a marginally consistent time-reversal of the discrete time forward process, given by

$$q(x_s \mid x_t, x_0) = \frac{q(x_t \mid x_s) q(x_s \mid x_0)}{q(x_t \mid x_0)}. \tag{7}$$

Using an estimate of the posterior $p_{0|t}$, x_0 can be marginalized out to sample from the closed-form transition distribution. When only evaluation of the discrete score function is possible, [4] proposed Tweedie τ-leaping, which uses the score function and the known forward transition distribution to reconstruct the posterior in Equation (7).

Although the closed-form transition distribution does not suffer from discretization errors incurred from assuming a constant rate matrix, the samplers still incur $O(\Delta t^2)$ errors from sampling multiple dimensions independently in parallel (Equation (5,6)).

B Theoretical Results

B.1 Discrete Probability Flow

For clarity, we present a proof that the DPF rate matrix indeed preserves the time marginals in Theorem B.1 below:

Theorem B.1 (DPF Preserves Time Marginals)**.** *The DPF rate matrix given by Equation (1) preserves time marginals p_t generated by the forward rate matrix R_t.*

Proof. The time marginal distributions p_t are generated from R_t if and only if they satisfy the Kolmogorov Forward Equation

$$\left[\frac{dp_t}{dt}\right]_x = [R_t^T p_t]_x = \sum_{y \neq x} \big(p_t(y) R_t(y, x) - p_t(x) R_t(x, y)\big). \tag{8}$$

Under time-reversal relation, we know that the forward rate matrix corresponding to the DPF matrix satisfies:

$$R_{t,DPF}(x, y) = \frac{p_t(y)}{p_t(x)} \hat{R}_{DPF}(y, x) = \frac{(p_t(x) R_t(x, y) - p_t(y) R_t(y, x))_+}{p_t(x)}. \tag{9}$$

We then plug the DPF rate matrix into the Equation 8:

$$\begin{aligned}\left[\frac{dp_t}{dt}\right]_x &= \sum_{y\neq x}\left(p_t(y)\frac{(p_t(y)R_t(y,x)-p_t(x)R_t(x,y))_+}{p_t(y)}\right)\\ &\quad-\sum_{y\neq x}\left(p_t(x)\frac{(p_t(x)R_t(x,y)-p_t(y)R_t(y,x))_+}{p_t(x)}\right)\\ &= \sum_{y\neq x}\big(p_t(y)R_t(y,x)-p_t(x)R_t(x,y)\big).\end{aligned}$$

□

Although written in a different form, the DPF rate matrix coincides with the minimal stochasticity probability flows defined in [15, 16, 33] for uniform and masking formulations. These works focus on more general settings where the time marginals may not necessarily be generated by a Markov chain.

When R_t is symmetric, the rate matrices become

$$R_{DPF}(x,y) = \left(1-\frac{p_t(y)}{p_t(x)}\right)_+ R_t(x,y), \tag{10}$$

$$\hat{R}_{DPF}(x,y) = \left(\frac{p_t(y)}{p_t(x)}-1\right)_+ R_t(x,y). \tag{11}$$

which can be seen as a thresholding of the discrete score function such that transitions are suppressed when $p_t(y) \leq p_t(x)$. The result is a flow toward states with higher probability mass.

Defining the probability mass per unit time flowing from state x to y as $v_t(x,y) = p_t(x)R_t(x,y)$, we find that $\min\{v_t(x,y), v_t(y,x)\}$ equally flows into and out of state x and y. The DPF rate matrix can be understood as removing redundant mutual positive flow, which only serves to swap probability mass equally between state pairs. The effective new flow is set to be the remainder $v_t(x,y) - \min\{v_t(x,y), v_t(y,x)\} = (v_t(x,y) - v_t(y,x))_+$.

In particular, masking processes do not have mutual flow between state pairs, whereas uniform processes do. In the event that mutual flow is zero, additional mutual flow can be added via the stochasticity schedule ν_t in Equation 2.

B.2 DDIM / ReMDM

In this section, we adopt the notation of $\boldsymbol{x}_t \in \mathbb{R}^S$ to represent a probability vector representing the time marginal distribution p_t.

Under DDIM, the sampling update is

$$\boldsymbol{x}_s = \sigma_t \boldsymbol{x}_t + (\alpha_s - \sigma_t\alpha_t)\boldsymbol{x}_0 + ((1-\alpha_s) - (1-\alpha_t)\sigma_t)\boldsymbol{\pi}, \tag{12}$$

where $\sigma_t = \frac{1-\alpha_s}{1-\alpha_t}$ eliminates the $\boldsymbol{\pi}$ term and minimizes stochasticity. DDIM preserves the time marginals generated by the relation $\boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + (1-\alpha_t)\boldsymbol{\pi}$, which can be verified by plugging in for $\boldsymbol{x}_t$ and checking that $\boldsymbol{x}_s$ is marginally consistent.

In Theorem B.2, we demonstrate that under these corruption processes, DPF and DDIM are equivalent:

Theorem B.2 (DPF and DDIM Equivalence)**.** *For corruption process $\boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + (1-\alpha_t)\boldsymbol{\pi}$, let $\hat{R}_{t,DPF}$ be the associated DPF rate matrix. The reverse transition distribution $q_{s|t}$ induced by sampling from the DPF CTMC can be expressed in closed-form:*

$$\boldsymbol{x}_s = \left(\frac{1-\alpha_s}{1-\alpha_t}\right)\boldsymbol{x}_t + \left(1-\frac{1-\alpha_s}{1-\alpha_t}\right)\boldsymbol{x}_0, \tag{13}$$

which exactly matches the DDIM update [12] outlined for the uniform diffusion case.

Proof. Following the result of Theorem 3 of [16], we know that the conditional rate matrix is given by

$$\hat{R}_{t,\mathrm{DPF}}(\cdot,\cdot \mid x_0) = \frac{-\dot{\alpha}_t}{1-\alpha_t}\left(\mathbf{1}\boldsymbol{e}_{x_0}^T - I\right),$$

where $-\dot{\alpha}_t \geq 0$ since α_t is decreasing.

Integrating over the reverse interval from t to $s < t$,

$$\begin{aligned} Q_{s|t,x_0} &= \exp\left(\left(\int_s^t \frac{-\dot{\alpha}_\tau}{1-\alpha_\tau}\, d\tau\right)\left(\mathbf{1}\boldsymbol{e}_{x_0}^T - I\right)\right) \\ &= \left(\frac{1-\alpha_s}{1-\alpha_t}\right) I + \left(1 - \frac{1-\alpha_s}{1-\alpha_t}\right)\mathbf{1}\boldsymbol{e}_{x_0}^T, \end{aligned}$$

using $(\mathbf{1}\boldsymbol{e}_{x_0}^T - I)^2 = -(\mathbf{1}\boldsymbol{e}_{x_0}^T - I)$.

Averaging over $p_{0|t}$, we find that the marginal transition matrix is

$$Q_{s|t} = \left(\frac{1-\alpha_s}{1-\alpha_t}\right) I + \left(1 - \frac{1-\alpha_s}{1-\alpha_t}\right)\mathbf{1}\boldsymbol{x}_0^T.$$

The sampling update is then given by

$$\boldsymbol{x}_s = Q_{s|t}^T \boldsymbol{x}_t = \left(\frac{1-\alpha_s}{1-\alpha_t}\right)\boldsymbol{x}_t + \left(1 - \frac{1-\alpha_s}{1-\alpha_t}\right)\boldsymbol{x}_0.$$

□

The ReMDM sampler matches the DDIM update in Equation (12), under the reparameterization $\sigma_t = \frac{1-\alpha_s-\sigma_t^{ReMDM}}{1-\alpha_t}$. Under this choice, the update becomes

$$\boldsymbol{x}_s = \left(\frac{1-\alpha_s-\sigma_t^{ReMDM}}{1-\alpha_t}\right)\boldsymbol{x}_t + \left(\frac{\alpha_s-\alpha_t+\sigma_t^{ReMDM}\alpha_t}{1-\alpha_t}\right)\boldsymbol{x}_0 + \sigma_t^{ReMDM}\boldsymbol{\pi},$$

where σ_t^{ReMDM} controls the probability of sampling from the stationary distribution, or the masked state.

Importantly, we also demonstrate that under added stochasticity σ_t, DDIM / ReMDM decomposes into a DDIM / DPF step, followed by a corrector step that preserves time marginals. According to Theorem 4.1 of [13], every update is a composition of Equation (13) and the following corrector step that preserves the marginal at time s:

$$\boldsymbol{x}_{s_c} = \left(1 - \frac{\sigma_t^{ReMDM}}{1-\alpha_s}\right)\boldsymbol{x}_s + \frac{\sigma_t^{ReMDM}\alpha_s}{1-\alpha_s}\boldsymbol{x}_0 + \sigma_t^{ReMDM}\boldsymbol{\pi}. \tag{14}$$

In general, we can also decompose the corrector step according to

$$\boldsymbol{x}_{s_b} = \left(1 - \frac{\sigma_t^{ReMDM}\alpha_s}{(1-\alpha_s)(1-\sigma_t^{ReMDM})}\right)\boldsymbol{x}_s + \frac{\sigma_t^{ReMDM}\alpha_s}{(1-\alpha_s)(1-\sigma_t^{ReMDM})}\boldsymbol{x}_0, \tag{15}$$

$$\boldsymbol{x}_{s_c} = (1-\sigma_t^{ReMDM})\boldsymbol{x}_{s_b} + \sigma_t^{ReMDM}\boldsymbol{\pi}. \tag{16}$$

B.3 Proofs of Main Theorems

Proof of Lemma 3.1 (Forward Process Contraction).
Contraction of masking corruption.

We use the following characterization of the SDPI constant for the KL divergence in terms of mutual information:

$$\eta_{KL}(p_s, q_{t|s}) = \sup_{P_{U|X_s}: U \to X_s \to X_t} \frac{I(U;X_t)}{I(U;X_s)},$$

where we denote $X_t \sim p_t$.

For masking diffusion, we simply use proof for the binary erasure channel $S = \{0, 1, M\}$ outlined in [34] and apply it to the q-ary case. Suppose we define an auxiliary random variable $B = \mathbf{1}_{X_t=M}$, where since B is a deterministic function of X_t:

$$I(U; X_t) = I(U; X_t, B) = I(U; B) + I(U; X_t \mid B) = I(U; X_t \mid B),$$

where $I(U; B) = 0$ since $U \perp\!\!\!\perp B$.

