EQUILIBRIUM DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models generate text through iterative, parallel token updates. When applying these models to reasoning tasks, we often seek any correct solution, rather than reproduce the distribution of answers in the training data. We introduce *Equilibrium Diffusion Language Models*, which approach generation as a process of finding and preserving valid solutions. We train the model with a single cross-entropy objective to reconstruct valid solutions from corrupted inputs. Our theoretical analysis identifies conditions on the corruption process under which valid solutions become absorbing states of the population-optimal model and repeated sampling reaches them almost surely, with exponentially fast convergence. The analysis also establishes a probability gap between preserving valid and invalid states, suggesting a criterion for stopping generation. We evaluate the approach on unconditional generation of valid Sudoku grids and program synthesis with training on TinyGSM and evaluation on GSM8K. On GSM8K, our model achieves 63.6% accuracy, comparable to 63.3% for the autoregressive greedy baseline.

1 INTRODUCTION

Diffusion language models (DLMs) generate text through iterative, parallel token updates (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024; Schiff et al., 2025; Nie et al., 2025). Updating multiple tokens per model call offers a promising route to efficient generation. Many generation tasks admit multiple valid outputs, any one of which can satisfy the same input. The goal is then to produce a high-quality output, rather than to reproduce a particular probability distribution over acceptable outputs. This motivates a generation process focused on finding valid outputs without prescribing their relative frequencies (Figure 1).

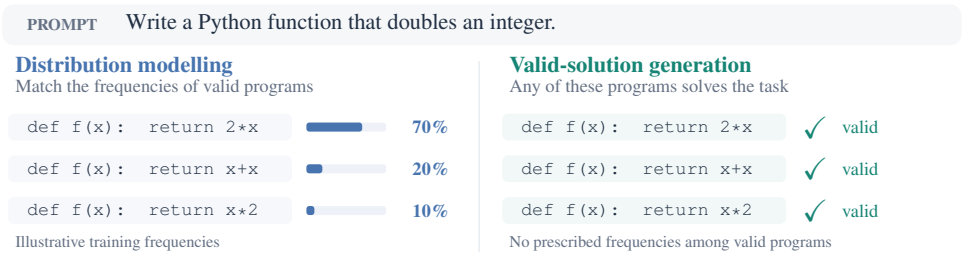

Figure 1: Distribution modelling matches solution frequencies, whereas valid-solution generation accepts any correct program. This distinction concerns the generation objective, not a particular sampling algorithm. Frequencies are illustrative.

Let c be a problem description and denote its valid solutions by

$$\mathcal{S}(c) = \{x : x \text{ correctly solves } c\}.$$

Our goal is to generate an answer $\widehat{X} \in \mathcal{S}(c)$ with high probability. Any distribution supported on $\mathcal{S}(c)$ achieves this goal, regardless of the frequencies of individual solutions.

We seek a process that continues revising invalid candidates and preserves a valid solution once it is found. It could retain solutions by recognizing them and stopping, or, more strongly, through updates

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

that leave them unchanged. We investigate whether one model can learn both to generate solutions and to recognize when to stop, without a separate verifier at inference.

We introduce *Equilibrium Diffusion Language Models*, which learn to find and preserve valid solutions through a single denoising cross-entropy objective. The same prediction updates a candidate and gives its probability of remaining unchanged, which we call the *stay probability*.

Our contribution is to establish when denoising cross-entropy enables validity recognition from the model's own predictions and supports convergent generation. At the population optimum, an invalid candidate cannot have a high probability of reproducing itself, which provides a criterion for stopping generation. Under suitable conditions, repeated updates reach a recognized valid solution with exponentially decaying failure probability. When corruption excludes transitions between distinct valid solutions, every valid solution becomes absorbing and remains unchanged without stopping.

Our main contributions are as follows:

1. We introduce Equilibrium Diffusion Language Models, which learn to find and preserve valid solutions by repeatedly applying one denoiser trained with a single cross-entropy objective (Section 3.1).
2. We prove that, at the population optimum, the stay probability supports validity recognition under arbitrary corruption with full input coverage. We establish exponential convergence with stopping under a uniform reachability condition, and exact absorption when corruption excludes transitions between distinct valid solutions (Sections 3.2–3.3, Theorems 1–3).
3. We empirically examine how these guarantees translate to learned models and demonstrate near-complete Sudoku validity with adaptive stopping and GSM8K accuracy comparable to the autoregressive (AR) greedy baseline (Sections 4.1–4.3).

2 Background

We review how discrete diffusion models are trained and used for generation. This establishes the notation and distinguishes revising a candidate from recognizing and retaining a valid solution.

Discrete diffusion. Discrete diffusion models learn to reverse gradual corruption (Austin et al., 2021; Sahoo et al., 2024; Schiff et al., 2025). Let $x \sim p_{\text{data}}$ be a sequence in $\mathcal{V}^L$, where $\mathcal{V}$ is a finite vocabulary and L is the sequence length. The forward process corrupts sequences at $t \in [0, 1]$:

$$z_t \sim q_t(\cdot \mid x), \qquad z_0 = x, \qquad q_1(\cdot \mid x) = \pi. \tag{1}$$

The noise prior is π. Typical corruption masks tokens or replaces them uniformly.

Reverse transitions and training. A neural network parameterizes transitions from time t to an earlier time $s < t$. Discrete diffusion models commonly update tokens in parallel by predicting a categorical distribution at each position and sampling independently across positions, conditional on the entire current sequence. The resulting transition factorizes as

$$p_\theta(z_s \mid z_t; s, t) = \prod_{i=1}^{L} p_{\theta,i}(z_{s,i} \mid z_t; s, t). \tag{2}$$

Different models convert network outputs into reverse-transition probabilities differently. For example, UDLM inserts the network output into an analytic reverse-posterior formula (Schiff et al., 2025).

Training maximizes a variational lower bound on the log-likelihood of clean data, averaged over $x \sim p_{\text{data}}$. The resulting loss depends on the corruption process and parameterization. For masked diffusion, predicting clean-token probabilities gives weighted cross-entropy on masked tokens in the continuous-time limit (Sahoo et al., 2024; Shi et al., 2024). Uniform-state diffusion uses a different variational loss, which includes contributions from both changed and unchanged token positions (Schiff et al., 2025; Deschenaux et al., 2026).

Generation. Sampling starts from the noise prior π and follows reverse transitions on a decreasing time grid $1 = t_K > \cdots > t_0 = 0$. These time-dependent transitions are trained to approximate the data distribution through parallel token updates. A time-dependent reverse process can nevertheless use a time-independent network, as in RADD (Ou et al., 2025).

Revising and preserving solutions. Whether the transitions can revise earlier predictions depends on the process and sampler. In conventional masked sampling, revealed tokens remain fixed, so an incorrect token cannot be revised. Uniform-state diffusion and remasking allow revisions (Wang et al., 2025). Predictor–corrector samplers also insert corrective updates into the reverse process (Campbell et al., 2022). However, the ability to revise a candidate does not by itself provide a criterion for recognizing a valid solution or ensure that subsequent updates preserve it.

We instead study a time-independent update that can be applied repeatedly, using the same prediction to revise a candidate and decide when to stop. This allows the stopping decision to depend on the current candidate rather than on reaching the end of a predefined diffusion schedule.

3 Equilibrium Diffusion Language Models (EQDLM)

We train one denoiser whose predictions guide candidate updates and determine when to stop.

We first define the denoising objective and generation process (Section 3.1). We then establish how the factorized population optimum recognizes valid solutions, reaches them with stopping, and preserves them under a stronger corruption condition (Section 3.2). Finally, we identify valid solutions that must pass the stopping test under uniform-token corruption (Section 3.3).

3.1 Learning and generation

Learning from corrupted solutions. For each problem c, candidate solutions belong to the finite sequence space $\mathcal{X} = \mathcal{V}^L$, with $L \geq 1$, and valid solutions form a subset $\mathcal{S}(c) \subseteq \mathcal{X}$. To learn candidate updates from valid solutions alone, we corrupt a valid training solution and train the model to recover its source. Uppercase letters denote random variables and lowercase letters their realizations. We draw $X \sim p_{\text{data}}(\cdot \mid C)$ and then $Z \sim q(\cdot \mid X, C)$. Corruption can leave the source unchanged, produce an invalid candidate, or produce a different valid solution (Figure 2). A time-independent denoiser predicts source tokens independently given the full candidate and problem:

$$p_\theta(x \mid z, c) = \prod_{i=1}^{L} p_{\theta,i}(x_i \mid z, c), \tag{3}$$

We train with a single denoising cross-entropy (CE) objective:

$$\mathcal{L}_{\text{eq}}(\theta) = \mathbb{E}_{C,X}\, \mathbb{E}_{Z\sim q(\cdot|X,C)} \left[-\sum_{i=1}^{L} \log p_{\theta,i}(X_i \mid Z, C) \right]. \tag{4}$$

Only the target X must be known to be valid. No validity label is needed for Z.

Appendix C provides training pseudocode and discusses optional weighting of the objective.

Updating and stopping. We initialize a candidate from a uniform or task-specific prior. At each iteration, we evaluate the stay probability $p_\theta(Z_k \mid Z_k, c)$ by multiplying the model's untempered token probabilities. We return the candidate and terminate generation if

$$p_\theta(Z_k \mid Z_k, c) > \tau. \tag{5}$$

Otherwise, a sampler proposes the next candidate. The proposal rule may vary with time and history, while the stopping test always uses the model's untempered probabilities. Categorical sampling draws tokens independently from the predicted marginals:

$$Z_{k+1} \sim p_\theta(\cdot \mid Z_k, c). \tag{6}$$

For this sampler, recognition and generation use the **same model call**. Condition tokens stay fixed, so the product in (3) includes only generated positions. A finite call budget may terminate generation without acceptance. The validity guarantee applies only to threshold-accepted candidates.

The analysis below motivates $\tau = e^{-1}$ and states the conditions under which stopping finds a valid solution. For learned models, we also evaluate other fixed thresholds. Appendix A gives the sampling algorithms and details of scoring and temperature scaling.

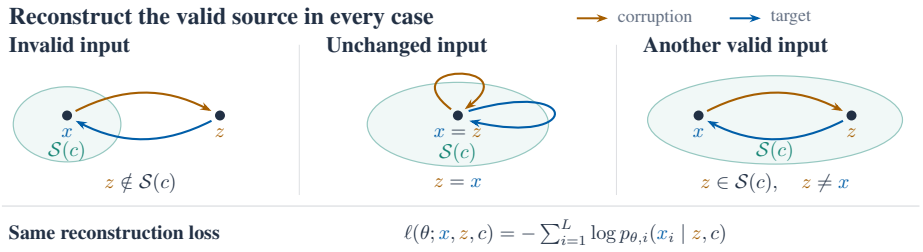

Figure 2: Denoising training for **EqDLM (ours)**. Every target is the valid source, including when corruption produces another valid solution. Theorem 1 covers all three cases. Excluding transitions to a *different* valid solution, shown in the rightmost panel, yields exact absorption (Theorem 3).

3.2 Recognition, Convergence, and Equilibrium

We first characterize what the denoising objective guarantees at its population optimum. Let $p^\star$ minimize (4) over all factorized categorical distributions, including probabilities zero and one. All statements fix a problem c and assume that the training population has support $\mathcal{S}(c)$. Every candidate must also have positive probability under corruption:

$$\bar{q}_c(z) = \sum_{x \in \mathcal{S}(c)} p_{\text{data}}(x \mid c) q(z \mid x, c) > 0 \quad \text{for every } z \in \mathcal{X}. \tag{7}$$

Coverage makes the objective constrain the model at every input, while allowing corruption between distinct valid solutions. This coverage condition is weaker than requiring $q(z \mid x, c) > 0$ for every source and input, which we invoke explicitly where needed. Proofs are in Appendix B.

The results establish four complementary properties:

1. **Recognition.** The threshold rejects every invalid candidate (Theorem 1).
2. **Why stopping matters.** Without it, even optimal categorical updates can retain invalid mass (Proposition 1).
3. **Convergence.** Uniform access to a nonempty accepted set gives exponentially decaying failure probability (Theorem 2).
4. **Absorption.** Excluding corruption between distinct valid solutions makes every solution absorbing (Theorem 3).

Recognizing valid solutions. Let $\pi_z(x) = \Pr(X = x \mid Z = z, C = c)$ be the posterior over valid source solutions. A draw from this exact joint distribution is always valid. The factorized model minimizing population denoising CE instead uses its token marginals, $p_i^\star(v \mid z, c) = \sum_x \pi_z(x) \mathbf{1}\{x_i = v\}$, whose independent draws can form an invalid combination. Our analysis addresses this factorized setting. Every valid source differs from an invalid candidate in at least one position. Thus, under the clean posterior, the expected number of mismatched tokens is at least one. This limits the product of the matching-token probabilities, yielding the bound below.

Theorem 1 (Validity recognition under arbitrary corruption)**.** *Under the population and coverage assumptions above, every invalid input satisfies*

$$z \notin \mathcal{S}(c) \quad \Longrightarrow \quad p^\star(z \mid z, c) \leq (1 - 1/L)^L < e^{-1}. \tag{8}$$

Consequently, the rule $p^\star(z \mid z, c) > e^{-1}$ accepts only valid solutions, regardless of how a candidate is proposed. This conclusion allows corruption between distinct valid solutions.