We further find that

$$I(U; X_t \mid B) = \left(1 - \frac{\alpha_t}{\alpha_s}\right) I(U; X_t \mid B = 1) + \frac{\alpha_t}{\alpha_s} I(U; X_t \mid B = 0) = \frac{\alpha_t}{\alpha_s} I(U; X_s).$$

$I(U; X_t \mid B = 1) = 0$ since $X_t = M$ deterministically, and $I(U; X_t \mid B = 0) = I(U; X_s)$ since not masking leaves the random variable unchanged. Since this holds for all such $P_{U|X_s}$, we have $\eta_{KL}(p_s, q^M_{t|s}) = \frac{\alpha_t}{\alpha_s}$, where the contraction inequality always holds with equality.

Contraction of uniform corruption.

For uniform diffusion, deriving the SDPI constant is generally difficult, and only upper bounds are known for the general case. For $S = 2$, an upper bound is $\eta_{KL}(q_{t|s}) = \sup_{p_s} \eta_{KL}(p_s, q_{t|s}) = (\frac{\alpha_t}{\alpha_s})^2$, the proof of which we defer to [34]. An upper bound for the general case is outlined in Proposition 38 of [35], given by $\eta_{KL}(p_s, q_{t|s}) \le \eta_{KL}(q_{t|s}) \le \frac{S(\frac{\alpha_t}{\alpha_s})^2}{(S-2)\frac{\alpha_t}{\alpha_s}+2}$. For large S, the upper bound approaches the worst-case contraction $\frac{\alpha_t}{\alpha_s}$. Intuitively, under a larger number of states S, there is more room for distributions to spread out without overlapping, whereas for small S, mixing tends to create more overlap in distribution.

Masking corruption attains the worst contraction.

Following [34], we have the bounds $\eta_{KL}(p_s, q_{t|s}) \le \eta_{TV}(q_{t|s})$. In particular, suppose we let $\boldsymbol{x}_t$ and $\hat{\boldsymbol{x}}_t$ be the probability vectors representing the ground-truth and perturbed distributions, respectively. The forward process can be expressed as the update $\boldsymbol{x}_t = \frac{\alpha_t}{\alpha_s}\boldsymbol{x}_s + (1 - \frac{\alpha_t}{\alpha_s})\boldsymbol{\pi}$.

We can then analyze the contraction in TV distance directly:

$$TV(\hat{\boldsymbol{x}}_t, \boldsymbol{x}_t) = \frac{1}{2}\|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_1 = \frac{1}{2}\frac{\alpha_t}{\alpha_s}\|\hat{\boldsymbol{x}}_s - \boldsymbol{x}_s\|_1 = \frac{\alpha_t}{\alpha_s} TV(\hat{\boldsymbol{x}}_s, \boldsymbol{x}_s).$$

Since the above holds for arbitrary input distributions, this implies that $\eta_{TV}(q_{t|s}) = \frac{\alpha_t}{\alpha_s}$ and that TV distance always contracts by exactly $\frac{\alpha_t}{\alpha_s}$. Putting the above together, we conclude that $\eta_{KL}(p_s, q^M_{t|s})$ indeed achieves the upper bound. □

Proof of Lemma 3.2 (Reverse Process Contraction).
Contraction of DDIM / DPF.

We begin by analyzing the contraction in Equation (13), given by $\boldsymbol{x}_s = \sigma_{s|t}\boldsymbol{x}_t + (1 - \sigma_{s|t})\boldsymbol{x}_0(\boldsymbol{x}_t)$. To further simplify analysis, we can express the posterior as

$$\begin{aligned} p_{0|t}(x_0 \mid x_t) &= \frac{p_{t|0}(x_t \mid x_0)p_0(x_0)}{p_t(x_t)} \\ &= \frac{(\alpha_t \delta_{x_0}(x_t) + (1 - \alpha_t)\pi(x_t))p_0(x_0)}{\alpha_t p_0(x_t) + (1 - \alpha_t)\pi(x_t)} \\ &= \left(\frac{\alpha_t p_0(x_t)}{\alpha_t p_0(x_t) + (1 - \alpha_t)\pi(x_t)}\right)\delta_{x_t}(x_0) + \left(\frac{(1 - \alpha_t)\pi(x_t)}{\alpha_t p_0(x_t) + (1 - \alpha_t)\pi(x_t)}\right)p_0(x_0). \end{aligned}$$

In vector notation, we have $\boldsymbol{x}_0(\boldsymbol{x}_t) = \boldsymbol{f}_t \odot \boldsymbol{x}_t + \langle 1 - \boldsymbol{f}_t, \boldsymbol{x}_t\rangle \boldsymbol{x}_0$, where $(\boldsymbol{f}_t)_i := \frac{\alpha_t(\boldsymbol{x}_0)_i}{\alpha_t(\boldsymbol{x}_0)_i + (1-\alpha_t)\boldsymbol{\pi}_i}$.

Using joint convexity, the KL divergence contraction

$$\begin{aligned} D_{KL}(\hat{\boldsymbol{x}}_s, \boldsymbol{x}_s) &= D_{KL}(\sigma_{s|t}\hat{\boldsymbol{x}}_t + (1 - \sigma_{s|t})\boldsymbol{x}_0(\hat{\boldsymbol{x}}_t) \| \sigma_{s|t}\boldsymbol{x}_t + (1 - \sigma_{s|t})\boldsymbol{x}_0(\boldsymbol{x}_t)) \\ &\le \sigma_{s|t} D_{KL}(\hat{\boldsymbol{x}}_t \| \boldsymbol{x}_t) + (1 - \sigma_{s|t}) D_{KL}(\boldsymbol{x}_0(\hat{\boldsymbol{x}}_t) \| \boldsymbol{x}_0(\boldsymbol{x}_t)) \\ &\le (\sigma_{s|t} + (1 - \sigma_{s|t})\eta_{KL}(p_t, p_{0|t})) D_{KL}(\hat{\boldsymbol{x}}_t \| \boldsymbol{x}_t). \end{aligned}$$

We can find an upper bound on $\eta_{KL}(p_t, p_{0|t})$ by analyzing the contraction in TV distance. To do so, we use Dobrushin's characterization:

$$\begin{aligned}
\eta_{TV}(p_{0|t}) &= \sup_{i \neq j} TV(\boldsymbol{x}_0(\boldsymbol{e}_i), \boldsymbol{x}_0(\boldsymbol{e}_j)) \\
&= \sup_{i \neq j} TV(\boldsymbol{f}_t \odot \boldsymbol{e}_i + \langle 1 - \boldsymbol{f}_t, \boldsymbol{e}_i \rangle \boldsymbol{x}_0, \boldsymbol{f}_t \odot \boldsymbol{e}_j + \langle 1 - \boldsymbol{f}_t, \boldsymbol{e}_j \rangle \boldsymbol{x}_0) \\
&= \sup_{i \neq j} \frac{1}{2} \| ((\boldsymbol{f}_t)_i \boldsymbol{e}_i + (1 - (\boldsymbol{f}_t)_i) \boldsymbol{x}_0) - ((\boldsymbol{f}_t)_j \boldsymbol{e}_j + (1 - (\boldsymbol{f}_t)_j) \boldsymbol{x}_0) \|_1 \\
&= \sup_{i \neq j} \frac{1}{2} \| ((\boldsymbol{f}_t)_i \boldsymbol{e}_i + (\|\boldsymbol{f}_t\|_\infty - (\boldsymbol{f}_t)_i) \boldsymbol{x}_0) - ((\boldsymbol{f}_t)_j \boldsymbol{e}_j + (\|\boldsymbol{f}_t\|_\infty - (\boldsymbol{f}_t)_j) \boldsymbol{x}_0) \|_1 \\
&\leq \sup_{i \neq j} \frac{1}{2} \| (\boldsymbol{f}_t)_i \boldsymbol{e}_i + (\|\boldsymbol{f}_t\|_\infty - (\boldsymbol{f}_t)_i) \boldsymbol{x}_0 \|_1 + \frac{1}{2} \| (\boldsymbol{f}_t)_j \boldsymbol{e}_j + (\|\boldsymbol{f}_t\|_\infty - (\boldsymbol{f}_t)_j) \boldsymbol{x}_0 \|_1 \\
&= \sup_{i \neq j} \frac{(\boldsymbol{f}_t)_i + \|\boldsymbol{f}_t\|_\infty - (\boldsymbol{f}_t)_i}{2} + \frac{(\boldsymbol{f}_t)_j + \|\boldsymbol{f}_t\|_\infty - (\boldsymbol{f}_t)_j}{2} \\
&= \|\boldsymbol{f}_t\|_\infty.
\end{aligned}$$

From the above we have that $\eta_{KL}(p_t, p_{0|t}) \leq \eta_{TV}(p_{0|t}) \leq \|\boldsymbol{f}_t\|_\infty$, where

$$\|\boldsymbol{f}_t\|_\infty = \max_{x_t \in S} \frac{\alpha_t p_0(x_t)}{\alpha_t p_0(x_t) + (1 - \alpha_t)\pi(x_t)}.$$

In the case of uniform diffusion we know that $\pi(x_t) > 0$ for all states $x_t \in S$, so $\|\boldsymbol{f}_t\|_\infty < 1$ and $\eta_{KL}(p_s, q_{t|s}) < 1$ for all times $t \in (0, 1)$. In particular, we find that contraction is tightest when $\alpha_t \approx 0$ near the noise distribution and weakest when $\alpha_t \approx 1$ near the data distribution.

For masking diffusion, $\pi(x_t) = 0$ if x_t is not a masking state, so $\|\boldsymbol{f}_t\|_\infty = 1$. Since $\eta_{KL}(p_t, p_{0|t}) \leq 1$, this suggests that masking diffusion fails to contract in the worst case.

We show that this is not pessimistic, at least in TV distance, and that perturbing mass on non-masking states while preserving the mass on the masking state will result in a failure to contract. Specifically, we consider the perturbation $\hat{\boldsymbol{x}}_t$ of $\boldsymbol{x}_t$ such that states $j, k \in \mathcal{S}$ are swapped:

$$(\hat{\boldsymbol{x}}_t)_i = \begin{cases} (\boldsymbol{x}_t)_j & \text{if } i = k \\ (\boldsymbol{x}_t)_k & \text{if } i = j \\ (\boldsymbol{x}_t)_i & \text{otherwise} \end{cases}$$

The posterior distribution $p_{0|t}$ is the identity channel when x_t is non-masking, and so $TV(\boldsymbol{x}_0(\hat{\boldsymbol{x}}_t), \boldsymbol{x}_0(\boldsymbol{x}_t)) = TV(\hat{\boldsymbol{x}}_t, \boldsymbol{x}_t)$ and $\eta_{TV}(p_{0|t}) = 1$. Intuitively, masking fails to contract since non-masking tokens can never change.

Leveraging the characterization in [24], we also have $\eta_{KL}(p_t, q_{t|s}^{\mathrm{DPF}}) = \eta_{KL}(p_t, p_{0|t}) = 1$ if and only if $p_{0|t}$ is not fully supported over all values in $\mathcal{S}$, which is true for masking diffusion. Hence the upper bound is indeed tight for masking diffusion.

In general, the upper bound $\eta_{KL}(p_s, q_{t|s}) \leq \sigma_{s|t} + (1 - \sigma_{s|t})\eta_{KL}(p_t, p_{0|t})$ is not always tight, but tightening the bound may require stronger assumptions such as independence $\boldsymbol{x}_0(\boldsymbol{x}_t) = \boldsymbol{x}_0$, which is only true at the noise distribution $t = 1$. In this case, when $\boldsymbol{x}_0 = \boldsymbol{e}_i$ for some state $i \in \mathcal{S}$, the transition update is a masking process and the upper bound is attained at $\sigma_{s|t}$. When $\boldsymbol{x}_0 = \frac{1}{S}\mathbf{1}$ the transition is a uniform process and the upper bound can be sharpened to $\frac{S(\sigma_{s|t})^2}{(S-2)\sigma_{s|t}+2}$.

Contraction under stochasticity.

Under the stochasticity schedule ν_t, we can express the Euler step using a marginally consistent update:

$$\boldsymbol{x}_s = \left(1 + \frac{(\alpha_t - \alpha_s)(\nu_t - \alpha_t - 2\alpha_t\nu_t)}{\alpha_t(1 - \alpha_t)}\right)\boldsymbol{x}_t + \frac{(1 + \nu_t)(\alpha_t - \alpha_s)}{1 - \alpha_t}\boldsymbol{x}_0 + \frac{\nu_t(\alpha_s - \alpha_t)}{\alpha_t}\boldsymbol{\pi}, \quad (17)$$

which corresponds to $\sigma_t^{ReMDM} = \frac{\nu_t(\alpha_s - \alpha_t)}{\alpha_t}$ [13]. While in principle ν_t can be driven arbitrarily large, the above update requires that the interval $[s, t]$ be chosen such that $\sigma_t^{ReMDM} \leq \min\{1, \frac{1-\alpha_s}{\alpha_t}\}$ to ensure that the update samples from a valid probability distribution.

We use the fact that the step can be decomposed into a DDIM update from time t to s, followed by a corrector step that preserves the marginal at time s in Equation (14). The corrector step itself decomposes into a DDIM step and a forward step in Equations (15,16). Let $q^b_{s|t}$ denote the DPF step followed by the backward corrector step (15), and $q^f_{t|s}$ the forward step (16).

Using the fact that under compositions of Markov chains, SDPI constants are upper-bounded multiplicatively, and hence

$$\eta_{KL}(p_t, q^{\nu_t}_{s|t}) \le \eta_{KL}(p_t, q^b_{s|t})\eta_{KL}(q^f_{t|s}).$$

Since the backward step reuses the posterior prediction made at time t, joint convexity gives the upper bounds

$$\begin{aligned}
\eta_{KL}(p_t, q^b_{s|t}) &\le \left(1 - \left(\frac{\frac{\nu_t(\alpha_s-\alpha_t)}{\alpha_t}}{1-\frac{\nu_t(\alpha_s-\alpha_t)}{\alpha_t}}\right)\left(\frac{\alpha_s}{1-\alpha_s}\right)\right)\eta_{KL}(p_t, q^{\mathrm{DPF}}_{s|t}) \\
&\quad + \left(\frac{\frac{\nu_t(\alpha_s-\alpha_t)}{\alpha_t}}{1-\frac{\nu_t(\alpha_s-\alpha_t)}{\alpha_t}}\right)\left(\frac{\alpha_s}{1-\alpha_s}\right)\eta_{KL}(p_t, p_{0|t}) \\
&=: \sigma^b_{s|t}\eta_{KL}(p_t, q^{\mathrm{DPF}}_{s|t}) + (1-\sigma^b_{s|t})\eta_{KL}(p_t, p_{0|t}), \\
\eta_{KL}(q^f_{t|s}) &\le \left(1 - \frac{\nu_t(\alpha_s-\alpha_t)}{\alpha_t}\right) \\
&=: \sigma^f_{t|s}
\end{aligned}$$

In general, the contraction coefficient can be set to the minimum value of $\alpha_s\eta_{KL}(p_t, p_{0|t})$ by setting $\sigma^{ReMDM}_t = \frac{\nu_t(\alpha_s-\alpha_t)}{\alpha_t} = 1-\alpha_s$, where $1-\alpha_s \le \min\{1, \frac{1-\alpha_s}{\alpha_t}\}$ is always satisfied. This corresponds to a stochasticity schedule of $\nu_t = \frac{\alpha_t(1-\alpha_s)}{\alpha_s-\alpha_t}$. In light of Equation (13), the update reduces to $\alpha_s \boldsymbol{x}_0(\boldsymbol{x}_t) + (1-\alpha_s)\boldsymbol{\pi}$. That is, samples rely entirely on the estimated posterior at time step t. However, this is not necessarily useful in practice due to the additional errors in the posterior estimate. □

Proof of Theorem 3.3 (Contraction Under Model Error).
Contraction of DDIM / DPF.

We first focus on showing the DPF sampling bound, where the per-step marginal error bound follows from the triangle inequality:

$$\begin{aligned}
TV(p^\theta_{t_{k+1}}, p_{t_{k+1}}) &= TV(\boldsymbol{x}^\theta_{t_{k+1}}, \boldsymbol{x}_{t_{k+1}}) \\
&\le \sigma_{t_{k+1}|t_k} TV(\boldsymbol{x}^\theta_{t_k}, \boldsymbol{x}_{t_k}) + (1-\sigma_{t_{k+1}|t_k})TV(\boldsymbol{x}^\theta_0(\boldsymbol{x}^\theta_{t_k}), \boldsymbol{x}_0(\boldsymbol{x}_{t_k}))
\end{aligned}$$

and

$$\begin{aligned}
TV(\boldsymbol{x}^\theta_0(\boldsymbol{x}^\theta_{t_k}), \boldsymbol{x}_0(\boldsymbol{x}_{t_k})) &\le TV(\boldsymbol{x}^\theta_0(\boldsymbol{x}^\theta_{t_k}), \boldsymbol{x}_0(\boldsymbol{x}^\theta_{t_k})) + TV(\boldsymbol{x}_0(\boldsymbol{x}^\theta_{t_k}), \boldsymbol{x}_0(\boldsymbol{x}_{t_k})) \\
&\le TV(p^\theta_{0|t_k}, p_{0|t_k} \mid p^\theta_{t_k}) + \eta_{TV}(p_{0|t_k})TV(\boldsymbol{x}^\theta_{t_k}, \boldsymbol{x}_{t_k}) \\
&= \epsilon_k + \eta_{TV}(p_{0|t_k})TV(p^\theta_{t_k}, p_{t_k}).
\end{aligned}$$

Putting the above together, we have

$$TV(p^\theta_{t_{k+1}}, p_{t_{k+1}}) \le (\sigma_{t_{k+1}|t_k} + (1-\sigma_{t_{k+1}|t_k})\eta_{TV}(p_{0|t_k}))TV(p^\theta_{t_k}, p_{t_k}) + (1-\sigma_{t_{k+1}|t_k})\epsilon_k.$$

Contraction under stochasticity.

Under the stochasticity schedule ν_t, we treat the step updates as a composition, which in the proof of Lemma 3.2 we showed to be true in the absence of discretization errors. We then apply the triangle

inequality in succession to separate the model errors:

$$\begin{aligned}
\mathrm{TV}(p^\theta_{t_k,b}, p_{t_k,b}) &\le (\sigma_{t_{k+1}|t_k} + (1-\sigma_{t_{k+1}|t_k})\eta_{\mathrm{TV}}(p_{0|t_k}))\,\mathrm{TV}(p^\theta_{t_k}, p_{t_k}) + (1-\sigma_{t_{k+1}|t_k})\epsilon_k \\
&:= A_{k,d}\mathrm{TV}(p^\theta_{t_k}, p_{t_k}) + B_{k,d}\epsilon_k \\
\mathrm{TV}(p^\theta_{t_k,f}, p_{t_k,f}) &\le \sigma^b_{t_{k+1}|t_k}\,\mathrm{TV}(p^\theta_{t_k,b}, p_{t_k,b}) + (1-\sigma^b_{t_{k+1}|t_k})\left(\eta_{\mathrm{TV}}(p_{0|t_k})\,\mathrm{TV}(p^\theta_{t_k}, p_{t_k}) + \epsilon_k\right) \\
&:= A^1_{k,b}\mathrm{TV}(p^\theta_{t_k,b}, p_{t_k,b}) + A^2_{k,b}\mathrm{TV}(p^\theta_{t_k}, p_{t_k}) + B_{k,b}\epsilon_k \\
\mathrm{TV}(p^\theta_{t_{k+1}}, p_{t_{k+1}}) &\le \sigma^f_{t_{k+1}|t_k}\,\mathrm{TV}(p^\theta_{t_k,f}, p_{t_k,f}) \\
&:= A_{k,f}\mathrm{TV}(p^\theta_{t_k,f}, p_{t_k,f}),
\end{aligned}$$

where $\sigma^b_{t_{k+1}|t_k}$ and $\sigma^f_{t_{k+1}|t_k}$ are coefficients associated with the corrector steps, defined in the proof of Lemma 3.2.