Some valid solutions may still fail the stopping test. Appendix B.1 proves the bound from the posterior marginals learned by token cross-entropy.

Why stopping matters. Validity recognition does not imply that ordinary updates eventually stay within the solution set when the stopping rule is disabled. For example, $\mathcal{S} = \{00, 11\}$ permits each token to be either 0 or 1, but independent choices also produce the invalid strings 01 and 10. Let $\mathcal{S}_i(c)$ be the tokens appearing in position i of valid solutions and write $\mathcal{C}(c) = \prod_{i=1}^{L} \mathcal{S}_i(c)$ for their coordinatewise combinations. Constraints between tokens can make $\mathcal{S}(c) \subsetneq \mathcal{C}(c)$.

Proposition 1 (Invalid mass without stopping). *If $q(z \mid x, c) > 0$ for every $z \in \mathcal{X}$, $x \in \mathcal{S}(c)$, and $\mathcal{S}(c) \subsetneq \mathcal{C}(c)$, categorical sampling from $p^\star$ without stopping converges, from every initialization, to a unique stationary distribution supported on $\mathcal{C}(c)$. This distribution assigns strictly positive probability to invalid states.*

From recognition to generation. The population-optimal model tests acceptance, while an arbitrary sampler proposes updates. Define the accepted set and its complement:

$$\mathcal{A} = \{z : p^\star(z \mid z, c) > e^{-1}\}, \qquad \mathcal{B} = \mathcal{X} \setminus \mathcal{A}.$$

Theorem 1 gives $\mathcal{A} \subseteq \mathcal{S}(c)$. Termination also requires a nonempty $\mathcal{A}$ and a sampler whose probability of entering $\mathcal{A}$ at each step before stopping is bounded below by a fixed positive constant.

Theorem 2 (Exponential convergence with threshold stopping). *Under the assumptions of Theorem 1, let $\mathcal{A} \neq \varnothing$ and let $T_{\mathcal{A}}$ be its first visit time. Suppose a sampler, possibly depending on time and history, satisfies*

$$\Pr(Z_{k+1} \in \mathcal{A} \mid Z_0, \ldots, Z_k) \geq \varepsilon > 0$$

for every history before stopping, with the same ε at every step. Then, for every initial distribution μ_0 and integer $k \geq 0$,

$$\Pr_{\mu_0}(T_{\mathcal{A}} > k) \leq (1 - \varepsilon)^k \leq e^{-\varepsilon k}. \tag{9}$$

Stopping at $T_{\mathcal{A}}$ returns a valid solution almost surely.

The model determines acceptance, while the sampler controls the hitting time. A common lower bound $\eta > 0$ on every state's proposal probability suffices, with $\varepsilon = |\mathcal{A}|\eta$. If accessibility holds only after N initial steps, then $\Pr(T_{\mathcal{A}} > N + k) \leq (1 - \varepsilon)^k$. Appendix B.3 proves this extension and accounts for accepted initial states.

Exact equilibrium as a stronger case. The stopping rule preserves an accepted candidate by terminating generation even if further updates could change it. To ensure that model updates preserve every valid solution, we additionally assume that corruption cannot map one valid solution to another, excluding the rightmost outcome in Figure 2. Self-transitions remain allowed:

$$q(z \mid x, c) = 0 \quad \text{for distinct } x, z \in \mathcal{S}(c). \tag{10}$$

Then a valid input has only itself as a possible denoising target.

Theorem 3 (Exact absorption and complete recognition). *Suppose the assumptions of Theorem 1 hold and corruption satisfies* (10)*. Then*

$$p^\star(z \mid z, c) = 1 \quad \Longleftrightarrow \quad z \in \mathcal{S}(c). \tag{11}$$

Hence $\mathcal{A} = \mathcal{S}(c)$: threshold stopping accepts every valid solution and rejects every invalid input. Categorical sampling reaches $\mathcal{S}(c)$ almost surely with an exponentially decaying hitting-time tail, and thereafter leaves the reached solution unchanged even without stopping.

Coverage (7) suffices without strict positivity of q: each input has a valid source whose tokens have positive posterior marginals, so categorical sampling can draw it. Finiteness gives a uniform positive hitting probability. Every valid solution is an equilibrium, $\delta_z p^\star = \delta_z$.

3.3 Which valid solutions are recognized?

Theorem 2 requires at least one valid solution to pass the stopping test. When corruption can map one valid solution to another, this is not automatic. For uniform-token corruption, we give a sufficient condition on a solution's population probability for acceptance, ensuring a nonempty accepted set.

Uniform corruption. Uniform replacement is a simple general corruption that does not require checking whether a proposed input is valid. For $t \sim \mathcal{U}[0, 1]$, replace each token independently with probability t, drawing the replacement uniformly from $V = |\mathcal{V}|$ tokens:

$$q_t(z \mid x) = \prod_{i=1}^{L} \left[(1 - t)\mathbf{1}\{z_i = x_i\} + t/V\right], \qquad q(z \mid x) = \int_0^1 q_t(z \mid x)\, dt. \tag{12}$$

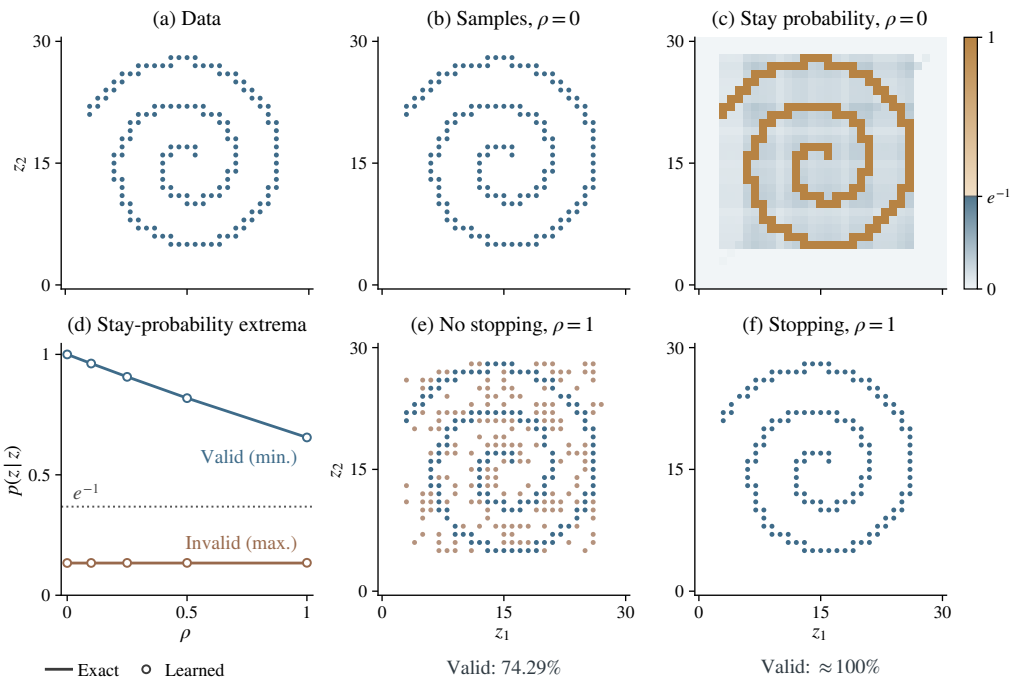

Figure 3: Generation and recognition on the discrete Swiss roll. **Top:** data, samples, and learned stay probabilities at $\rho = 0$. **Bottom:** (d) minimum valid and maximum invalid stay probabilities versus ρ for optimal (lines) and learned (circles) models, (e,f) learned generation at $\rho = 1$ without and with e^{-1} stopping. Blue/brown points denote valid/invalid states. Panel (c) uses a stay-probability color scale with boundary e^{-1}. Panels (b,e,f) show unique samples within 64 updates from invalid states. Validity is computed from full distributions.

The same valid input z can then be paired with different targets. Corruption of z itself teaches the model to preserve z, whereas corruption of another solution into z teaches it to recover that other solution. The next bound gives a sufficient condition for z to pass the threshold.

Proposition 2 (Recognition under uniform corruption)**.** *For this corruption process, let $L \geq 1$ and $V = |\mathcal{V}| \geq 5$. At the population optimum, every solution x satisfying*

$$p_{\text{data}}(x \mid c) > \delta_{L,V}, \qquad \delta_{L,V} = \frac{1}{1 + L(V-1)(1 - V^{-(L+1)})} \tag{13}$$

has $p^\star(x \mid x, c) > e^{-1}$ and is therefore accepted. The cutoff $\delta_{L,V}$ is of order $1/(LV)$.

One solution above the cutoff makes the accepted set nonempty. Strictly positive uniform corruption then ensures categorical sampling reaches it (Theorem 2, accessibility proof in Appendix B.3). This guarantees acceptance, not absorption, and Theorem 1 still rejects all invalid candidates. Appendix B.5 gives the retention bound, accepted-mass guarantee, and extension to weights $(1-t)^p$, $p \geq 0$.

Further connections to denoising and adaptive stopping methods are discussed in Appendix H.

4 Experiments

We test the theory across three levels of complexity. The discrete Swiss roll allows exhaustive checks over a finite state space (Section 4.1), Sudoku tests longer structured solutions (Section 4.2), and TinyGSM tests conditional program generation (Section 4.3). Across these settings, we examine how closely learned models reproduce validity recognition, convergence, and preservation, and whether these properties support effective generation. We measure sampling cost by the number of model function evaluations (NFE). Settings and diagnostics are in Appendices G, D, and F, respectively.

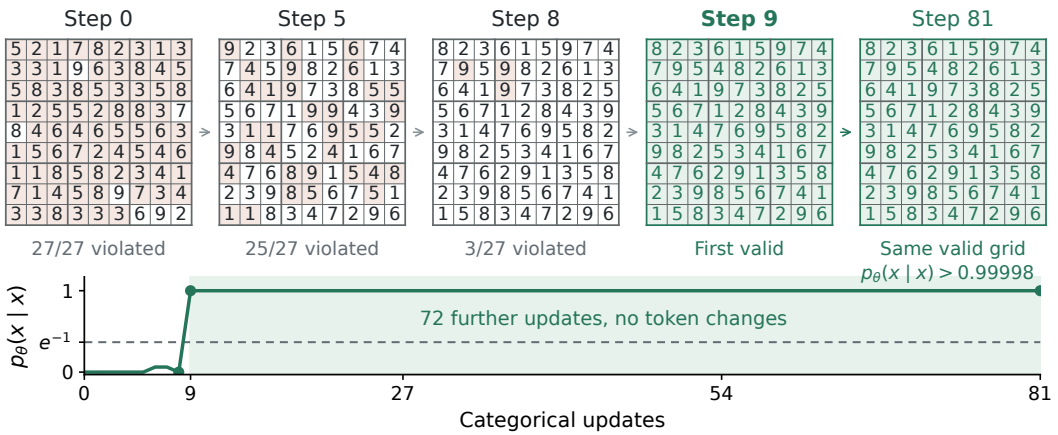

Figure 4: Categorical generation with **EqDLM (ours)**, without stopping. Red cells contain repeated digits; counts give violated row, column, and box constraints. Green marks valid grids. From update 9 through 81, the solution stays unchanged with stay probability above 0.99998. Dashed line: e^{-1}.

4.1 Discrete Swiss roll

We use a discrete Swiss roll to distinguish validity recognition from absorption by allowing corruption between valid points.

Data and corruption. Each point is a pair of tokens in $\mathcal{X} = \{0, \ldots, 30\}^2$. The data distribution is uniform on a spiral $\mathcal{S}$ of 139 points (Figure 3a). We use uniform-token corruption (12). When a corrupted sample lands on a different valid point, we retain it with probability ρ and otherwise restore the original point. Thus, $\rho = 0$ prevents transitions between valid points, while $\rho = 1$ leaves uniform corruption unchanged. This construction uses the known support, with q_ρ defined in (37).

Training and sampling. For each $\rho \in \{0, 0.1, 0.25, 0.5, 1\}$, we train the factorized denoiser with the exact population CE in (4) under q_ρ and compare it with the optimum computed from exact posterior marginals. Generation starts uniformly outside $\mathcal{S}$ and uses 64 categorical updates at temperature one. Appendix G gives architecture, training, and evaluation details.

At $\rho = 0$, the learned model separates all valid and invalid inputs at e^{-1} and places nearly all generated mass on the support (Figure 3a–c). Valid states are approximately absorbing. Appendix G reports valid-state retention and invalid-state stay probabilities.

Allowing transitions between valid points. Changing ρ leaves corruption probabilities at invalid inputs unchanged, so their clean posterior and population-optimal denoiser also remain unchanged. On valid inputs, competing clean targets can reduce retention. Figure 3d confirms this distinction: retention decreases as ρ grows, yet valid and invalid stay probabilities remain separated by e^{-1} throughout the sweep. The learned denoisers closely follow the population prediction.

This separation does not make unrestricted generation valid. At $\rho = 1$, both the optimal and learned models retain substantial invalid mass after 64 updates, consistent with Proposition 1. Applying the same e^{-1} stopping threshold restores near-complete validity across the sweep (Figure 3e,f and stopping protocol in Appendix G).

4.2 Unconditional Sudoku

The discrete Swiss roll illustrates recognition and preservation in a small state space. We next test these properties in unconditional Sudoku generation, where validity couples row, column, and box constraints.