The overall update reduces to

$$TV(p^\theta_{t_{k+1}}, p_{t_{k+1}}) \le (A_{k,d}A^1_{k,b}A_{k,f} + A^2_{k,b}A_{k,f})TV(p^\theta_{t_k}, p_{t_k}) + (A_{k,f}A^1_{k,b}B_{k,d} + A_{k,f}B_{k,b})\epsilon_k.$$

If $\nu_t = \frac{\alpha_t(1-\alpha_s)}{\alpha_s-\alpha_t}$, the update each step is $\boldsymbol{x}_{t_{k+1}} = \alpha_{t_{k+1}}\boldsymbol{x}_0(\boldsymbol{x}_{t_k}) + (1-\alpha_{t_{k+1}})\boldsymbol{\pi}$. The inequality then reduces to

$$TV(p^\theta_{t_{k+1}}, p_{t_{k+1}}) \le \alpha_{t_{k+1}}\eta_{TV}(p_{0|t_k})TV(p^\theta_{t_k}, p_{t_k}) + \alpha_{t_{k+1}}\epsilon_k.$$

In particular, we know that $\alpha_{t_{k+1}} \ge (1-\sigma_{t_{k+1}|t_k})$. This suggests that the stochasticity schedule ν_t uses updates that rely on the posterior $\boldsymbol{x}_0(\boldsymbol{x}_{t_k})$ at least as much as that of the DPF updates, resulting in a larger ϵ_k coefficient. On the other hand, if the posterior is not contractive and $\eta_{TV}(p_{0|t_k}) = 1$, as is the case under masking diffusion, the stochasticity schedule ν_t can ensure that the contraction is no more than $\alpha_{t_{k+1}}$.

In both cases, this reduces to

$$TV(p^\theta_{t_{k+1}}, p_{t_{k+1}}) \le A_k TV(p^\theta_{t_k}, p_{t_k}) + B_k\epsilon_k.$$

for some coefficients A_k and B_k.

The total error bound over sampling K steps proceeds via induction by successively applying upper bounds on the per-step marginal error:

$$TV(p^\theta_0, p_0) \le \left(\prod_{k=1}^K A_k\right) TV(p^\theta_{t_1}, p_{t_1}) + \sum_{k=1}^K \left(\prod_{j=k+1}^K A_j\right) B_k\epsilon_k = \sum_{k=1}^K \left(\prod_{j=k+1}^K A_j\right) B_k\epsilon_k,$$

where $p^\theta_{t_1} = p_{t_1}$ since the noise distribution at $t=1$ is always sampled exactly.

DCRS Error Bound.

Under one restart of DCRS without any churning steps, the sampling bound for DPF carries over, except the forward process contracts by a factor of $\frac{\alpha_{t_{\max}}}{\alpha_{t_{\min}}}$. Suppose that $t_{\min} = t_{k_{\min}}$ and $t_{\max} = t_{k_{\max}}$.

$$TV(p^\theta_0, p_0) \le \frac{\alpha_{t_{\max}}}{\alpha_{t_{\min}}}\left(\prod_{j=k_{\max}}^K A_j\right)\sum_{k=1}^{k_{\min}-1}\left(\prod_{j=k+1}^{k_{\min}-1} A_j\right) B_k\epsilon_k + \sum_{k=k_{\max}}^K \left(\prod_{j=k+1}^K A_j\right) B_k\epsilon_k.$$

□

B.4 Extension to higher dimensions

Theorem B.3 (KL Contraction Under Model Error)**.** *Let $1 = t_1 > \cdots > t_{K+1} = 0$ be a time discretization, and suppose the marginal distributions evolve according to*

$$p_{t_{k+1}} = p_{t_k}p_{t_{k+1}|t_k}, \qquad p^\theta_{t_{k+1}} = p^\theta_{t_k}p^\theta_{t_{k+1}|t_k},$$

Suppose the exact transition admits the posterior representation

$$p_{t_{k+1}|t_k}(y \mid x) = \sum_{\boldsymbol{x}_0} p_{0|t_k}(\boldsymbol{x}_0 \mid x) p_{t_{k+1}|t_k,0}(y \mid x, \boldsymbol{x}_0).$$

Define the parallel transition using the product of the exact posterior marginals,

$$p^{\mathrm{par}}_{t_{k+1}|t_k}(y \mid x) = \sum_{\boldsymbol{x}_0} \prod_{i=1}^{D} p^{(i)}_{0|t_k}(x_0^i \mid x) p_{t_{k+1}|t_k,0}(y \mid x, \boldsymbol{x}_0),$$

and let the learned parallel transition be

$$p^{\theta}_{t_{k+1}|t_k}(y \mid x) = \sum_{\boldsymbol{x}_0} \prod_{i=1}^{D} p^{\theta,(i)}_{0|t_k}(x_0^i \mid x) p_{t_{k+1}|t_k,0}(y \mid x, \boldsymbol{x}_0).$$

Assume the following conditions hold:

1. ***Posterior model error.*** *For every $k = 1, \ldots, K$,*

$$\mathbb{E}_{x \sim p^{\theta}_{t_k}} \left[\sum_{i=1}^{D} D_{\mathrm{KL}} \left(p^{\theta,(i)}_{0|t_k}(\cdot \mid x) \middle\| p^{(i)}_{0|t_k}(\cdot \mid x) \right) \right] \le \epsilon_k^2.$$

2. ***Parallel decoding error.*** *Define*

$$\delta_k^2 = \mathbb{E}_{x \sim p^{\theta}_{t_k}} \left[D_{\mathrm{KL}} \left(p_{0|t_k}(\cdot \mid x) \middle\| \prod_{i=1}^{D} p^{(i)}_{0|t_k}(\cdot \mid x) \right) \right].$$

3. ***Conditional decoder contraction.*** *For every x, the conditional transition $p_{t_{k+1}|t_k,0}(\cdot \mid x, \boldsymbol{x}_0)$, viewed as a channel from $\boldsymbol{x}_0$ to y, satisfies*

$$\sup_x \eta_{\mathrm{KL}} \left(p_{t_{k+1}|t_k,0}(\cdot \mid x, \cdot) \right) \le \xi_k.$$

4. ***Transition contraction.*** *Each exact transition satisfies*

$$\eta_{KL}(p_{t_{k+1}|t_k}) \le \eta_k.$$

Then the final sampling error satisfies

$$\mathrm{TV}(p^{\theta}_{t_{K+1}}, p_{t_{K+1}}) \le \frac{1}{\sqrt{2}} \sum_{k=1}^{K} \sqrt{\xi_k}(\epsilon_k + \delta_k) \prod_{j=k+1}^{K} \sqrt{\eta_j}.$$

Remark. The previous dependence on stochasticity manifests through the contraction coefficient of the conditional decoder ξ_k, where the overall contraction of the conditional decoder and posterior manifests through η_k.

Contraction of the conditional decoder in this case is now highly dependent on the input distributions and must be handled carefully. In particular, since the conditional decoder is a product of site-wise channels, we obtain the dimension-dependent bound

$$\eta_{\mathrm{KL}}(q_{t|s}^{\otimes D}) \le 1 - \left(1 - \eta_{\mathrm{KL}}(q_{t|s})\right)^D.$$

This guarantees strict contraction whenever the one-dimensional channel is strictly contractive, but it also exposes a worst-case curse of dimensionality: the coefficient approaches one exponentially fast in dimension D. This behavior is unavoidable without assumptions on the input dependence structure; for example, highly coupled distributions supported on repeated-symbol sequences can mix extremely slowly under coordinate-wise corruption (e.g. AAAA, BBBB). Under additional assumptions that relax how tightly coupled the dimensions are, the exponent can be reduced depending on the maximum number of dimensions a particular coordinate influences or is influenced by [24]. In particular, under independent inputs, we recover stronger contraction

$$\eta_{\mathrm{KL}}(q_{t|s}^{\otimes D}, q_s^{\otimes D}) = \eta_{\mathrm{KL}}(q_{t|s}, q_s).$$

Proof. **Proof outline.**

The proof proceeds in three stages. First, a telescoping argument and the triangle inequality in TV separate the error introduced at each step into a posterior model error and a parallel decoding error (Step 1). Next, Pinsker's inequality converts each term to KL divergence, where the decoder contraction ξ_k and the transition contraction $\prod_{j>k} \eta_j$ bound each error source (Steps 2–3). Finally, summing the per-step contributions yields the global bound (Step 4).

Step 1: Telescoping decomposition.

For $1 \le k \le K+1$, define the hybrid distribution

$$\widetilde{p}^{(k)}_{t_{K+1}} = p^{\theta}_{t_k} p_{t_{K+1}|t_k},$$

where $p_{t_{K+1}|t_k}$ denotes the composition of the exact transition kernels from t_k to t_{K+1}, with the convention $p_{t_{K+1}|t_{K+1}} = I$.

Since $p^{\theta}_{t_1} = p_{t_1}$, $\widetilde{p}^{(1)}_{t_{K+1}} = p_{t_{K+1}}$ and $\widetilde{p}^{(K+1)}_{t_{K+1}} = p^{\theta}_{t_{K+1}}$.

By the triangle inequality,

$$\mathrm{TV}(p^{\theta}_{t_{K+1}}, p_{t_{K+1}}) \le \sum_{k=1}^{K} \mathrm{TV}\left(\widetilde{p}^{(k+1)}_{t_{K+1}}, \widetilde{p}^{(k)}_{t_{K+1}}\right). \tag{18}$$

For each k, we have

$$\widetilde{p}^{(k+1)}_{t_{K+1}} = p^{\theta}_{t_k} p^{\theta}_{t_{k+1}|t_k} p_{t_{K+1}|t_{k+1}}, \qquad \widetilde{p}^{(k)}_{t_{K+1}} = p^{\theta}_{t_k} p_{t_{k+1}|t_k} p_{t_{K+1}|t_{k+1}}.$$

Introduce the intermediate distribution

$$\overline{p}^{(k)}_{t_{K+1}} = p^{\theta}_{t_k} p^{\mathrm{par}}_{t_{k+1}|t_k} p_{t_{K+1}|t_{k+1}}.$$

Applying the triangle inequality again,

$$\mathrm{TV}\left(\widetilde{p}^{(k+1)}_{t_{K+1}}, \widetilde{p}^{(k)}_{t_{K+1}}\right) \le \mathrm{TV}\left(\widetilde{p}^{(k+1)}_{t_{K+1}}, \overline{p}^{(k)}_{t_{K+1}}\right) + \mathrm{TV}\left(\overline{p}^{(k)}_{t_{K+1}}, \widetilde{p}^{(k)}_{t_{K+1}}\right). \tag{19}$$

The first term represents posterior model error, while the second represents parallel decoding error.