Setup. We generate completed 81-cell grids without clues using the backtracking generator of Agarwal et al. (2026), with 9.99 million training grids and 10,000 validation grids. The denoiser uses the unweighted CE objective (4) and uniform-token corruption (12). We omit self-conditioning, which feeds information from a preceding model prediction into the next model call (Chen et al., 2023). Appendix D gives the architecture and training settings.

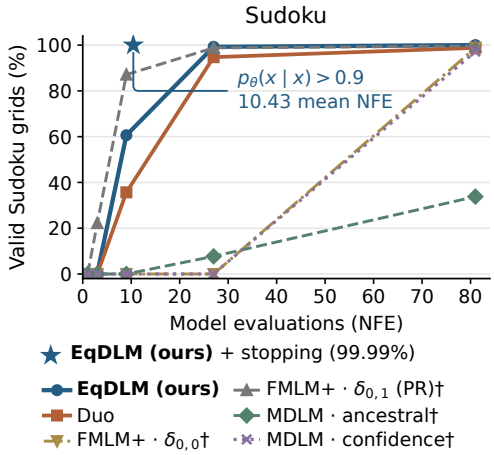

Figure 5: Sudoku generation quality. Fixed budgets use 10,000 samples. †: MDLM and FMLM+ results from Agarwal et al. (2026), Table 6. PR: posterior refinement. The star denotes EqDLM stopping at $p_\theta(x \mid x) > 0.9$: 99.99% validity at 10.43 mean NFE on 10,000 separate trajectories. Appendix D gives sampling and stopping details.

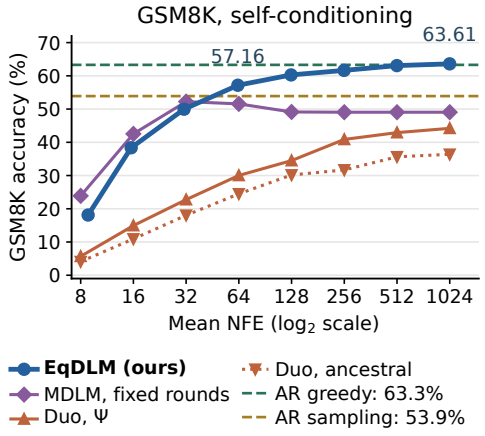

Figure 6: GSM8K with self-conditioning (1,319 problems). EqDLM uses annealed categorical sampling with stopping, counting all model calls. Baseline budgets are fixed; MDLM uses fixed rounds. Dashed lines: published AR accuracies. Appendix F gives evaluation details.

Generation. From independent uniform digits, we apply categorical sampling (6), stopping according to (5) or returning the grid produced by the final update at the 81-call cap. Figure 4 shows a trajectory reaching a valid grid and preserving it through subsequent updates.

Among the evaluated methods, **EqDLM (ours)** with threshold stopping reaches near-perfect Sudoku validity with the fewest model evaluations (Figure 5). At fixed budgets, it also leads the evaluated diffusion methods (fixed-budget results in Appendix Table 3). FMLM+ with posterior refinement (PR) instead uses learned flow maps (Agarwal et al., 2026). Diffusion models can also be accelerated through distillation, which we do not use (Deschenaux & Gulcehre, 2025; Li et al., 2026).

Recognition. The learned stay probability supports reliable stopping at a higher threshold: $\tau = 0.9$ accepts no invalid grid in this evaluation, whereas e^{-1} produces false stops on 2.76% of trajectories (stopping diagnostics in Appendix Table 4). This discrepancy reflects imperfect learning, since Theorem 1 already permits corruption between valid grids.

Retention and diversity. Valid grids have mean stay probability above 0.99998 (retention measurements in Appendix D). Stopping prevents rare departures from invalidating the output. Another run yields valid grids, distinct even after digit relabeling and absent from training.

4.3 TinyGSM

Sudoku demonstrates reliable recognition and near-absorbing behavior in a constrained domain. We next study program generation, where the model must learn validity across diverse problems and solutions. We train on TinyGSM (Liu et al., 2023) and evaluate on all 1,319 GSM8K test problems (Cobbe et al., 2021), checking executed Python programs against reference answers.

Setup. We train with weighted denoising CE, $w(t) = 1 - t$, and uniform-token corruption (12). We evaluate separately trained EqDLM variants with and without self-conditioning (SC), alongside masked diffusion MDLM (Sahoo et al., 2024) and uniform-state diffusion Duo (Sahoo et al., 2025) in both settings. For SC, we feed hidden activations from the preceding call into the next, following LDDM (Jo et al., 2025). Training, SC, and baseline sampling settings are in Appendix F.

Annealed categorical sampling. Direct categorical sampling at $T = 1$ quickly stabilizes at incorrect programs and gives near-zero accuracy in the tested SC and no-SC runs (categorical sampling results in Appendix F). We hypothesize a mismatch between corrupted training inputs and model-generated

Table 1: GSM8K accuracy (%) on 1,319 problems. AR references: sampling 53.9%, greedy 63.3%. *Mean total NFE including continuation and stopping checks. †: AR/CANDI/FLM/S-FLM from Deschenaux & Gulcehre (2026), Table 2 (full S-FLM, top-1 decoding), and FMLM+ from Agarwal et al. (2026), Table 8. PR: posterior refinement. Full curves and protocols: Appendix F.

	Without self-conditioning				With self-conditioning			
	≈ 64 calls		≈ 1024 calls		≈ 64 calls		≈ 1024 calls	
Method	NFE	Acc.	NFE	Acc.	NFE	Acc.	NFE	Acc.
MDLM, fixed rounds	64	**40.64**	1024	36.92	64	51.55	1024	49.05
Duo, ancestral	64	15.47	1024	19.64	64	24.49	1024	36.39
Duo, Ψ	64	19.26	1024	26.54	64	30.02	1024	44.20
EqDLM (ours)	63.32*	33.13	1023.08*	**45.64**	63.17*	**57.16**	1023.20*	**63.61**

Published baselines without self-conditioning

Method	NFE	Acc. (%)
CANDI†	1024	0.2
FLM†	1024	0.3
S-FLM†	1024	18.0

Method	NFE	Acc. (%)
FMLM+, PR (scratch)†	64	19.1
FMLM+, PR (MDLM distill)†	64	23.4
FMLM+, PR (MDLM init)†	64	33.6

states. To reduce it, we apply the training corruption to predictions at decreasing noise levels t_k:

$$Z_{k+1,i} \overset{\text{ind}}{\sim} \alpha_{t_k} p^{(T)}_{\theta,i}(\cdot \mid Z_k, c) + (1 - \alpha_{t_k})\text{Unif}(\mathcal{V}), \tag{14}$$

where $1 - \alpha_{t_k}$ is the token replacement probability. Over N updates, we gradually decrease t_k according to the noise schedule in Appendix A. We use greedy predictions ($T = 0$), with stochasticity from token replacement. The last annealing update adds no noise. We then continue with ordinary categorical sampling ($T = 1$).

Before every transition in both phases, we compare the current state's untempered probability of reproducing itself with e^{-1} and return the first accepted state. For a population-optimal score and uniformly accessible accepted states, Theorem 2 guarantees exponential convergence of the categorical continuation when run without a call limit. Mean NFE counts both phases and recognition. Algorithm 3 gives SC propagation and the capped return policy, with at most 128 continuation calls. Execution is used only for evaluation.

Generation stops shortly after annealing: with at least 62 annealing steps, the categorical phase adds only 1.08–1.32 mean model calls, including the stopping check (stopping statistics in Appendix F).

Generation quality. With SC, EqDLM exceeds published AR sampling accuracy near 64 model calls and is comparable to published AR greedy near 1,024 calls (Table 1). Figure 6 shows further gains with more calls and the highest accuracy among tested discrete diffusion models with SC at larger budgets. Without SC, EqDLM trails MDLM near 64 calls but overtakes it near 1,024.

Recognition. Some incorrect programs pass the stopping threshold (stopping precision in Appendix F). Since Theorem 1 permits corruption between valid solutions, these violations of the population-optimal bound reflect imperfect learning of the recognition signal.

Multiple attempts under a shared budget. Across the tested total NFE budgets, EqDLM finds at least one correct answer more often than either AR decoder (budget-matching protocol and pass@k results in Appendix F).

5 Conclusion

Equilibrium Diffusion Language Models connect denoising cross-entropy to finding and preserving valid solutions. At the population optimum, stay probabilities enable validity recognition, and threshold stopping yields exponential convergence under the stated reachability conditions. Stronger conditions on corruption make valid solutions absorbing without stopping. Experiments closely reproduce these properties on Swiss roll and Sudoku, while strong program-generation results coexist with imperfect recognition. These findings establish a theoretical foundation for valid-solution generation and demonstrate its practical promise.

AI USE STATEMENT

AI tools assisted with literature search, manuscript drafting and editing, and the development of mathematical claims and proofs. For research execution, they assisted with code implementation and computational checks. All results were verified by the authors, who take responsibility for the claims, results, and final manuscript.

REFERENCES

Manan Agarwal, Sheel Shah, Chanhyuk Lee, Jaehoon Yoo, Jerry Huang, Seunghoon Hong, Aditi Raghunathan, Jinwoo Kim, and Nicholas M. Boffi. Posterior refinement: Fast language generation via any-order flow maps. *arXiv preprint arXiv:2606.24773*, 2026. URL https://arxiv.org/abs/2606.24773.

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*, volume 34, pp. 17981–17993, 2021. URL https://arxiv.org/abs/2107.03006.

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, volume 26, 2013. URL https://arxiv.org/abs/1305.6663.

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*, volume 35, 2022. URL https://arxiv.org/abs/2205.14987.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://arxiv.org/abs/2202.04200.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2208.04202.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Justin Deschenaux and Caglar Gulcehre. Beyond autoregression: Fast LLMs via self-distillation through time. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.21035.

Justin Deschenaux and Caglar Gulcehre. Language modeling with hyperspherical flows. *arXiv preprint arXiv:2605.11125*, 2026. URL https://arxiv.org/abs/2605.11125.

Justin Deschenaux, Caglar Gulcehre, and Subham Sekhar Sahoo. The diffusion duality, chapter II: Ψ-samplers and efficient curriculum. *arXiv preprint arXiv:2602.21185*, 2026. URL https://arxiv.org/abs/2602.21185.

Mingyu Jo, Jaesik Yoon, Justin Deschenaux, Caglar Gulcehre, and Sungjin Ahn. Loopholing discrete diffusion: Deterministic bypass of the sampling wall. *arXiv preprint arXiv:2510.19304*, 2025. URL https://arxiv.org/abs/2510.19304.

Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z. Pan, Hyeji Kim, Sham Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction. *arXiv preprint arXiv:2510.01384*, 2025. URL https://arxiv.org/abs/2510.01384.

Gregor Kornhardt, Jannis Chemseddine, Christian Wald, and Gabriele Steidl. Self-aware markov models for discrete reasoning. *arXiv preprint arXiv:2603.16661*, 2026. URL https://arxiv.org/abs/2603.16661.

Kirill Korolev, Nikita Morozov, Stepan Pavlenko, Esmeralda S. Whitammer, and Sergey Samsonov. Stop the sampler! classifier-based adaptive stopping for sampling kernels. *arXiv preprint arXiv:2606.16073*, 2026. URL https://arxiv.org/abs/2606.16073.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. Flow map language models: One-step language modeling via continuous denoising. *arXiv preprint arXiv:2602.16813*, 2026. URL https://arxiv.org/abs/2602.16813.

José Lezama, Huiwen Chang, Lu Jiang, and Irfan Essa. Improved masked image generation with token-critic. *arXiv preprint arXiv:2209.04439*, 2022. URL https://arxiv.org/abs/2209.04439.

David Li, Nikita Gushchin, Dmitry Abulkhanov, Eric Moulines, Ivan Oseledets, Maxim Panov, and Alexander Korotin. IDLM: Inverse-distilled diffusion language models. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.19066.

Bingbin Liu, Sébastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. TinyGSM: Achieving >80% on GSM8K with small language models. *arXiv preprint arXiv:2312.09241*, 2023. URL https://arxiv.org/abs/2312.09241.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Adrian Müller, Antoine Gonon, Zebang Shen, Ya-Ping Hsieh, and Niao He. Support before frequency in discrete diffusion. *arXiv preprint arXiv:2605.13999*, 2026. URL https://arxiv.org/abs/2605.13999.

Kirill Neklyudov, Evgenii Egorov, and Dmitry Vetrov. The implicit metropolis-hastings algorithm. *arXiv preprint arXiv:1906.03644*, 2019. URL https://arxiv.org/abs/1906.03644.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025. URL https://arxiv.org/abs/2502.09992.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2406.03736.

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. CANDI: Hybrid discrete-continuous diffusion models. *arXiv preprint arXiv:2510.22510*, 2025. URL https://arxiv.org/abs/2510.22510.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.07524.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T. Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 52584–52619, 2025. URL https://proceedings.mlr.press/v267/sahoo25a.html.

Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2112.06749.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.10193.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.04329.

Assaf Shocher, Amil Dravid, Yossi Gandelsman, Inbar Mosseri, Michael Rubinstein, and Alexei A. Efros. Idempotent generative network. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2311.01462.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2503.00307.

A SAMPLING ALGORITHMS

For generated positions I, the stopping score uses the probability of reproducing the current generated sequence:

$$p_\theta(z_I \mid z, c) = \prod_{i \in I} p_{\theta,i}(z_i \mid z, c). \tag{15}$$

The probability is not normalized by sequence length. It is computed before any temperature scaling used to sample an update. Theoretical thresholds concern the population optimum. For learned models, thresholds are fixed before testing. Task-specific masks are described in Appendix F.

This appendix summarizes categorical sampling from Section 3.1 and the annealed categorical sampling variant introduced for TinyGSM in Section 4.3. Algorithm 1 specifies generation with stopping, and Algorithm 2 specifies the annealing transition. Categorical sampling draws all coordinates independently. Annealed categorical sampling adds uniform corruption to these predictions, with decreasing noise over the trajectory.

For direct categorical generation, initialize Z_0 from the chosen prior. Each iteration makes one denoiser call. Its probabilities both evaluate (5) and define the proposed update. If stopping is enabled and the threshold is met, return the current state. Otherwise, sample the update. At the cap, return the last scored candidate without drawing an unevaluated update. The two-phase TinyGSM protocol is specified separately in Algorithm 3.

Algorithm 1 Categorical sampling with threshold stopping.

Require: condition c, call cap $B \geq 1$, threshold τ
1: Initialize generated tokens z_I from the prior and fix the condition.
2: $h \leftarrow 0$ ▷ Omit h for a model without self-conditioning.
3: **for** $j = 1, \ldots, B$ **do**
4: $(p, h') \leftarrow F_\theta(z, c, h)$ ▷ One model call.
5: $s \leftarrow \sum_{i \in I} \log p_i(z_i)$ ▷ Original, untempered probabilities.
6: **if** $s > \log \tau$ **then**
7: **return** z, accepted, NFE $= j$.
8: **end if**
9: **if** $j < B$ **then**
10: Draw $z_i \sim p_i$ independently for $i \in I$ and set $h \leftarrow h'$.
11: **end if**
12: **end for**
13: **return** z, budget fallback, NFE $= B$.

annealed categorical sampling. Let N be the number of model calls, with updates indexed by $k = 0, \ldots, N-1$. We use the grid

$$t_k = t_{\min} + (1 - t_{\min}) \left(1 - \frac{k}{N}\right)^\gamma, \qquad \alpha_t = 1 - (1 - \varepsilon)t, \quad t_{\min} = 10^{-5}, \quad \varepsilon = 10^{-3}. \tag{16}$$

The linear grid has $\gamma = 1$. Our TinyGSM annealing phase uses $\gamma = 0.125$, which allocates more calls to high noise levels. The same uniform replacement process defines the training corruptions. To return a denoiser proposal on the final call, we set

$$a_k = \begin{cases} \alpha_{t_k}, & k < N-1, \\ 1, & k = N-1. \end{cases} \qquad q_{k,i}(v) = a_k p^{(T)}_{\theta,i}(v \mid Z_k, c) + \frac{1 - a_k}{|\mathcal{V}|}. \tag{17}$$

For $T > 0$, temperature scaling gives $p^{(T)}_{\theta,i}(v \mid z, c) \propto p_{\theta,i}(v \mid z, c)^{1/T}$. For $T = 0$, it is a point mass at the highest-probability token, with ties broken by token index. Temperature acts on the denoiser prediction before adding uniform noise. At $T = 1$, (17) is equivalent to drawing a clean proposal from the factorized denoiser and then independently corrupting each generated token with probability $1 - a_k$. The last call uses $a_k = 1$ and counts toward the budget N. Condition tokens remain fixed throughout. For every intermediate update, $1 - a_k \geq \beta = (1 - \varepsilon)t_{\min} > 0$. Thus, every sequence of L generated tokens has conditional probability at least $(\beta/|\mathcal{V}|)^L$, independently of the prediction, temperature, or carried hidden state. This verifies the proposal condition of Theorem 2 through the noisy updates if stopping is checked. Our full protocol checks stopping in this phase as well as in the subsequent untempered categorical phase. The last clean annealing transition can lack this

lower bound. The finite-prefix extension in Appendix B.3 covers the complete process whenever the categorical continuation has uniform accessibility.

With self-conditioning, each denoiser call also reads and updates a hidden state, as specified in Appendix F. This adds no extra call at inference.

Algorithm 2 annealed categorical sampling step.

Require: state Z_k, condition c, update index k, budget N, temperature T
1: Evaluate $p_\theta(\cdot \mid Z_k, c)$ and apply temperature T.
2: If SC is enabled, carry this call's hidden state to the next call.
3: Set a_k using (17).
4: Draw each generated token $Z_{k+1,i} \sim q_{k,i}$ independently.
5: Preserve condition tokens and return Z_{k+1}.

Complete annealed categorical sampling. Algorithm 3 checks the same untempered stopping probability on every call in both phases. The transition changes from annealed categorical sampling to ordinary categorical sampling. At the population optimum, the recognition guarantee is independent of this choice. If the continuation has uniform accessibility, the finite-prefix extension of Theorem 2 gives $\Pr(T_{\mathcal{A}} > N + k) \le (1 - \varepsilon)^k$ for the complete process (proof in Appendix B.3). This statement concerns continuation without a hard cap. Experiments report capped fallback outcomes separately.

Algorithm 3 Annealed categorical sampling with threshold stopping.

Require: condition c, annealing calls N, continuation cap $B \ge 1$, threshold τ, annealing temperature T_{ann}
1: Initialize generated tokens z_I uniformly and fix condition tokens.
2: $h \leftarrow 0$ ▷ Omit h without self-conditioning.
3: **for** $j = 1, \ldots, N + B$ **do**
4: $(p, h') \leftarrow F_\theta(z, c, h)$ ▷ One call, $p_i(v) = p_{\theta,i}(v \mid z, c, h)$.
5: $s \leftarrow \sum_{i \in I} \log p_i(z_i)$ ▷ Before temperature or added noise.
6: **if** $s > \log \tau$ **then**
7: **return** z, accepted, NFE $= j$.
8: **end if**
9: **if** $j = N + B$ **then**
10: **return** z, budget fallback, NFE $= j$.
11: **end if**
12: **if** $j \le N$ **then**
13: Set a_{j-1} from (17).
14: Draw $z_i \sim a_{j-1} p_i^{(T_{\text{ann}})} + (1 - a_{j-1})\text{Unif}(\mathcal{V})$ for $i \in I$.
15: **else**
16: Draw $z_i \sim p_i$ independently for $i \in I$ ▷ Categorical, $T = 1$.
17: **end if**
18: $h \leftarrow h'$ and preserve condition tokens.
19: **end for**

B PROOFS

We first derive the population denoiser and prove validity recognition (Section B.1), then analyze generation without stopping (Section B.2), convergence with stopping (Section B.3), and exact absorption (Section B.4). We next prove the recognition bound under uniform corruption (Section B.5) and study a permutation example (Section B.6).

B.1 THE POPULATION DENOISER AND VALIDITY RECOGNITION

What the denoiser learns. An input z can be produced by several valid source solutions. Their posterior probabilities are

$$\pi_z(x) = \Pr(X = x \mid Z = z, C = c) = \frac{p_{\text{data}}(x \mid c) q(z \mid x, c)}{\bar{q}_c(z)}. \tag{18}$$

Minimizing token cross-entropy recovers the marginals of this posterior:

$$p_i^\star(a \mid z, c) = \Pr(X_i = a \mid Z = z, C = c) = \sum_{x \in \mathcal{S}(c)} \pi_z(x) \mathbf{1}\{x_i = a\}. \tag{19}$$

The prediction factorizes, but every marginal averages over the same posterior on valid solutions. This shared origin is what constrains the probability of reproducing an invalid input.

Fix c and suppress it where this causes no ambiguity. Coverage gives $\bar{q}_c(z) > 0$ at every input. Conditional on such an input, the contribution of position i to (4) is

$$-\sum_{a \in \mathcal{V}} \Pr(X_i = a \mid Z = z, C = c) \log p_{\theta,i}(a \mid z, c).$$

It equals the entropy of the source-posterior marginal plus its KL divergence to the predicted marginal. Minimization over all categorical distributions therefore gives (19) at every z.

Theorem 1 (restated). *Validity recognition under arbitrary corruption.* Fix c and let $p^\star$ minimize population denoising cross-entropy over all factorized categorical distributions on $\mathcal{V}^L$. Assume the training population has support $\mathcal{S}(c)$ and $\bar{q}_c(z) > 0$ for every z. Then every invalid input satisfies

$$z \notin \mathcal{S}(c) \quad \Longrightarrow \quad p^\star(z \mid z, c) \le (1 - 1/L)^L < e^{-1}.$$

Consequently, the rule $p^\star(z \mid z, c) > e^{-1}$ accepts only valid solutions, regardless of how a candidate is proposed. This conclusion allows corruption between distinct valid solutions.

Proof of Theorem 1. Let $u_i = p_i^\star(z_i \mid z, c)$. Write $d_{\text{H}}(X, z) = \sum_{i=1}^L \mathbf{1}\{X_i \neq z_i\}$ for the Hamming distance, the number of positions at which the sequences differ. Linearity of expectation gives

$$\sum_i (1 - u_i) = \sum_i \Pr(X_i \neq z_i \mid Z = z, C = c) = \mathbb{E}[d_{\text{H}}(X, z) \mid Z = z, C = c]. \tag{20}$$

If $z \notin \mathcal{S}(c)$, every source X differs from z in at least one position. Hence $\sum_i u_i \le L - 1$. By arithmetic–geometric mean,

$$p^\star(z \mid z, c) = \prod_i u_i \le \left(\frac{\sum_i u_i}{L} \right)^L \le (1 - 1/L)^L < e^{-1}.$$

For $L = 1$, the product is zero. For $L > 1$, the strict inequality follows from $\log(1 - v) < -v$ for $0 < v < 1$. This argument uses only valid reconstruction targets and the posterior identity. It does not require (10). Any candidate passing the threshold is therefore valid, independently of the procedure that produced it. □

Distance interpretation. Define $d_{\text{H}}(z, \mathcal{S}(c)) = \min_{x \in \mathcal{S}(c)} d_{\text{H}}(z, x)$, the minimum Hamming distance from z to a valid solution. The same identity gives a more quantitative statement for every input, including valid ones:

$$d_{\text{H}}(z, \mathcal{S}(c)) \le \sum_i [1 - p_i^\star(z_i \mid z, c)] \le -\log p^\star(z \mid z, c). \tag{21}$$

The first inequality follows because every source lies in support. The second applies $1 - u \le -\log u$ coordinatewise, with $-\log 0 = +\infty$. Thus the stay probability also bounds distance to support, while Theorem 1 gives the sharper product bound needed for recognition.

B.2 Invalid mass without stopping

Proposition 1 (restated). *Invalid mass without stopping.* Let $\mathcal{S}_i(c)$ be the tokens occurring at position i in valid solutions and $\mathcal{C}(c) = \prod_i \mathcal{S}_i(c)$.

If $q(z \mid x, c) > 0$ for every $z \in \mathcal{X}$, $x \in \mathcal{S}(c)$, and $\mathcal{S}(c) \subsetneq \mathcal{C}(c)$, categorical sampling from $p^\star$ without stopping converges, from every initialization, to a unique stationary distribution supported on $\mathcal{C}(c)$. This distribution assigns strictly positive probability to invalid states.

Proof of Proposition 1. Strict positivity of q implies $\pi_z(x) > 0$ for every valid source x and input z. By (19), the support of the ith optimal marginal is exactly $\mathcal{S}_i(c)$. Therefore $P(z, y) = p^\star(y \mid z, c) > 0$ if and only if $y \in \mathcal{C}(c)$. The chain enters $\mathcal{C}(c)$ after one update. Restricted to this finite set, its transition matrix is strictly positive, hence irreducible and aperiodic. It has a unique stationary distribution ν positive on every state of $\mathcal{C}(c)$, and every initial distribution converges to ν. Since $\mathcal{C}(c) \setminus \mathcal{S}(c) \neq \varnothing$, $\nu(\mathcal{X} \setminus \mathcal{S}(c)) > 0$.

In fact, the failure is bounded away from zero at every positive time. Finiteness and positivity give $b = \min_{z \in \mathcal{X}} P(z, \mathcal{C}(c) \setminus \mathcal{S}(c)) > 0$. Thus $\Pr(Z_k \notin \mathcal{S}(c)) \geq b$ for all $k \geq 1$, regardless of initialization. The assumption on coordinatewise combinations matters: if $\mathcal{S}(c) = \mathcal{C}(c)$, every proposal is valid after one update. □

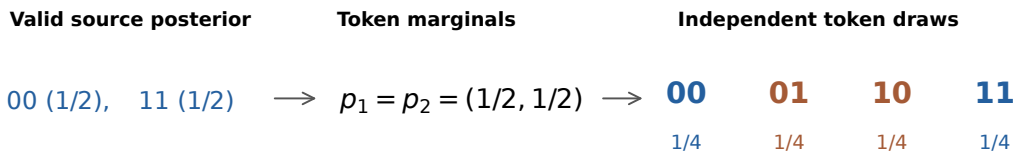

Figure 7: Factorized sampling can recombine valid sources into invalid candidates. With $\mathcal{S} = \{00, 11\}$ and input 01, a uniform source posterior gives equal token marginals. Half the independent draws are invalid (brown), although every source is valid (blue). The four-state example below gives a strictly positive corruption and the resulting exact population transition matrix.