Step 2: Bounding the posterior model error.

By Pinsker's inequality,

$$\mathrm{TV}\left(\widetilde{p}^{(k+1)}_{t_{K+1}}, \overline{p}^{(k)}_{t_{K+1}}\right) \le \left[\frac{1}{2} D_{\mathrm{KL}}\left(p^{\theta}_{t_k} p^{\theta}_{t_{k+1}|t_k} p_{t_{K+1}|t_{k+1}} \middle\| p^{\theta}_{t_k} p^{\mathrm{par}}_{t_{k+1}|t_k} p_{t_{K+1}|t_{k+1}}\right)\right]^{1/2}.$$

By successive applications of KL-SDPI,

$$\begin{aligned} &D_{\mathrm{KL}}\left(p^{\theta}_{t_k} p^{\theta}_{t_{k+1}|t_k} p_{t_{K+1}|t_{k+1}} \middle\| p^{\theta}_{t_k} p^{\mathrm{par}}_{t_{k+1}|t_k} p_{t_{K+1}|t_{k+1}}\right) \\ &\le \left(\prod_{j=k+1}^{K} \eta_j\right) D_{\mathrm{KL}}\left(p^{\theta}_{t_k} p^{\theta}_{t_{k+1}|t_k} \middle\| p^{\theta}_{t_k} p^{\mathrm{par}}_{t_{k+1}|t_k}\right). \end{aligned} \tag{20}$$

By joint convexity of KL divergence,

$$D_{\mathrm{KL}}\left(p^{\theta}_{t_k} p^{\theta}_{t_{k+1}|t_k} \middle\| p^{\theta}_{t_k} p^{\mathrm{par}}_{t_{k+1}|t_k}\right) \le \mathbb{E}_{x \sim p^{\theta}_{t_k}}\left[D_{\mathrm{KL}}\left(p^{\theta}_{t_{k+1}|t_k}(x,\cdot) \middle\| p^{\mathrm{par}}_{t_{k+1}|t_k}(x,\cdot)\right)\right].$$

For fixed x, both transitions are obtained by applying the same conditional decoder $p_{t_{k+1}|t_k,0}$ to different posterior distributions.

By the conditional decoder KL-SDPI,

$$D_{\mathrm{KL}}\left(p^{\theta}_{t_{k+1}|t_k}(x,\cdot)\middle\|p^{\mathrm{par}}_{t_{k+1}|t_k}(x,\cdot)\right)\le \xi_k D_{\mathrm{KL}}\left(\prod_{i=1}^{D} p^{\theta,(i)}_{0|t_k}(\cdot\mid x)\middle\|\prod_{i=1}^{D} p^{(i)}_{0|t_k}(\cdot\mid x)\right).$$

Since KL divergence is additive over products,

$$D_{\mathrm{KL}}\left(\prod_{i=1}^{D} p^{\theta,(i)}_{0|t_k}(\cdot\mid x)\middle\|\prod_{i=1}^{D} p^{(i)}_{0|t_k}(\cdot\mid x)\right)=\sum_{i=1}^{D} D_{\mathrm{KL}}\left(p^{\theta,(i)}_{0|t_k}(\cdot\mid x)\middle\|p^{(i)}_{0|t_k}(\cdot\mid x)\right).$$

Taking expectations and applying the posterior model-error assumption gives

$$D_{\mathrm{KL}}\left(p^{\theta}_{t_k}p^{\theta}_{t_{k+1}|t_k}\middle\|p^{\theta}_{t_k}p^{\mathrm{par}}_{t_{k+1}|t_k}\right)\le \xi_k\epsilon_k^2.$$

Combining the preceding inequalities,

$$\mathrm{TV}\left(\widetilde{p}^{(k+1)}_{t_{K+1}},\overline{p}^{(k)}_{t_{K+1}}\right)\le \frac{1}{\sqrt{2}}\sqrt{\xi_k}\epsilon_k\prod_{j=k+1}^{K}\sqrt{\eta_j}. \tag{21}$$

Step 3: Bounding the parallel decoding error.

Similarly, by Pinsker's inequality,

$$\mathrm{TV}\left(\overline{p}^{(k)}_{t_{K+1}},\widetilde{p}^{(k)}_{t_{K+1}}\right)\le\left[\frac{1}{2}D_{\mathrm{KL}}\left(p^{\theta}_{t_k}p_{t_{k+1}|t_k}p_{t_{K+1}|t_{k+1}}\middle\|p^{\theta}_{t_k}p^{\mathrm{par}}_{t_{k+1}|t_k}p_{t_{K+1}|t_{k+1}}\right)\right]^{1/2}.$$

Notice that the direction of KL divergence has been reversed, which is permitted because total variation is symmetric.

Applying KL-SDPI to the subsequent exact transitions,

$$D_{\mathrm{KL}}\left(p^{\theta}_{t_k}p_{t_{k+1}|t_k}p_{t_{K+1}|t_{k+1}}\middle\|p^{\theta}_{t_k}p^{\mathrm{par}}_{t_{k+1}|t_k}p_{t_{K+1}|t_{k+1}}\right)\le\left(\prod_{j=k+1}^{K}\eta_j\right)D_{\mathrm{KL}}\left(p^{\theta}_{t_k}p_{t_{k+1}|t_k}\middle\|p^{\theta}_{t_k}p^{\mathrm{par}}_{t_{k+1}|t_k}\right).$$

By joint convexity,

$$D_{\mathrm{KL}}\left(p^{\theta}_{t_k}p_{t_{k+1}|t_k}\middle\|p^{\theta}_{t_k}p^{\mathrm{par}}_{t_{k+1}|t_k}\right)\le\mathbb{E}_{x\sim p^{\theta}_{t_k}}\left[D_{\mathrm{KL}}\left(p_{t_{k+1}|t_k}(x,\cdot)\middle\|p^{\mathrm{par}}_{t_{k+1}|t_k}(x,\cdot)\right)\right].$$

Both transitions again share the same conditional decoder. Hence,

$$D_{\mathrm{KL}}\left(p_{t_{k+1}|t_k}(x,\cdot)\middle\|p^{\mathrm{par}}_{t_{k+1}|t_k}(x,\cdot)\right)\le\xi_k D_{\mathrm{KL}}\left(p_{0|t_k}(\cdot\mid x)\middle\|\prod_{i=1}^{D}p^{(i)}_{0|t_k}(\cdot\mid x)\right).$$

By the definition of δ_k,

$$D_{\mathrm{KL}}\left(p^{\theta}_{t_k}p_{t_{k+1}|t_k}\middle\|p^{\theta}_{t_k}p^{\mathrm{par}}_{t_{k+1}|t_k}\right)\le\xi_k\delta_k^2.$$

Consequently,

$$\mathrm{TV}\left(\overline{p}^{(k)}_{t_{K+1}},\widetilde{p}^{(k)}_{t_{K+1}}\right)\le\frac{1}{\sqrt{2}}\sqrt{\xi_k}\delta_k\prod_{j=k+1}^{K}\sqrt{\eta_j}. \tag{22}$$

Step 4: Combining the inequalities.

Substituting equation 21 and equation 22 into equation 19, we obtain

$$\mathrm{TV}\left(\widetilde{p}^{(k+1)}_{t_{K+1}}, \widetilde{p}^{(k)}_{t_{K+1}}\right) \leq \frac{1}{\sqrt{2}}\sqrt{\xi_k}(\epsilon_k + \delta_k) \prod_{j=k+1}^{K} \sqrt{\eta_j}.$$

Finally, summing over k using equation 18 gives

$$\mathrm{TV}(p^{\theta}_{t_{K+1}}, p_{t_{K+1}}) \leq \frac{1}{\sqrt{2}} \sum_{k=1}^{K} \sqrt{\xi_k}(\epsilon_k + \delta_k) \prod_{j=k+1}^{K} \sqrt{\eta_j},$$

which completes the proof. □

B.5 Contraction for General Corruption Processes

The discretized-Gaussian corruption used for CIFAR10 and CelebA is not contained in the family of corruption processes analyzed by our theory. Nevertheless, the error-correction mechanism is not restricted to that family. For a general finite-state corruption channel $p_{t|0}$, we can evaluate its Dobrushin contraction coefficient, $\eta_{KL}(p_{t|0}) \leq \eta_{TV}(p_{t|0}) = \sup_{x,x'} TV(p_{t|0}(\cdot|x), p_{t|0}(\cdot|x'))$,

which provides a computable upper bound on its KL contraction coefficient. We evaluated this quantity for the discretized-Gaussian channel and also estimated the KL and TV contraction ratios over 1,000 randomly sampled pairs of input distributions, whose maximum lower-bounds the true contraction coefficient. The results show strict contraction away from $t = 0$:

Table 2: Empirical KL and TV contraction (maximum ratio over 1,000 random input pairs, a lower bound on the contraction coefficient) compared with the Dobrushin TV coefficient of the discretized-Gaussian channel.

t	Empirical KL contraction	Empirical TV contraction	Discrete Gaussian Dobrushin TV
0.0000	0.9999	0.9999	1.0000
0.2500	0.0514	0.2793	0.9949
0.5000	0.0218	0.1812	0.7243
0.7500	0.0014	0.0466	0.1872
1.0000	0.0000	0.0006	0.0025

Thus, the image corruption process exhibits the same qualitative contraction mechanism, although our closed-form comparison between masking and uniform corruption does not extend to it.

C Algorithms

C.1 EDM Discretization Scheme

To improve sampling quality Karras et al. [18] proposed a heuristic time discretization scheme that decreases the step sizes towards the end of the generation process. We also adopt this scheme with $\rho = 7$ when sampling with DCRS, where the discretization scheme is applied to the overall time interval $[0, 1]$, as well as within the restarted interval $[t_{\text{min}}, t_{\text{max}}]$. We present the exact mathematical formulation of the scheme in Algorithm 1.

Algorithm 1 EDM_Discretization_Scheme$(t_{\text{min}}, t_{\text{max}}, N, \rho)$

1: **return** $\left\{\left(t_{\text{max}}^{\frac{1}{\rho}} + \frac{i}{N-1}\left(t_{\text{min}}^{\frac{1}{\rho}} - t_{\text{max}}^{\frac{1}{\rho}}\right)\right)^{\rho}\right\}_{i=0}^{N-1}$

C.2 Discrete Churn and Restart Sampling

DCRS consists of running the reverse process through τ-leaping and running the forward process through sampling from the closed-form categorical distribution defined by the transition matrix

$q_{t_{\text{max}}|t_{\text{min}}}$. We present pseudocode for both restarting with the forward process and τ-leaping in Algorithms 2 and 3, respectively.