A four-state illustration. Let $L = V = 2$, $\mathcal{S} = \{00, 11\}$, and let the data be uniform on these two strings. Independently replace each source token by a uniform binary token with probability $1/2$. Every $q(z \mid x)$ is positive. At inputs 00 and 11, the source posterior assigns probability 0.9 to that input. At 01 and 10, it is uniform over the two sources. In state order $(00, 01, 10, 11)$, the factorized population transition matrix is

$$P = \begin{pmatrix} .81 & .09 & .09 & .01 \\ .25 & .25 & .25 & .25 \\ .25 & .25 & .25 & .25 \\ .01 & .09 & .09 & .81 \end{pmatrix}.$$

Its stationary distribution is $(25, 9, 9, 25)/68$, with invalid mass $9/34 > 0$. Nevertheless, the diagonal scores are 0.81 on valid states and 0.25 on invalid states, so threshold e^{-1} separates them exactly. This illustrates why Proposition 1 concerns continued sampling, not false acceptance by the stopping test.

B.3 Convergence with threshold stopping

Let $\mathcal{F}_k = \sigma(Z_0, \ldots, Z_k)$ denote the candidate history. We stop at $T_{\mathcal{A}} = \inf\{k \geq 0 : Z_k \in \mathcal{A}\}$, before proposing another update.

Theorem 2 (restated). *Exponential convergence with threshold stopping.* Under the assumptions of Theorem 1, let $\mathcal{A} = \{z : p^\star(z \mid z, c) > e^{-1}\} \neq \varnothing$ and let $T_\mathcal{A}$ be its first visit time. Suppose a sampler, possibly depending on time and history, satisfies

$$\Pr(Z_{k+1} \in \mathcal{A} \mid Z_0, \ldots, Z_k) \geq \varepsilon > 0$$

for every history before stopping, with the same ε at every step. Then, for every initial distribution μ_0 and integer $k \geq 0$,

$$\Pr_{\mu_0}(T_\mathcal{A} > k) \leq (1-\varepsilon)^k \leq e^{-\varepsilon k}.$$

Stopping at $T_\mathcal{A}$ returns a valid solution almost surely.

Proof of Theorem 2. If $\mathcal{B}$ is empty, the process stops immediately. Otherwise, the uniform hitting condition gives

$$\begin{aligned}\Pr(T_\mathcal{A} > k+1) &= \mathbb{E}[\mathbf{1}\{T_\mathcal{A} > k\} \Pr(Z_{k+1} \notin \mathcal{A} \mid \mathcal{F}_k)] \\ &\leq (1-\varepsilon) \Pr(T_\mathcal{A} > k).\end{aligned}$$

Since $\Pr(T_\mathcal{A} > 0) = \mu_0(\mathcal{B})$, induction proves (9). Its right-hand side tends to zero, so termination is almost sure. Theorem 1 ensures that $Z_{T_\mathcal{A}} \in \mathcal{S}(c)$. This proof does not require a fixed transition matrix or the Markov property. If every next state has conditional probability at least $\eta > 0$, summing over $\mathcal{A}$ gives the sufficient bound $\varepsilon = |\mathcal{A}|\eta$. □

Applicability to categorical sampling. For the fixed population update $P(z, y) = p^\star(y \mid z, c)$, finiteness of the state space implies that $P(z, \mathcal{A}) > 0$ for every $z \in \mathcal{B}$ is sufficient:

$$\varepsilon = \min_{z \in \mathcal{B}} \sum_{a \in \mathcal{A}} P(z, a) > 0. \tag{22}$$

Strictly positive corruption ensures this condition when $\mathcal{A} \neq \varnothing$. For any $a \in \mathcal{A} \subseteq \mathcal{S}(c)$,

$$\pi_z(a) = \frac{p_{\text{data}}(a \mid c) q(z \mid a, c)}{\bar{q}_c(z)} > 0.$$

Each marginal $p_i^\star(a_i \mid z, c) \geq \pi_z(a)$, hence $P(z, a) \geq \pi_z(a)^L > 0$ for every z.

A finite initial phase. Suppose the accessibility condition in Theorem 2 holds only from step N onward. Conditional on any history surviving to that step, iteration of the same one-step bound gives

$$\Pr(T_\mathcal{A} > N + k \mid Z_0, \ldots, Z_N, T_\mathcal{A} > N) \leq (1-\varepsilon)^k.$$

Averaging yields $\Pr(T_\mathcal{A} > N+k) \leq \Pr(T_\mathcal{A} > N)(1-\varepsilon)^k \leq (1-\varepsilon)^k$. Thus arbitrary finitely many initial transitions preserve almost-sure termination and an exponentially decaying tail. Checking the same validity criterion during those transitions can only stop earlier at a valid candidate. This covers a finite annealing prefix followed by a uniformly accessible categorical continuation, without requiring the last clean annealing transition itself to have full support.

Mean time and the limiting distribution. Summing the tail yields $\mathbb{E}_{\mu_0} T_\mathcal{A} \leq \mu_0(\mathcal{B})/\varepsilon$. Checking before every update uses $T_\mathcal{A} + 1$ denoiser calls, including recognition of the final state when each check uses one call. Let $\widetilde{Z}_k = Z_{\min(k, T_\mathcal{A})}$ be the stopped process, μ_k its law, and μ_∞ the law of $Z_{T_\mathcal{A}}$. Mass at each accepted state can only increase, while μ_∞ is supported on $\mathcal{A}$. Therefore,

$$\|\mu_k - \mu_\infty\|_{\text{TV}} = \mu_k(\mathcal{B}) = \Pr(T_\mathcal{A} > k) \leq \mu_0(\mathcal{B})(1-\varepsilon)^k.$$

The limiting law can differ from $p_{\text{data}}(\cdot \mid c)$.

B.4 Absorption without transitions between valid solutions

Theorem 3 (restated). *Exact absorption and complete recognition.* Suppose the assumptions of Theorem 1 hold and corruption satisfies $q(z \mid x, c) = 0$ for distinct $x, z \in \mathcal{S}(c)$. Then

$$p^\star(z \mid z, c) = 1 \quad \Longleftrightarrow \quad z \in \mathcal{S}(c).$$

Hence $\mathcal{A} = \mathcal{S}(c)$: threshold stopping accepts every valid solution and rejects every invalid input. Categorical sampling reaches $\mathcal{S}(c)$ almost surely with an exponentially decaying hitting-time tail, and thereafter leaves the reached solution unchanged even without stopping.

Proof of Theorem 3. For $z \in \mathcal{S}(c)$, condition (10) excludes every source other than $X = z$. Coverage ensures this identity event has positive probability. Thus $\pi_z(z) = 1$, each $p_i^\star(z_i \mid z, c) = 1$, and $p^\star(z \mid z, c) = 1$. Theorem 1 excludes all invalid states from both the absorbing and accepted sets. Therefore $\mathcal{A} = \mathcal{S}(c)$ and valid states are exactly the absorbing states of the categorical chain.

For any $z \notin \mathcal{S}(c)$, coverage supplies some $x \in \mathcal{S}(c)$ with $\pi_z(x) > 0$. Every $p_i^\star(x_i \mid z, c) \geq \pi_z(x)$, and hence $p^\star(x \mid z, c) \geq \pi_z(x)^L > 0$. Finiteness then gives a uniform positive hitting probability in (22). Theorem 2 applies with $\mathcal{A} = \mathcal{S}(c)$, giving the exponentially decaying hitting-time tail and almost-sure arrival at a valid solution. Such a solution is subsequently unchanged. Finally, equality of the accepted set and support means that the threshold fires exactly at the first support visit along any sequence of proposed candidates. □

B.5 Recognition under uniform corruption

We retain the valid-data assumption of Section 3.1 but allow corruption between distinct valid solutions. Fix c and write $\pi(x) = p_{\text{data}}(x \mid c)$ and $S = \mathcal{S}(c)$. Here L counts the corrupted solution positions. Any condition prefix stays fixed and is excluded from L. The corruption process draws one shared $t \sim \mathcal{U}[0, 1]$ per sequence and replaces each token independently with a uniform token:

$$q(z \mid x) = \int_0^1 \prod_{i=1}^{L} \left[(1 - t)\mathbf{1}\{z_i = x_i\} + \frac{t}{V} \right] dt. \tag{23}$$

The time-independent denoiser minimizes the population CE in (4) over all factorized categorical distributions. Because (23) gives every input positive probability, its optimum at every z is $p_i^\star(a \mid z) = \Pr(X_i = a \mid Z = z)$.

To prove Proposition 2, write $C_{L,V} = 1 - V^{-(L+1)}$. We first establish the more quantitative retention bound, valid for every $z \in S$:

$$p^\star(z \mid z, c) \geq \exp\left[-\frac{1 - \pi(z)}{L(V - 1)C_{L,V}\pi(z)} \right]. \tag{24}$$

Together with $p^\star(z \mid z, c) \leq (1 - 1/L)^L < e^{-1}$ for $z \notin S$, this gives the recognition guarantee in the main text. We also quantify recall under the data distribution. Suppose a subset of S with at most M elements carries probability at least $1 - \eta$. Then

$$\Pr_{X \sim p_{\text{data}}(\cdot \mid c)}[p^\star(X \mid X, c) > e^{-1}] \geq 1 - \eta - M\delta_{L,V}. \tag{25}$$

This is a bound on accepted probability mass, rather than on the fraction of distinct solutions. At most η can be lost outside the subset. Inside it, at most M solutions can each lose mass at most $\delta_{L,V}$ by falling below the sufficient probability cutoff.

Proposition 2 (restated). *Recognition under uniform corruption.* For the uniform-token corruption in (23), let $L \geq 1$ and $V = |\mathcal{V}| \geq 5$. At the population optimum, every solution x satisfying

$$p_{\text{data}}(x \mid c) > \delta_{L,V}, \qquad \delta_{L,V} = \frac{1}{1 + L(V - 1)(1 - V^{-(L+1)})}$$

has $p^\star(x \mid x, c) > e^{-1}$ and is therefore accepted. The cutoff $\delta_{L,V}$ is of order $1/(LV)$.

Proof of Proposition 2. We first relate self-reconstruction to contributions from other valid sources. For $z \in S$, put

$$r_x = \frac{\pi(x)q(z \mid x)}{\pi(z)q(z \mid z)} \quad (x \in S \setminus \{z\}), \qquad B = \sum_{x \neq z} r_x, \qquad H = \sum_{x \neq z} d_{\text{H}}(x, z) r_x.$$

The denominator is positive. Let $b_i = \sum_{x \neq z} r_x \mathbf{1}\{x_i \neq z_i\}$. Then $0 \leq b_i \leq B$, $\sum_i b_i = H$, and $p_i^\star(z_i \mid z) = 1 - b_i/(1 + B)$. For $B > 0$, concavity of $\log(1 - b/(1 + B))$ on $[0, B]$ gives

$$\log p^\star(z \mid z, c) = \sum_i \log\left(1 - \frac{b_i}{1 + B}\right) \geq -\frac{H}{B} \log(1 + B) \geq -H.$$

If $B = 0$, the denoiser preserves z with probability one and the same bound holds.

Next, uniform corruption depends only on $d = d_{\mathrm{H}}(x, z)$:

$$K_d = q(z \mid x) = \int_0^1 (1 - t + t/V)^{L-d}(t/V)^d \, dt, \qquad K_0 = \frac{VC_{L,V}}{(V-1)(L+1)}.$$

Substituting $u = (1 - 1/V)t$ and extending the integration interval to $[0, 1]$ gives, for $d \geq 1$,

$$\frac{K_d}{K_0} \leq \frac{L+1}{C_{L,V}(V-1)^d} \int_0^1 u^d (1-u)^{L-d} \, du = \frac{1}{C_{L,V}\binom{L}{d}(V-1)^d}.$$

Since $\binom{L}{d} \geq L/d$ and $d^2 \leq 4^{d-1} \leq (V-1)^{d-1}$, we obtain

$$d\frac{K_d}{K_0} \leq \frac{1}{L(V-1)C_{L,V}}, \qquad H \leq \frac{1 - \pi(z)}{L(V-1)C_{L,V}\pi(z)}.$$

Combining this estimate with $p^\star(z \mid z, c) \geq e^{-H}$ proves (24). Its right-hand side exceeds e^{-1} whenever $\pi(z) > \delta_{L,V}$.

For $z \notin S$, every possible source differs from z in at least one position. Hence $\sum_i [1 - p_i^\star(z_i \mid z)] = \mathbb{E}[d_{\mathrm{H}}(X, z) \mid Z = z] \geq 1$. The arithmetic–geometric mean inequality gives the claimed upper bound on $p^\star(z \mid z, c)$, exactly as in Theorem 1. This argument does not require filtering.

Finally, let $G \subseteq S$ have at most M elements and mass at least $1 - \eta$. The mass of programs with $\pi(x) \leq \delta_{L,V}$ outside G is at most η. Their mass inside G is at most $M\delta_{L,V}$. All remaining programs pass the threshold by Proposition 2, proving (25). □

A rate bound for Theorem 2. For the population categorical sampler, the likelihoods defined above satisfy $K_L \leq q(z \mid x) \leq K_0$, with

$$K_L = \frac{1}{(L+1)V^L}, \qquad \beta = \frac{K_L}{K_0} = \frac{V-1}{V^{L+1} - 1}.$$

Thus $\pi_z(a) \geq \beta\pi(a)$ for every $a \in \mathcal{A}$ and input z. Substituting $P(z, a) \geq \pi_z(a)^L$ into (22) yields

$$\varepsilon \geq \beta^L \sum_{a \in \mathcal{A}} \pi(a)^L > 0. \tag{26}$$

This bound certifies an exponential rate but does not guarantee a small stopping time. The threshold guarantees validity, while the rate depends on reaching the accepted set. For any strictly positive corruption process with a nonempty accepted set, the same proof applies with the rate in (22).

Interpreting the cutoff. At $V = 49{,}153$ and $L = 400$, the cutoff is approximately 5.0863×10^{-8}, with reciprocal approximately 19.7 million. This is a probability of an entire program conditional on one fixed question, not a token probability or a frequency across different tasks. If 100,000 programs carry 99% of this conditional distribution, (25) gives an accepted mass of at least $0.99 - 100{,}000\delta_{L,V} > 0.9849$. This example does not impose a probability floor on the remaining programs.

A small cutoff alone does not imply high coverage. For example, a uniform distribution on 2^{25} distinct valid programs assigns each probability $2^{-25} < \delta_{400,49153}$. The pointwise bound then guarantees no accepted mass, although the denoiser may still recognize these programs. The concentration premise in (25) is what turns the per-program guarantee into a guarantee on most of the target mass.

Noise schedules and time weights. The recognition bounds (24) and (25) also hold for $\alpha(t) = 1 - (1 - \epsilon)t$, $0 \leq \epsilon < 1$, and CE weights $(1 - t)^p$, $p \geq 0$, using the effective kernel of Appendix C. To make the dependence explicit, let $K_d^{\epsilon,p}$ denote the effective corruption probability between strings at Hamming distance d. Set $a = (1 - \epsilon)(V - 1)/V = u_{\max}$. Matching positions contribute $1 - at$, whereas mismatching positions contribute $at/(V - 1)$. The common normalization of the time weights cancels in the ratio:

$$\begin{aligned}\frac{K_d^{\epsilon,p}}{K_0^{\epsilon,p}} &= \frac{\int_0^1 (1-t)^p (1-at)^{L-d}[at/(V-1)]^d \, dt}{\int_0^1 (1-t)^p (1-at)^L \, dt} \\ &= \frac{\int_0^a (1-u/a)^p (1-u)^L \left[\frac{u}{(V-1)(1-u)}\right]^d du}{\int_0^a (1-u/a)^p (1-u)^L \, du},\end{aligned}$$

where the second line substitutes $u = at$. Thus, defining the normalized density

$$f_{a,p}(u) = \frac{(1-u)^L(1-u/a)^p}{\int_0^a (1-v)^L(1-v/a)^p\,dv}, \qquad 0 \le u \le a,$$

gives

$$\frac{K_d^{\epsilon,p}}{K_0^{\epsilon,p}} = \mathbb{E}_{U\sim f_{a,p}}\left[\left(\frac{U}{(V-1)(1-U)}\right)^d\right].$$

For $d \ge 1$, the function inside the expectation is increasing. When $p = 0$, increasing ϵ decreases a and simply truncates the reference density proportional to $(1-u)^L$ to a smaller interval. For fixed a, multiplying that density by the decreasing weight $(1-u/a)^p$ with $p \ge 0$ also decreases the expectation of an increasing function. Hence $K_d^{\epsilon,p}/K_0^{\epsilon,p} \le K_d^{0,0}/K_0^{0,0}$, so the ratio bounds used above remain valid. The effective corruption remains strictly positive, so the convergence proof also applies whenever the accepted set is nonempty. Its rate is given by (22) for the corresponding population denoiser.

B.6 Parallel generation of uniformly distributed permutations

This example separates the cost of irreversible token commitment from the cost of repeated parallel correction. Let $\mathcal{V} = \{1, \dots, L\}$ and let $\mathcal{S}_L$ be all permutations of $\mathcal{V}$. The data distribution is uniform on $\mathcal{S}_L$. Both methods below have access to their population-optimal token predictions. A denoiser evaluation returns all token distributions for the current input. We count these evaluations, rather than scalar arithmetic operations used to compute the analytical predictions.

Exact irreversible unmasking. We consider the carry-over masked-sampling mechanism used by MDLM (Sahoo et al., 2024): revealed tokens never change, and newly revealed values are sampled independently from the exact conditional marginals. The positions to reveal are chosen before these values are drawn. The sampler may adapt this choice to the current partial sequence and may resample empty reveal attempts without another denoiser call. It does not reject sampled values, remask tokens, or condition the joint draw on the permutation constraint.

Proposition 3 (Calls required by independent irreversible unmasking)**.** *Within this sampling class, exact generation from the uniform law on $\mathcal{S}_L$, starting with all positions masked, requires at least L nonempty denoiser updates almost surely. Sequentially revealing one position per update attains this bound and samples the target law exactly.*

Proof. An exact sampler can only visit partial sequences with distinct revealed symbols. If n positions remain masked, each of their conditional marginals is uniform over the n unused symbols. Suppose an update reveals $r \ge 2$ positions with positive probability. Independence gives probability at least $1/n > 0$ that two of the revealed values coincide. These tokens cannot subsequently change, so this event forces an invalid output. Exact generation therefore reveals at most one token per nonempty update, giving the lower bound L. Choosing one position and sampling uniformly among unused symbols at each step yields each permutation with probability $\prod_{n=1}^{L} n^{-1} = 1/L!$. This proves attainability. □

The restriction to irreversible independent draws is essential. Value-dependent rejection, joint constrained sampling, and revisable uniform-state diffusion are different sampling mechanisms. In particular, this argument does not give an L-call lower bound for UDLM. Empty masked updates do not contribute to the lower bound.

EqDLM (ours) with ordinary uniform corruption. We keep (4) and use the kernel (23) with $V = L$:

$$q(z \mid x) = \int_0^1 \prod_{i=1}^{L}\left[(1-t)\mathbf{1}\{z_i = x_i\} + \frac{t}{L}\right] dt. \tag{27}$$

The factorized population optimum is $p^\star(x \mid z) = \prod_i \Pr(X_i = x_i \mid Z = z)$. The next state is drawn from this product, with no temperature change, masking, or repeated corruption. Before drawing it, we stop if $p^\star(z \mid z) > e^{-1}$.

Recognition and the output law. The off-support argument of Theorem 1 gives $p^\star(z \mid z) \le (1-1/L)^L < e^{-1}$ for every nonpermutation. For a permutation x, the number of other permutations

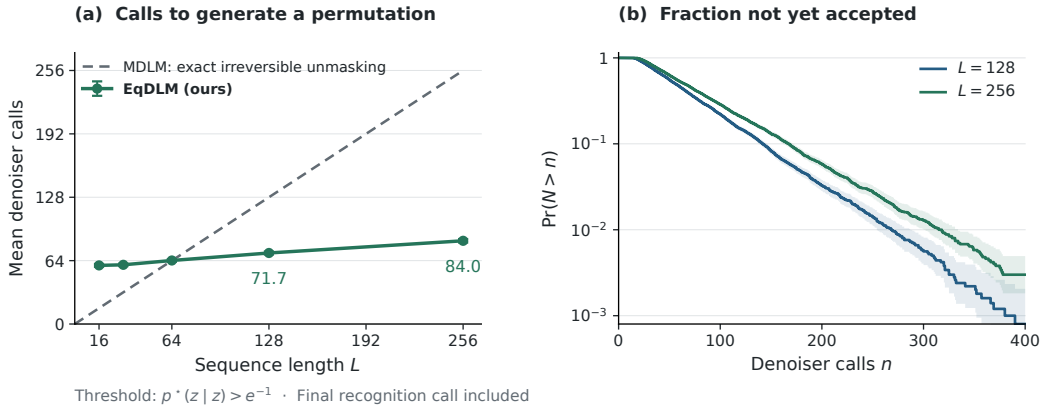

Figure 8: Generating uniformly distributed permutations with population denoisers. **EqDLM (ours)** uses uniform-token corruption, categorical updates, and stopping at $p^\star(z \mid z) > e^{-1}$. **(a)** Mean denoiser calls from 5,000 independent trajectories per length. Error bars give 95% Monte Carlo intervals. The dashed line is the exact L-call cost of sequential irreversible unmasking, which is optimal within the independent masked-sampling class in Proposition 3. **(b)** Empirical stopping-time tails on a logarithmic probability axis, with pointwise 95% intervals. All counts for **EqDLM (ours)** include the final recognition call. Validity and the uniform output law are proved in Appendix B.6, while the mean stopping times are numerical estimates.

at Hamming distance d is exactly $\binom{L}{d} D_d$, where D_d is the number of derangements of d symbols. For a uniform source distribution, the posterior-ratio argument in Appendix B.5 gives $p^\star(x \mid x) \geq \exp(-\sum_{d=1}^{L} d\binom{L}{d} D_d K_d / K_0)$. Using $K_d/K_0 \leq [(1 - L^{-(L+1)})\binom{L}{d}(L-1)^d]^{-1}$ therefore gives

$$p^\star(x \mid x) \geq e^{-H_L}, \qquad H_L = \frac{1}{1 - L^{-(L+1)}} \sum_{d=2}^{L} \frac{d D_d}{(L-1)^d}. \tag{28}$$

With $D_0 = 1$, $D_1 = 0$, and $D_d = (d-1)(D_{d-1} + D_{d-2})$, rational evaluation certifies $H_L < 0.013 < 1$ at every plotted length $L \in \{16, 32, 64, 128, 256\}$. Thus the accepted set is exactly $\mathcal{S}_L$. Uniform corruption is strictly positive, so Theorem 2 gives almost-sure termination and a exponentially decaying stopping-time tail from every initial state.

We start from $Z_0 = (1, \ldots, 1)$, which is invariant under permutations of the coordinate positions. The target law, corruption, denoiser, and stopping rule are all equivariant under the same transformations. Consequently, the terminal law is invariant under coordinate permutations. Since this group acts transitively on $\mathcal{S}_L$, the terminal law is uniform on $\mathcal{S}_L$. This establishes exact generation of the target distribution despite factorized updates at every call.

Sampling cost. Figure 8 evaluates the stopping time using the analytical denoiser. At $L = 128$ and 256, generation takes 71.7 and 84.0 calls on average, compared with 128 and 256 for exact irreversible unmasking. These counts include the final call that recognizes the solution. The measured advantage appears at larger L, rather than uniformly across lengths. This example shows how revisable parallel updates and recognition can handle token dependence without committing to one token at a time.

Analytical posterior evaluation. We compute token marginals without enumerating $L!$ permutations or discretizing the corruption time. Expanding (27) over sets of matching positions gives

$$q(z \mid x) = \sum_{A \subseteq \{1,\ldots,L\}} w_{|A|} \prod_{i \in A} \mathbf{1}\{x_i = z_i\}, \qquad w_k = \frac{k!(L-k)!}{(L+1)! L^{L-k}}. \tag{29}$$

Let c_a count occurrences of symbol a in z, and define coefficients $E_k(z)$ by

$$\prod_{a=1}^{L} (1 + c_a u) = \sum_{k=0}^{L} E_k(z) u^k.$$

A set of k matched positions must use k distinct symbols and has $(L-k)!$ possible permutation completions. Therefore its contribution to the posterior normalizer is $w_k(L-k)!E_k(z)$. For a stable common rescaling, define

$$b_0 = 1, \qquad b_{k+1} = b_k \frac{(k+1)L}{(L-k)^2}, \qquad B(z) = \sum_{k=0}^{L} b_k E_k(z). \tag{30}$$

For a fixed position i and proposed symbol a, remove position i from the counts and then remove all occurrences of symbol a. Let $E_k^{i,a}(z)$ be the polynomial coefficients of these remaining counts. The token posterior is

$$p_i^\star(a \mid z) = \frac{\displaystyle\sum_{k=0}^{L-1} E_k^{i,a}(z) \left[\frac{b_k}{L-k} + \mathbf{1}\{a = z_i\} b_{k+1} \right]}{B(z)}. \tag{31}$$

The two terms correspond to leaving position i unselected in the matching expansion and selecting it when $a = z_i$. All coefficients are nonnegative.

Stopping-time measurement. For each length, we simulate $5{,}000$ independent trajectories using (31) in float64. All trajectories finish at the probability threshold, and all outputs are valid. Validity is checked only to verify the implementation, not to choose an update or decide whether to stop. If T categorical updates occur before the first accepted state, the reported cost is $N = T + 1$, including its recognition call. Our implementation exploits symbol and position symmetries by storing occupancy counts. It samples the same transition probabilities as the full token process.

We verify every token posterior against direct Bayes evaluation for all inputs at $L = 2, 3, 4, 5$. At $L = 2, \ldots, 6$, we also enumerate the transition matrix over occupancy classes and solve its hitting-time linear system. Independent simulations agree with these exact finite-state means. These checks test the optimized categorical sampler separately from the recognition guarantees.

Table 2: Permutation generation with population denoisers. **EqDLM (ours)** costs are means and 95% Monte Carlo intervals from $5{,}000$ trajectories per length. MDLM costs are the attainable lower bound of Proposition 3. The final column is the analytical stay probability at any valid permutation, rounded to six decimal places.

L	**EqDLM (ours)**	MDLM	$p^\star(x \mid x),\ x \in \mathcal{S}_L$
16	59.18 ± 1.44	16	0.988065
32	59.78 ± 1.32	32	0.997674
64	64.14 ± 1.36	64	0.999470
128	71.73 ± 1.49	128	0.999873
256	84.03 ± 1.77	256	0.999969

The intervals in Table 2 quantify Monte Carlo uncertainty in mean stopping times. They are not analytical upper bounds on these means. The exponential-convergence theorem proves termination, while this calculation measures its cost on the specified distribution. In particular, shorter sequences do not show a speed advantage over sequential unmasking.