Algorithm 2 Restart_Step($x_{t_{\text{min}}}, q_{t_{\text{max}}|t_{\text{min}}}$)

$x_{t_{\max}} \sim \text{Cat}(x_{t_{\text{min}}} q_{t_{\text{max}}|t_{\text{min}}})$
return $x_{t_{\text{max}}}$

Algorithm 3 τ-Leaping_Step($t_i, t_{i+1}, x_{t_i}, \hat{R}_t(\cdot)$)

for $y \in B_1(x_{t_i})$ **do**
  $P_{y,i} \sim \text{Poisson}((t_i - t_{i+1})\hat{R}_t(y, x_{t_i}))$
end for
$x_{t_{i+1}} = x_{t_i} + \sum_{y \in B_1(x_{t_i})} P_{y,i}(y - x_{t_i})$
$x_{t_{i+1}} = \text{Clamp}\{x_{t_{i+1}}, \text{min} = 1, \text{max} = S\}$
return $x_{t_{i+1}}$

We also explicitly define churning steps in Algorithm 4, which injects a small amount of noise before every step from a solver of choice, such as τ-leaping or trapezoidal steps:

Algorithm 4 Churn_Step(t_i, x_{t_i}, γ)

$t_{\text{churn}} = (1 + \gamma)t_i$
$x_{(1+\gamma)t_i} \sim q_{(1+\gamma)t_i|t_i}(\cdot \mid x_{t_i})$
return $t_{\text{churn}}, x_{t_{\text{churn}}}$

C.3 Higher-Order Solver

As previously noted, τ-leaping steps produce errors on the order of $O(\Delta t^2)$, rendering the algorithm first-order accurate. To obtain a second-order accurate solver, Ren et al. [26] proposes using trapezoidal iterations inspired by SDE solvers. The iterations evaluate the reverse rate matrix both at the beginning of the interval and at an intermediate step. The step is then performed, taking a weighted combination of reverse rate matrix evaluations at both steps, and can be used as a plugin replacement for τ-leaping. For full details of the trapezoidal solver, we refer to Algorithm 5.

Algorithm 5 Trapezoidal_Step($t_i, t_{i+1}, x_{t_i}, \hat{R}_t(\cdot), \theta$)

$t_{\text{mid}} = \theta(t_{i+1} - t_i) + t_i$
$w = \frac{1}{2\theta(1-\theta)}$
$x_{t_{\text{mid}}} = \tau$-Leaping_Step($t_i, t_{\text{mid}}, x_{t_i}, \hat{R}_t(x_{t_i})$)
$x_{t_{i+1}} = \tau$-Leaping_Step($t_{\text{mid}}, t_{i+1}, x_{t_{\text{mid}}}, \left(w\, \hat{R}_{t_{\text{mid}}}(x_{t_{\text{mid}}}) - (w-1)\, \hat{R}_{t_i}(x_{t_i})\right)_+$)
return $x_{t_{i+1}}$

C.4 Discrete Churn and Restart Sampling Algorithm

We present the full pseudocode for DCRS, where we denote the main hyperparameters of interest N_{Main} and $N_{\text{Restart},j}$ to indicate the number of discretization steps used in the overall main interval $[t_{\text{stop}}, 1]$ and the jth restarted intervals $[t_{\text{min},j}, t_{\text{max},j}]$, respectively, for up to l different restarted intervals. We also introduce the hyperparameter K_j to indicate the number of restarts performed for the jth restart interval.

Algorithm 6 Discrete_Churn_and_Restart_Sampling
$(\pi, \rho, \gamma, t_{\text{stop}}, \hat{R}_t(\cdot), N_{\text{Main}}, (N_{\text{Restart},j}, K_j, t_{\min,j}, t_{\max,j})_{j=1}^{l})$

$\{t_i\}_{i=0}^{N_{\text{Main}}} = \text{EDM_Discretization_Scheme}(t_{\text{stop}}, 1, N_{\text{Main}} + 1, \rho)$
Round $\{t_{\min,j}\}_{j=1}^{l}$ to the nearest neighbor in $\{t_i\}_{i=0}^{N_{\text{Main}}}$
$x_{t_0} \sim \text{Cat}(\pi)$
for $i = 0, \ldots, N_{\text{Main}} - 1$ **do**
  $x_{t_{i+1}} = \tau\text{-Leaping_Step}(t_i, t_{i+1}, x_{t_i}, \hat{R}_{t_i}(\cdot))$
  if $t_{i+1} = t_{\min,j}$ for some $j \in \{1, \ldots, l\}$ **then**
    $\{s_m\}_{m=0}^{N_{\text{Restart},j}} = \text{EDM_Discretization_Scheme}(t_{\min,j}, t_{\max,j}, N_{\text{Restart},j} + 1, \rho)$
    $x^0 = x_{t_{i+1}}$
    for $k = 0, \ldots, K_j - 1$ **do**
      $y_{s_0} = \text{Restart_Step}(x^k, q_{t_{\max,j}|t_{\min,j}})$
      for $m = 0, \ldots, N_{\text{Restart},j} - 1$ **do**
        $s_{\text{churn}}, y_{s_{\text{churn}}} = \text{Churn_Step}(s_m, y_{s_m}, \gamma)$
        $y_{s_{m+1}} = \text{Trapezoidal_Step}(s_{\text{churn}}, s_{m+1}, y_{s_{\text{churn}}}, \hat{R}_t(\cdot), \frac{1}{2})$
      end for
      $x^{k+1} = y_{s_{N_{\text{Restart},j}}}$
    end for
    $x_{t_{i+1}} = x^{K_j}$
  end if
end for
$x_0 \sim \hat{p}_{0|t_{\text{stop}}}(\cdot \mid x_{t_{N_{\text{Main}}}})$ ▷ final denoising step
return x_0

D Additional Experimental Details

D.1 Tables

	NFE	FID
DPF	8	150.99
	16	51.66
	32	33.66
	64	39.43
	128	43.29
	256	45.22
	512	46.47
τ-Leaping	8	366.74
	16	330.90
	32	316.35
	64	151.22
	128	21.73
	256	7.11
	512	6.09
DCRS	12	31.33
	26	10.80
	54	11.37
	110	14.01
	222	15.32
	446	16.69

Table 3: CIFAR10

	NFE	FID
DPF	8	98.96
	16	45.22
	32	30.61
	64	33.55
	128	36.39
	256	38.15
	512	39.07
τ-Leaping	8	482.36
	16	467.12
	32	390.57
	64	281.28
	128	65.64
	256	22.46
	512	17.08
DCRS	12	53.52
	26	20.44
	54	15.90
	110	19.71
	222	23.19
	446	25.33

Table 4: CelebA

	NFE	W_1 Distance
DPF	4	0.70
	8	0.60
	16	0.60
	32	0.62
	64	0.58
	128	0.59
	256	0.58
	512	0.57
τ-Leaping	4	2.60
	8	2.56
	16	2.48
	32	2.45
	64	2.02
	128	1.00
	256	0.42
	512	0.41
DCRS	4	0.81
	8	0.50
	16	0.43
	32	0.41
	64	0.40
	128	0.41
	256	0.40
	512	0.39

Table 5: Uniform Gaussians 8D

D.2 Sampling Details

Previous works have noted that the discrete score function $\frac{p_t(y)}{p_t(x)}$ becomes poorly conditioned as t approaches 0. This is attributed to the data distribution behaving nearly deterministically conditioned on the other tokens, making the discrete score function essentially singular [3]. To mitigate this effect, iterations are typically stopped at some minimum time $t_{\text{stop}} = 10^{-3}$, which we also employ.

Because absorbing diffusion models tend not to fully unmask all tokens after sampling, we perform an additional model evaluation at t_{stop} and sample from the categorical distribution $p_{t_{\text{stop}}}(x^d \mid x^{\setminus d})$.

D.3 Toy Datasets

D.3.1 1D Dataset

Following [26], we choose a ground-truth distribution over states uniformly at random from the simplex, where the number of states is set to $S = 15$. We analytically derive the posterior $p_{0|t}$, where performance under model approximation errors is tested through random perturbations of the score function by a multiplicative constant drawn from $\text{Uniform}(0, 1)$ at every iteration, where $t \in (0, 0.1)$. The KL divergence is evaluated over 1 million samples for each experimental iteration for a given NFE, sampler, and noise schedule.

For the 1D dataset with 15 states, we employ commonly used noise schedules β_t applied to a uniform rate matrix:

Linear. The linear noise schedule simply assumes time homogeneity of the rate matrix:

$$\beta_t = 1$$
$$\int_0^t \beta(s)ds = t$$

Geometric. The geometric noise schedule defines an exponentially growing rate with exponential base b and constant factor a:

$$\beta_t = a(b^t)\ln(b)$$
$$\int_0^t \beta(s)ds = a(b^t) - a$$

where we choose $a = 3$ and $b = 100$ for our experiments.

Loglinear. The loglinear noise schedule is defined below, where we add an additional factor a to accentuate the sensitivity of samplers to model approximation errors:

$$\beta_t = \frac{a(1-\epsilon)}{1-(1-\epsilon)t}$$
$$\int_0^t \beta(s)ds = -a\ln(1-(1-\epsilon)t)$$

where we set $\epsilon = 10^{-3}$ and $a = 10$ for our experiments.

We provide a full comparison of all sampling configurations and experimental conditions in Figure 4.