C Training kernel and time weights

Algorithm 4 implements the reconstruction objective.

Algorithm 4 One training step of an Equilibrium Diffusion Language Model.

Require: Denoiser p_θ, corruption kernels q_t, weights $w(t)$, step size $\eta > 0$

1: Sample a training pair (C, X)
2: Sample $t \sim \mathrm{Unif}[0, 1]$
3: Sample $Z \sim q_t(\cdot \mid X, C)$
4: $\ell \leftarrow -\sum_{i=1}^{L} \log p_{\theta,i}(X_i \mid Z, C)$
5: $\theta \leftarrow \theta - \eta\, w(t) \nabla_\theta \ell$

Uniform corruption and transitions between solutions. Our experiments start from independent uniform-token replacements,

$$\tilde{q}_t(z_i \mid x_i) = \alpha_t \, \mathbf{1}\{z_i = x_i\} + (1 - \alpha_t) \, \frac{1}{|\mathcal{V}|}, \tag{32}$$

where α_t decreases as the noise level increases. Using these proposals directly permits transitions between valid solutions. When a validity test is available during training, we can instead reject proposals at a different valid solution and retain identity pairs and invalid inputs. Returning rejected mass to the source X also enforces (10). The Swiss-roll experiment varies the probability of retaining transitions between solutions. Sudoku and TinyGSM use corruption.

In Section 3.1, we sample a noise level t and an input from q_t. The denoiser does not receive t. Averaging over the sampled noise level gives the kernel q used in (4). More generally, weighting the loss by $w(t) \geq 0$ with $0 < \int_0^1 w(t)\, dt < \infty$ gives, up to a positive constant factor, the same objective with

$$q(z \mid x, c) = \frac{\int_0^1 w(t) q_t(z \mid x, c)\, dt}{\int_0^1 w(t)\, dt}. \tag{33}$$

Setting $w(t) = 1$ recovers unweighted training. The exact-absorption guarantees require (7) and (10) for the effective kernel. Appendix B.5 treats recognition under uniform corruption separately.

D SUDOKU TRAINING AND EVALUATION

Data and model. We generate $10{,}000{,}000$ completed Sudoku grids using randomized backtracking with data seed 42. Each grid is a row-major sequence of 81 digits with vocabulary size nine. The final $10{,}000$ grids form the validation set, leaving $9{,}990{,}000$ training examples. The Diffusion Transformer has hidden size 512, eight blocks, eight attention heads, condition dimension 128, and dropout 0.1. The model is time-independent and does not use self-conditioning.

Training. We minimize (4), averaged over all 81 tokens, using Algorithm 4 with $w(t) = 1$. The corruption is (32) with $\alpha_t = 1 - t$ and one shared $t \sim \mathcal{U}[0, 1]$ per example. Every position is corrupted independently conditional on t. Corruption allows transitions between valid grids. Training uses $100{,}000$ AdamW updates with global batch size $1{,}024$, learning rate 3×10^{-4}, $\beta = (0.9, 0.999)$, $\varepsilon = 10^{-8}$, zero weight decay, and gradient clipping at norm one. The learning rate is constant after $2{,}500$ warmup updates. We train in BF16 on eight A100 GPUs, with 128 examples per GPU, and evaluate the final checkpoint's exponential moving average (EMA) weights with decay 0.9999.

Sampling and score precision. Each chain starts with 81 independent uniform digits and uses (6) at temperature one. The Transformer retains its BF16 computation, while log-softmax is computed in FP32 and categorical sampling and score accumulation use FP64. TF32 is disabled. We compute $\log p_\theta(x \mid x) = \sum_{i=1}^{81} \log p_{\theta,i}(x_i \mid x)$ without length normalization. The same denoiser call provides both this score and the categorical update.

Fixed-budget comparison. The MDLM and FMLM+ rows of Table 3 are taken from Table 6 of Agarwal et al. (2026), without retraining those models. Here $\delta_{\mathbf{0},\mathbf{0}}$ denotes their MDM-like token prediction from noise and available context, with confidence-based token selection. The $\delta_{\mathbf{0},\mathbf{1}}$ variant uses a noise-to-data flow-map jump with posterior refinement (PR), retaining confident tokens from the jointly proposed sequence. We evaluate **EqDLM (ours)** and our trained Duo baseline on $10{,}000$ generated grids at each budget $\{1, 3, 9, 27, 81\}$, batch size 128, and no stopping. **EqDLM (ours)** produces 0, 31, $6{,}063$, $9{,}925$, and $10{,}000$ valid grids, respectively. Duo produces 0, 0, $3{,}567$, $9{,}473$, and $9{,}878$. The 81-call **EqDLM (ours)** sample also provides the diversity measurements below. Threshold calibration and stopping evaluation use separate cohorts of $2{,}048$ and $10{,}000$ trajectories, respectively.

Duo baseline. We use the discrete uniform-diffusion ELBO implementation of S-FLM (Deschenaux & Gulcehre, 2026). The model uses the same training and validation grids, Transformer dimensions, $100{,}000$ updates, batch size, optimizer, learning-rate schedule, and EMA as **EqDLM (ours)**. Both models omit self-conditioning. Duo uses time conditioning and antithetic time sampling on $[0.001, 1)$, with the same corruption schedule $\alpha_t = 1 - t$. Inference uses ancestral posterior sampling at temperature one and a linear time grid, with the final denoising transition included in NFE. We retain

Table 3: Unconditional Sudoku validity (%). †: results taken from Agarwal et al. (2026), Table 6. We train Duo following the S-FLM implementation (Deschenaux & Gulcehre, 2026) and **EqDLM (ours)** on the same data with the same training budget. Both rows use 10,000 generated grids at each NFE. Both stopping rules use the same separate set of 10,000 trajectories, with the stopping check included in mean NFE.

Method	Fixed NFE				
	1	3	9	27	81
MDLM, ancestral†	0.0	0.0	0.0	7.7	33.8
MDLM, confidence†	0.0	0.0	0.0	0.0	97.2
FMLM+, $\delta^{\dagger}_{\mathbf{0},\mathbf{0}}$	0.0	0.0	0.0	0.0	98.8
FMLM+, $\delta_{\mathbf{0},\mathbf{1}}$ (PR)†	0.0	**22.2**	**87.1**	98.8	99.6
Duo, ancestral	0.00	0.00	35.67	94.73	98.78
EqDLM (ours), categorical	0.00	0.31	60.63	**99.25**	**100.00**
EqDLM (ours), stopping at $p_\theta(x \mid x) > e^{-1}$	97.24% at 10.16 mean NFE				
EqDLM (ours), stopping at $p_\theta(x \mid x) > 0.9$	**99.99**% at **10.43** mean NFE				

Table 4: Stopping-rule diagnostics on the same 10,000 categorical trajectories, with a cap of 81 calls. The cap is an upper limit, while the adaptive rows report the calls actually required by each policy. A false stop accepts an invalid grid through the threshold.

Stopping rule	Valid (%)	Mean NFE	False stops
No stopping	99.98	81.00	—
$p_\theta(x \mid x) > e^{-1}$	97.24	10.16	276
$p_\theta(x \mid x) > 0.822546$ (calibrated)	99.99	10.43	0
$p_\theta(x \mid x) > 0.9$	99.99	10.43	0

the numerical precision settings described above. Since $t = 1$ gives an infinite noise embedding when $\alpha_t = 1 - t$, we replace only the first time by the nearest FP32 value below one, $1 - 2^{-24}$. The remaining grid points are unchanged and the final transition targets $\alpha = 1$.

Threshold selection and call accounting. We evaluate the first 81 updates of 2,048 calibration trajectories. The largest invalid-state score is 0.645110 and the smallest valid-state score is 0.999983. Their midpoint, $\tau = 0.822546$, is frozen before generating 10,000 new test trajectories. We also specify the fixed thresholds e^{-1} and 0.9 before this test. The calibrated threshold and 0.9 give identical stopping decisions. Calibration and test seeds are 271828 and 314159, respectively, with sampling batch size 128.

We evaluate each stopping rule on saved paths truncated to 81 updates. If x_k is the state after k updates, accepting x_k costs $k + 1$ calls, including its score evaluation. For a budget of K calls we check only $x_0, \ldots, x_{K-1}$. If no state passes the threshold, we return x_K. A separate score evaluation at the terminal state is used only for diagnostics and is excluded from the stopping policies. The model has no time or latent dependence, so the retained prefix of each path implements the corresponding stopping rule. With $\tau = 0.9$, the median cost is nine calls and the 95th percentile is 18. The 95% Clopper–Pearson interval for final validity is $[99.9443\%, 99.9997\%]$. No false stop is observed on the 10,000 test paths, giving an upper endpoint of 0.0369% for its two-sided 95% interval.

Stability of valid states. Across the test paths, 1,273 invalid-state observations exceed e^{-1}, belonging to 276 trajectories. Once a valid state is reached, the paths contain 16 valid-to-invalid transitions and no transitions to a different valid grid. The model scores predict 10.28 departures from valid states in total. Of the 16 trajectories that lose a solution, 15 recover it within the budget. One remains invalid at update 81, and a separate trajectory never reaches a solution. This accounts for the 9,998 valid outputs without stopping, compared with 9,999 under threshold stopping.

To measure single-step retention separately, we take 1,024 held-out valid grids and 1,024 valid outputs from the test trajectories. For each input we draw 4,096 independent categorical successors. Of 4,194,304 transitions per group, 49 and 57, respectively, produce invalid grids. All other successors equal the input. The mean probabilities of preserving the input are 0.99998561 and 0.99998565

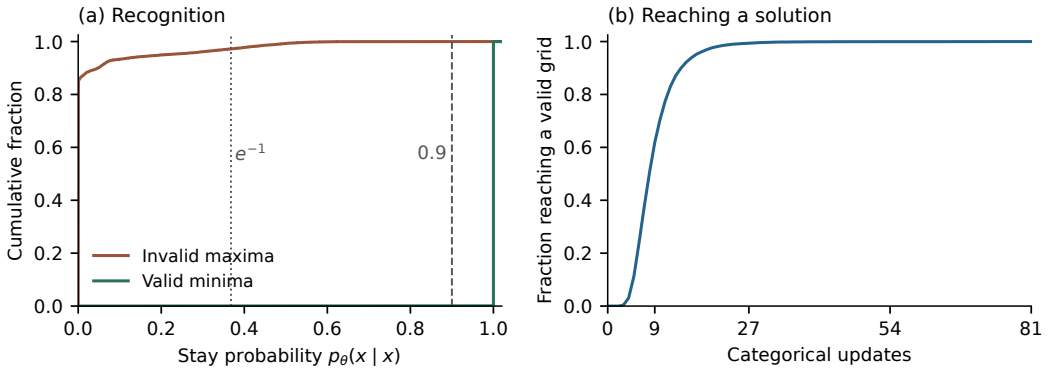

Figure 9: Recognition along categorical Sudoku trajectories. (a) For each test trajectory, the largest stay probability at an invalid state and the smallest at a valid state. The latter is defined for the 9,999 trajectories that reach a solution. Curves show their empirical cumulative distributions. (b) Fraction of test trajectories that have reached a valid state after each categorical update. Paths are analyzed through 81 updates, with 9,999/10,000 trajectories reaching a solution.

for the two groups. These repeated draws estimate transitions from fixed input cohorts, rather than treating every successor as a new independent input grid.

Any different valid Sudoku must change at least four cells. For an arbitrary factorized denoiser and valid input x, let D count the changed cells in its categorical proposal Y. Then

$$0 \le \Pr(Y \in \mathcal{S} \mid x) - p_\theta(x \mid x) = \Pr(Y \in \mathcal{S} \setminus \{x\} \mid x) \le \Pr(D \ge 4). \tag{34}$$

Here D has a Poisson-binomial distribution with change probabilities $1 - p_{\theta,i}(x_i \mid x)$, so its tail can be computed directly. For the two tested cohorts, the largest tail is below 3.7×10^{-21}. This is an upper bound on the probability of moving to a *different* valid grid. Thus the probability of producing any valid grid differs negligibly from that of preserving the same grid on these inputs. The high probability of preservation itself is measured separately above, rather than established by this tail bound.

Diversity. In a separate evaluation, we generate 10,000 grids using 81 categorical updates and no stopping. All are valid and distinct, and none exactly matches any of the 9,990,000 training grids. To remove variation due only to digit names, we relabel each grid's first row as $1, \ldots, 9$ and apply the same permutation throughout the grid. All 10,000 grids remain distinct after this operation. The mean nearest-neighbor Hamming distance is 43.470 cells within the generated sample and 43.372 within the 10,000 held-out grids, excluding self-matches. These checks measure repetition and local concentration, without establishing uniform generation over all solutions.

Recorded generation example. Figure 4 shows test trajectory 69 (zero-indexed) from the 10,000-sample stopping cohort. All intermediate grids were retained, and validity was recomputed from the row, column, and box constraints. The stay probabilities come from the corresponding model calls. Scoring the grid after 81 updates requires the next call and is used only for the illustration. This call is excluded from the stopping policies. The example illustrates correction and retention, whose frequencies are measured by the aggregate evaluations above.