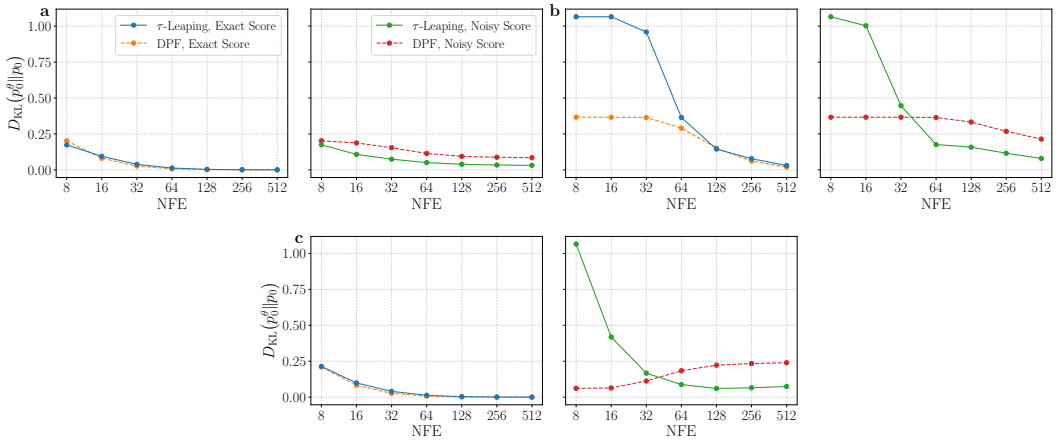


Figure 4: Empirical KL divergence between the ground-truth distribution p_{data} and generated distribution $\hat{p}_0$ for different noise schedules β_t: **(a)** linear, **(b)** geometric, and **(c)** loglinear. All experiments assume a uniform corruption process, where sampling performance is compared between using the exact discrete score function and a perturbed noisy score function.

D.3.2 Synthetic Dataset

We use a mixture of Gaussians dataset containing 8 modes in 2 dimensions. We also present results on a separate synthetic dataset projected into 8 dimensions by randomly sampling a fixed projection matrix P with independent standard Gaussian entries ($P_{ij} \sim \mathcal{N}(0,1)$). To preserve the scale of the dataset, the columns of P are normalized to have unit length. Following Zhang et al. [14], Sun et al. [27], each coordinate is then quantized and is subsequently mapped onto 16-bit binary strings according to a Gray code.

D.3.3 Experimental Setup

For all experiments, we use the Adam optimizer with a learning rate of 1e-4 and a batch size of 128. We parameterize models according to Sun et al. [27] and use a 3-layer MLP. An EMA was applied to all model weights with a constant of 0.9999. Instead of parameterizing the denoising posterior, the models instead learn the conditional distribution of each dimension $p_t(x^d \mid x^{\setminus d})$. By evaluating the model $D \times S$ times for a sequence of length D with state space size S, we can evaluate the discrete score $s_\theta(x^d, y^d) \approx \frac{p_t(y^d|x^{\setminus d})}{p_t(x^d|x^{\setminus d})}$.

All models are trained using a single NVIDIA A6000 GPU for $100,000$ iterations. For absorbing state diffusion models, we also pass the model inputs through an additional embedding matrix. We train all models with a linear noise schedule $\beta_t = 1$ and perform inference with the geometric schedule $\beta_t = ab^t \log(b)$ with $a = 3$ and $b = 100$, following Campbell et al. [3].

D.3.4 Sample Visualization

After converting from binary strings back to floating point values, we measure sample quality through the Wasserstein-1 distance under the Euclidean distance metric. Performance comparisons are made between 10000 samples from both the ground-truth and learned distributions.

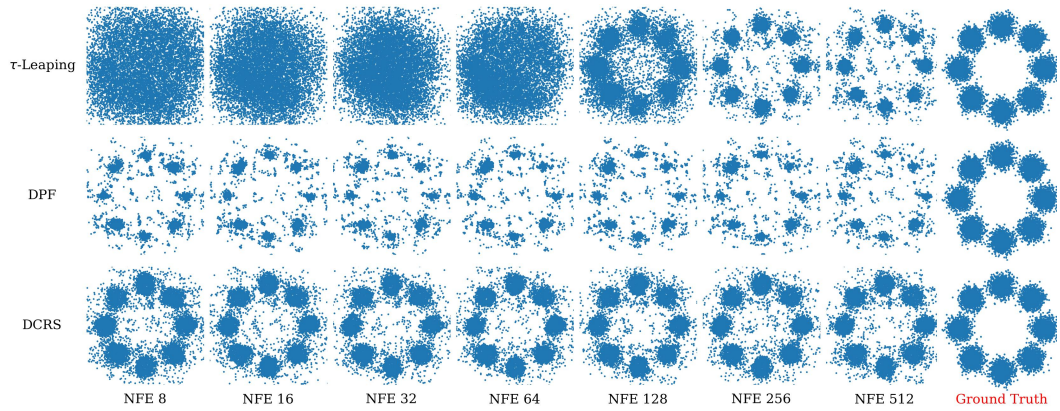


Figure 5: Plots of 10K generated samples as the number of discretization steps is increased for a toy mixture of Gaussians dataset in 8D from 2D experiments trained using a uniform corruption process. Whereas the τ-leaping sampler converges slowly to the modes, the DPF sampler reaches the modes immediately but distributes mass unevenly. The best balance between sampling speed and quality is achieved with DCRS.

D.4 Images

D.4.1 Experimental Setup

For the CIFAR10 and CelebA datasets, we use pretrained checkpoints provided by Campbell et al. [3] and Zhang et al. [14]. All experiments are conducted on 8 NVIDIA A6000 GPUs. To enable comparisons between sampling algorithms, each sample is generated starting from a unique fixed seed.

D.4.2 Effect of Multiple Restart Iterations

We provide results using a time discretization of 8 steps and a restart window of $[t_{\min,\max}] = [0.7, 0.8]$ with 3 discretization steps. All discretization schemes are found using Algorithm 1 with $\rho = 1$ and $\rho = 7$, which corresponds to taking uniformly spaced steps and steps that are clustered near $t = 0$, respectively. We present the effect on FID over 10000 generated samples as the number of restart iterations is changed. Our results in Figure 6 illustrate that the sampling quality tends to improve up until a certain number of iterations, after which the sample quality starts degrading due to the compounding of sampling errors. Furthermore, under non-uniform discretizations, such as when $\rho = 7$, the sample quality does not improve much with restart iterations, where multiple restart iterations tend to become harmful.

D.4.3 Failure of Higher-Order Solvers on DPF

We observe that applying a higher-order solver outside of the restart window tends to severely degrade quality, as shown in Figure 7. In particular, we observe that images tend to saturate in pixel intensities. We hypothesize that the added noise during the restart step tends to cancel out compounding errors under the DPF rate matrix, which are amplified under higher-order solvers.

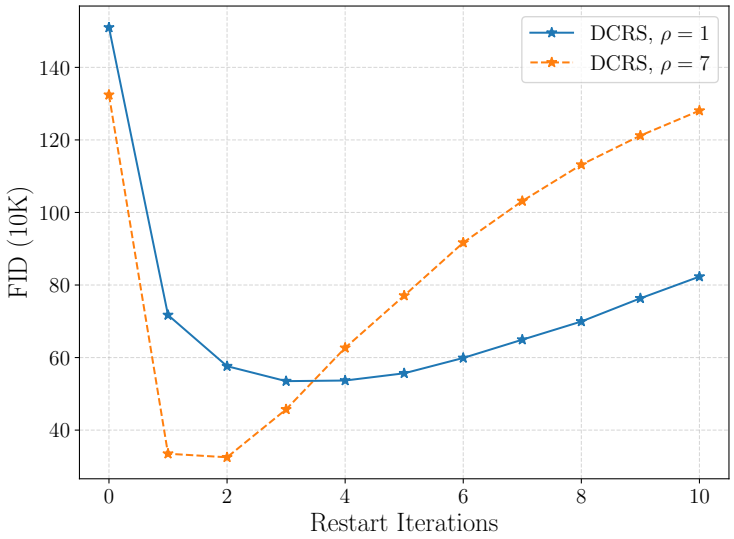


Figure 6: Effect of multiple restart iterations on sampling quality (FID) for a pre-trained model checkpoint trained on CIFAR10.

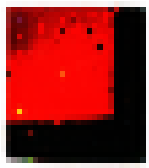


Figure 7: Example sample from running the trapezoidal solver (Algorithm 5) using the DPF rate matrix.

D.4.4 Restart Window Sensitivity

Motivated by our theoretical results that the stochasticity of the transition matrix affects the sensitivity to sampling error, we offer a heuristic that estimates the average log-magnitude in the rate matrix across time steps during sampling. In Table 6 we report empirical evidence that suggests that the optimal restart window we found, (0.7,0.8), is located approximately where the rate matrix is most sparse (0.87, 0.80), suggesting approximate agreement with our heuristic. We also include the rate matrix log-magnitudes for the default rate matrix, which do not experience the same dip in magnitude early on.

For the image domain experiments, we can find the optimal restart window via progressively refining a grid search, which we demonstrate results for in Table 7:

Table 6: Values at each sampling step for CIFAR-10 and CelebA, under stochastic and nearly deterministic sampling.

		CIFAR-10		CelebA	
Step	t	Stochastic	Nearly Det.	Stochastic	Nearly Det.
0	1.00	17.70	10.06	20.47	12.83
1	0.93	17.12	9.72	19.89	12.52
2	**0.87**	16.67	**9.59**	19.44	**12.52**
3	**0.80**	16.28	**9.91**	19.05	**12.79**
4	0.74	15.92	10.27	18.70	12.96
5	0.67	15.59	10.48	18.36	13.08
6	0.60	15.27	10.59	18.04	13.18
7	0.54	14.96	10.66	17.74	13.31
8	0.47	14.98	10.73	17.75	13.44
9	0.41	14.92	10.76	17.70	13.53
10	0.34	14.65	10.75	17.42	13.56
11	0.27	14.36	10.71	17.14	13.53
12	0.21	14.07	10.67	16.84	13.49
13	0.14	13.77	10.67	16.54	13.49
14	0.08	13.47	10.83	16.25	13.64

Table 7: CIFAR-10 FID by restart interval for $\rho = 1$ and $\rho = 7$, at three interval lengths.

Restart interval	$\rho = 1$ FID $\downarrow$	$\rho = 7$ FID $\downarrow$
Interval length 0.25		
0.00–0.25	259.89	339.12
0.25–0.50	37.31	77.68
0.50–0.75	**24.52**	**18.25**
0.75–1.00	34.65	45.66
Interval length 0.2		
0.00–0.20	261.85	356.41
0.20–0.40	45.72	82.68
0.40–0.60	23.75	38.49
0.60–0.80	**15.54**	**11.00**
0.80–1.00	38.79	45.66
Interval length 0.1		
0.00–0.10	249.24	359.36
0.10–0.20	45.91	87.68
0.20–0.30	56.63	76.75
0.30–0.40	50.44	70.56
0.40–0.50	44.26	51.54
0.50–0.60	28.20	31.31
0.60–0.70	22.91	23.88
0.70–0.80	**16.03**	**11.00**
0.80–0.90	31.08	27.12
0.90–1.00	46.84	81.68

D.4.5 Ablation Studies

For completeness, we provide additional results that illustrate the impact of higher-order solvers and discretization schemes applied to the τ-leaping sampler in Figure 8. Our results indicate non-uniform discretization schemes where $\rho = 7$ can improve performance slightly in the low-NFE regime, but ultimately result in worse performance in the high-NFE regime. Furthermore, the higher-order solver seems to be somewhat ineffective, which we attribute to the stochasticity of the default reverse rate matrix. Like the τ-leaping sampler, we expect higher-order solvers to converge faster on the DPF rate matrix, where sample trajectories exhibit fewer transitions. Ultimately, the results suggest that the discretization scheme and solver choices matter primarily when the stochasticity of the rate matrix is low enough.

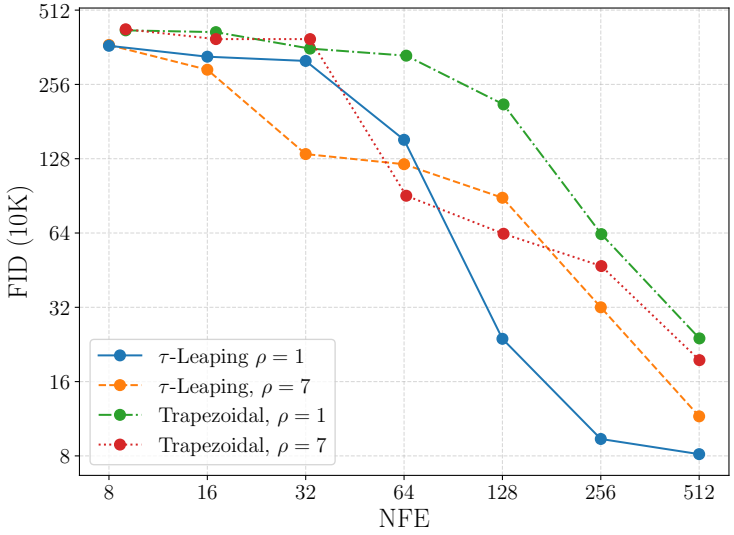


Figure 8: Effect of discretization scheme choices and higher-order solvers with the default τ-leaping sampler on sampling quality (FID) for a pre-trained model checkpoint trained on CIFAR10.

We also provide an additional predictor-corrector baseline, together with the DCRS algorithm and other sampling baselines for comparison for the CIFAR10 dataset. Specifically, we follow the experimental conditions in [3], and use 3 predictor-corrector iterations. We use the same time discretization as DCRS ($\rho = 7$) and also provide results for the uniform discretization ($\rho = 1$). We also perform further ablations using the DPF rate matrix or the default rate matrix.

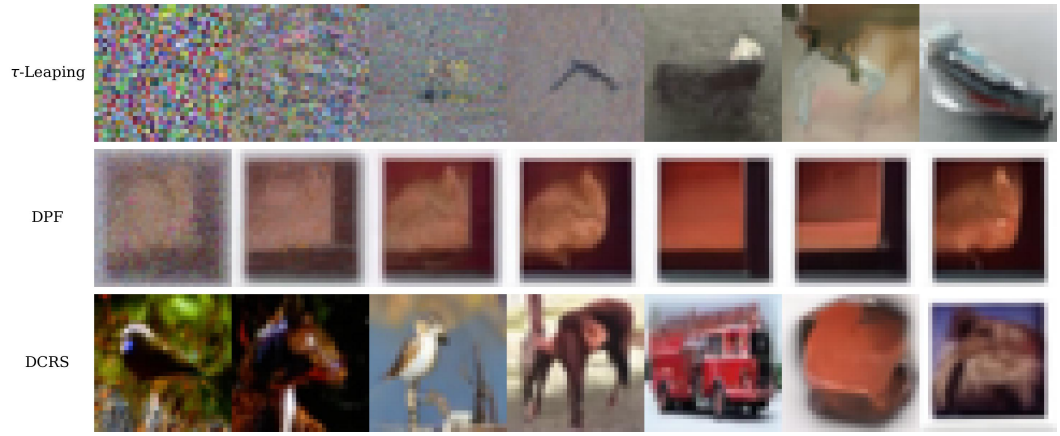
Table 8: Predictor-corrector baseline comparison.

Method	Total Steps (NFE)	FID ↓
DPF	8	150.99
DPF	16	51.66
DPF	32	33.66
DPF	64	39.43
DPF	128	43.29
DPF	256	45.22
DPF	512	46.47
τ-Leaping	8	366.74
τ-Leaping	16	330.90
τ-Leaping	32	316.35
τ-Leaping	64	151.22
τ-Leaping	128	21.73
τ-Leaping	256	7.11
τ-Leaping	512	6.09
DCRS	12	31.33
DCRS	26	10.80
DCRS	54	11.37
DCRS	110	14.01
DCRS	222	15.32
DCRS	446	16.69
DPF+PC ($\rho = 7$)	16	268.73
DPF+PC ($\rho = 7$)	36	62.13
DPF+PC ($\rho = 7$)	70	43.82
DPF+PC ($\rho = 7$)	144	35.09
DPF+PC ($\rho = 7$)	289	33.28
DPF+PC ($\rho = 7$)	576	33.73
DPF+PC ($\rho = 1$)	10	428.48
DPF+PC ($\rho = 1$)	21	130.26
DPF+PC ($\rho = 1$)	40	41.49
DPF+PC ($\rho = 1$)	81	23.58
DPF+PC ($\rho = 1$)	163	28.43
DPF+PC ($\rho = 1$)	327	31.45
DPF+PC ($\rho = 1$)	652	33.30
τ-Leaping+PC	10	410.36
τ-Leaping+PC	21	322.03
τ-Leaping+PC	40	292.98
τ-Leaping+PC	81	134.31
τ-Leaping+PC	163	20.23
τ-Leaping+PC	327	7.11
τ-Leaping+PC	652	6.06

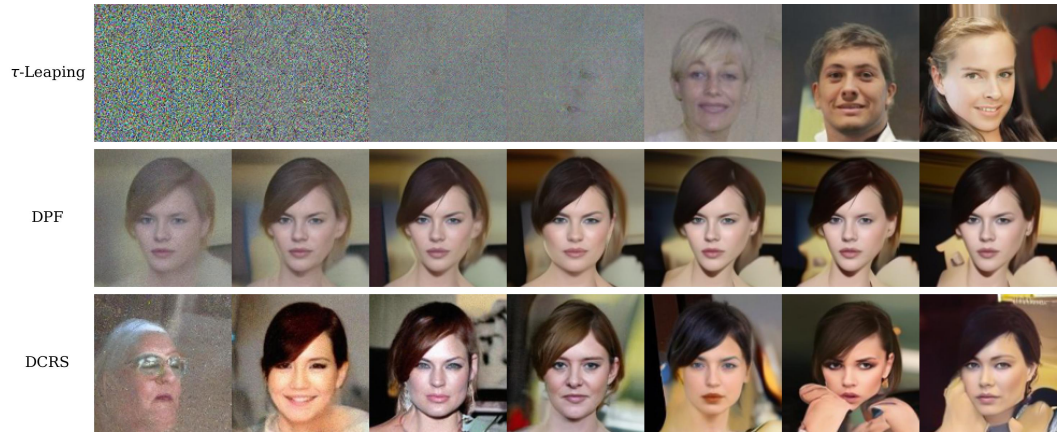
Predictor-corrector sampling consistently under performs DCRS except at high NFE. It improves quality at high NFE but performs poorly at low NFE, which can be attributed to elevated discretization error from larger rates.

D.4.6 Sample Visualization

We illustrate the difference in convergence rates for each dataset and choice of sampler in Figures 9a and 9b below. For each sampler, we use the samples generated by the same fixed seed using different total NFE.



(a) Generated samples using increasing NFE from left to right from a fixed seed for a pre-trained CIFAR10 checkpoint.



(b) Generated samples using increasing NFE from left to right from a fixed seed for a pre-trained CelebA checkpoint.

Figure 9: Sample comparisons between the vanilla τ-leaping, DPF, and DCRS sampler for large-scale image datasets.

D.5 Language

Method	Generative PPL. (↓)				Entropy (↑)			
	T=16	*T=32*	*T=64*	*T=128*	*T=16*	*T=32*	*T=64*	*T=128*
D3PM	256.42	207.75	193.91	186.79	6.806	6.849	6.865	6.886
DDIM†	253.60	213.57	205.21	190.66	6.831	6.859	6.891	6.898
Euler	272.97	225.41	197.87	190.10	6.921	6.922	6.890	6.902
Euler DPF†	257.62	215.07	204.87	193.30	6.843	6.850	6.890	6.901
	T=12	*T=26*	*T=54*	*T=110*	*T=12*	*T=26*	*T=54*	*T=110*
Euler DCRS	298.44	245.06	331.60	333.73	6.704	6.823	7.080	7.115

Table 9: Generative perplexity and entropy evaluated for a uniform diffusion language model (UDLM) trained on LM1B over 1024 sequences of length 128. † indicates a near-deterministic sampler that samples from the DPF CTMC.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction outline the theoretical contribution of the paper, as well as the experimental findings across large scale dataset benchmarks.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A discussion about limitations is included at the end near the conclusion.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: All theoretical results state the assumptions about the underlying diffusion model and forward process used. Proofs are provided in the appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The experiments section outlines the exact dataset and metrics used to evaluate the methods of the paper, and additional details about the experimental setup can be found in the appendix.

   Guidelines:

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

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

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

Answer: [No]

Justification: Code and data will be released upon acceptance.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The experimental setup is stated where appropriate, and additional details are left to the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The sample size used to take average statistics of all experimental metrics is reported in the experimental details. For large scale experiments that are relevant to the main claims of the paper, error bars are left out as they would be too computationally expensive.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The appendix outlines the type of GPUs used to run all experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The paper does not present any ethical violations.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: A discussion regarding broader impacts is left at in appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The focus of the work is in accelerating model inference, and hence does not involve releasing data or models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All pre-trained model checkpoints are credited to the original paper authors from which the checkpoint was taken from.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not introduce any new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not crowdsource or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not crowdsource or research with human subjects.

    Guidelines:

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

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLM usage was primarily for writing and editing, and the core methods of the paper are not influenced.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.