E Compute

Sudoku training uses a global batch of 1,024 and 100,000 optimizer updates. Sampling and trajectory diagnostics use one A100 GPU. The TinyGSM checkpoint configurations use eight GPUs, a global batch of 512, and 250,000 optimizer updates. Self-conditioning adds a precursor forward pass on 90% of training microbatches. The reported GSM8K evaluations run on A100 GPUs. The full project used additional compute on failed experiments beyond the runs reported here.

F TINYGSM TRAINING AND EVALUATION SETTINGS

Data and training. Both EqDLM TinyGSM checkpoints use a Diffusion Transformer with about 168M parameters, hidden size 768, 12 blocks, 12 attention heads, conditioning width 128, dropout 0.1, sequence length 512, and a 49,153-token vocabulary based on the SmolLM-135M tokenizer. The network does not receive the noise level. We sample $t \sim \mathcal{U}[0,1]$ and use uniform-token corruption with $\alpha_t = 1 - 0.999t$, restricted to answer and pad positions. The token CE in (4) is weighted by $w(t) = 1 - t$, as in (33). Both checkpoints are evaluated at 250,000 training updates with EMA decay 0.9999, before any replay-based continuation. The GSM8K prefix is [BOS], the question, and the separator. That prefix stays clean and is excluded from the loss. Pad tokens in the answer region are included in the training loss. Sandbox scoring then strips special tokens from the executed program. Training uses a global batch of 512, AdamW with learning rate 3×10^{-4}, $\beta = (0.9, 0.999)$, $\varepsilon = 10^{-8}$, zero weight decay, and gradient clipping at norm one. The learning rate is constant after 2,500 warmup updates. Both saved configurations use eight GPUs, physical batches of 32, and two accumulation steps. These are separately trained variants rather than an ablation changing only SC. Training-time validation is token denoising NLL on the TinyGSM holdout, not GSM8K accuracy.

Self-conditioning. We adopt the latent propagation and two-pass training of LDDM (Jo et al., 2025). The last Transformer block's hidden activations, before the output head, are passed through a learned LayerNorm and added to the next call's token embeddings. The LayerNorm scale and bias are initialized to zero. Writing the denoiser and its hidden output as F_θ, inference uses

$$(p_k, h_{k+1}) = F_\theta(Z_k, c, h_k), \qquad h_0 = 0. \tag{35}$$

Each call supplies both the token prediction p_k for (17) and the next hidden state. During training, with probability 0.9 per microbatch, a precursor call on the same corrupted input with zero hidden state supplies a detached latent to the loss-bearing call. Otherwise, the latter uses zero hidden state. Inference carries the latent on every step, with no extra refinement calls. We retain our time-independent denoiser and weighted CE objective. The convergence theorem permits history-dependent proposals. As in the state-only case, its stopping guarantee requires validity recognition and uniform accessibility of accepted candidates. The no-SC checkpoint has no latent pathway.

Self-conditioning and the population target. For fixed precursor parameters, write its detached training latent as $H = g(Z, c, \xi)$, where ξ is any precursor randomness independent of X conditional on (Z, c). Then

$$\Pr(X_i = v \mid Z = z, C = c, H = h) = \Pr(X_i = v \mid Z = z, C = c).$$

Thus, on the training input law, minimizing conditional token CE over all categorical predictions has the same target marginals with or without this latent. This includes the zero-latent branch. Detaching H implements the two-pass training procedure, while the equality follows from using the same corrupted input and no additional target information. The statement concerns the population prediction target, not an identical optimization trajectory for finite neural networks. At inference the latent is carried from preceding states, as shown explicitly in Algorithm 3.

Program execution and scoring. Each training example concatenates a question, a separator, and a gold program defining `simple_math_problem`, including its docstring. Generated programs are executed in a sandbox with a 2s timeout for EqDLM, MDLM, and Duo evaluations. Timeouts and exceptions count as incorrect. The sandbox restricts imports to `math` and a small builtin whitelist. After converting near-integers to `int`, answers are compared exactly when both are integers, or with absolute tolerance 10^{-3} otherwise.

Population recognition condition. For $V = 49{,}153$ and $L = 400$, Proposition 2 guarantees population recognition for programs with conditional data probability above 5.1×10^{-8}, even with corruption between valid solutions. The invalid-state bound still applies, including under output-preserving corruption. This is a sufficient condition for the ideal denoiser, not an empirical estimate of individual program probabilities.

Corruption check. Each of 100,000 checks draws one $t \sim \mathrm{Unif}[0,1]$ and one corruption on the coordinates used during training. Of the source programs, 96,850 execute to a finite number. Within this subset, 309 corrupted programs return the same number: 138 are unchanged and 171 have a different token sequence. We compare with the source program's return value, rather than the GSM8K gold answer. This is an empirical check of output preservation, not a proof of the support condition.

Current protocol: stopping on every call. We use Algorithm 3 with $T_{\text{ann}} = 0$, $\gamma = 0.125$, $B = 128$, and $\tau = e^{-1}$. Every call checks the original joint probability of the current candidate before any temperature transformation or transition. Annealing uses the grid in (16), with no uniform noise in its last transition. Categorical continuation uses $T = 1$. The cap permits 128 continuation scoring calls and 127 updates. Thus the maximum cost is $N + 128$ calls. The SC latent is carried between all calls, with no extra refinement. For scoring, native log probabilities are normalized in FP64 and summed over generated positions, including EOS and padding but excluding the fixed question prefix. Categorical draws also use FP64. Both models use batches of 32, with a fixed permutation of all 1,319 problems and per-batch seeds. Execution has a 2s timeout.

Table 5: Current SC protocol with stopping checked in both phases. Accuracy includes budget fallbacks. Precision concerns accepted outputs only. No trajectory stopped during annealing in this evaluated grid.

Annealing N	Accuracy (%)	Mean NFE	Stopped (%)	Precision (%)
6	18.12	8.82	100.00	18.12
14	38.36	15.54	99.92	38.39
30	49.96	31.14	100.00	49.96
62	57.16	63.17	100.00	57.16
126	60.27	127.19	99.92	60.24
254	61.64	255.26	99.92	61.61
510	63.08	511.19	99.92	63.05
1022	63.61	1023.20	99.92	63.58

For $N = 62$, all 1,319 trajectories stop, including 754 correct programs. For $N = 1022$, 1,318 stop and 838 of these are correct (63.58% precision). The remaining trajectory returns a correct answer at the cap, giving $839/1319$ overall accuracy.

Table 6: Current no-SC protocol with stopping checked in both phases. All points evaluate 1,319 problems. Accuracy includes budget fallbacks. No trajectory stopped during annealing.

Annealing N	Accuracy (%)	Mean NFE	Stopped (%)	Precision (%)
6	12.28	9.09	99.85	12.30
14	22.06	16.01	100.00	22.06
30	29.34	31.59	99.92	29.36
62	33.13	63.32	100.00	33.13
126	37.30	127.15	100.00	37.30
254	41.09	255.15	100.00	41.09
510	43.97	511.15	100.00	43.97
1022	45.64	1023.08	100.00	45.64

Without SC, the current protocol yields $437/1319$ correct outputs at $N = 62$ and $602/1319$ at $N = 1022$. Every trajectory stops at these budgets, so stopping precision equals accuracy. Both SC settings now use the same all-phase stopping protocol.

The SC and no-SC curves use separately trained checkpoints. Their training precision and loss normalization differ, so the comparison does not isolate the effect of latent propagation. Figure 10 gives the no-SC comparison, complementing the SC curves in Figure 6.

Multiple attempts under a shared budget. For a task admitting many valid programs, a useful search procedure can trade the accuracy of one attempt for additional attempts within the same computational budget. We therefore compare the probability of obtaining at least one correct completed answer within a total budget $C \in \{128, 256, 512, 1024, 2048\}$ model calls per problem. Each attempt is charged its actual NFE. Failed attempts and calls spent on an unfinished final attempt also count, but only outputs completed within the budget can contribute a correct answer. Correctness is evaluated afterwards and never used to decide when to stop generating.

We use the self-conditioned 250k-step EMA EqDLM checkpoint with annealing at $T = 0, \gamma = 0.125$, followed by categorical sampling at $T = 1$. In this experiment, the current state's untempered probability is checked against e^{-1} on *every* call, including during annealing, and the first accepted

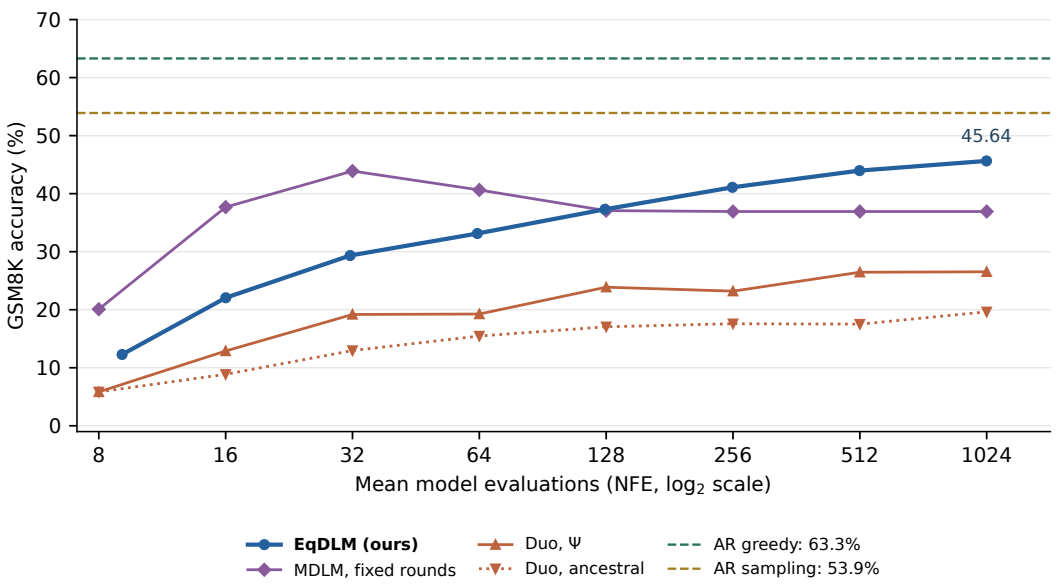

Figure 10: GSM8K generation without self-conditioning. EqDLM uses the current annealed categorical protocol, with stopping checked in both phases, as in the SC evaluation. Each point uses all 1,319 problems. EqDLM's NFE includes recognition and is averaged per sample. MDLM and Duo execute fixed budgets. The EqDLM grid is in Table 6, and baseline values are in Table 8.

state is returned. The categorical phase has a cap of 128 calls. Latent state is carried across phases and reset between attempts. For each total budget, we select the annealing length from $\{8, 16, 32, 64, 128\}$ using 256 GSM8K training problems, disjoint from the test set. All five budgets select 16 annealing updates. The calibration set is not guaranteed to be disjoint from TinyGSM training. AR sampling uses $T = 1$, and AR greedy uses $T = 0$. We evaluate the frozen choices on all 1,319 GSM8K test problems.

Table 7: Probability (%) of at least one correct completed GSM8K answer under a shared total NFE budget. All entries use our evaluations. The last column gives the paired difference between EqDLM and AR sampling in percentage points, with a 95% bootstrap interval.

Total NFE	**EqDLM (ours)**	AR sample	AR greedy	Difference (pp)
128	**70.81**	21.61	23.43	49.20 [46.32, 52.01]
256	**77.48**	55.42	61.94	22.06 [19.56, 24.64]
512	**82.87**	70.36	63.08	12.51 [10.31, 14.78]
1024	**85.97**	79.15	63.08	6.82 [4.93, 8.87]
2048	**89.31**	85.37	63.08	3.94 [2.35, 5.69]

EqDLM exceeds both AR decoders at every tested total budget (Table 7). Its advantage over AR sampling ranges from 49.20 percentage points at 128 calls to 3.94 at 2,048 calls. The intervals use 4,000 paired bootstrap resamples of test problems, conditional on the collected generations and frozen calibration choices. NFE counts per-sequence model calls, not FLOPs or wall-clock time. AR uses a KV cache, and batched EqDLM execution can perform additional forwards for rows that have already stopped.

This advantage concerns computational budget rather than a fixed number of attempts. Using the first 16 independent generations per problem to estimate ordinary pass@k, AR sampling is stronger at each tested $k \in \{1, 2, 4, 8, 16\}$. EqDLM attempts are substantially cheaper: their mean cost is 17.47 calls, versus 159.34 for AR sampling. For example, EqDLM achieves pass@$4 = 64.44\%$ at 69.87 mean total calls, whereas AR sampling achieves 75.89% at 637.37 calls. Using a fixed pool of 16 attempts for pass@k avoids a pool size that depends on how quickly generation completes. AR greedy is evaluated once per problem, since repeating it does not produce a different answer.

Updated single-attempt curve. Figure 11 reports the corresponding single-attempt evaluation with stopping checked in both phases. The annealing lengths are $6, 14, 30, 62, 126, 254, 510, 1022$, two updates below the respective nominal budgets $8, 16, 32, 64, 128, 256, 512, 1024$. We plot the measured total mean NFE, including categorical continuation and recognition. The 62-update run solves $754/1{,}319$ problems (57.16%) at 63.17 mean NFE, and the $1{,}022$-update run solves $839/1{,}319$ (63.61%) at $1{,}023.20$ mean NFE.

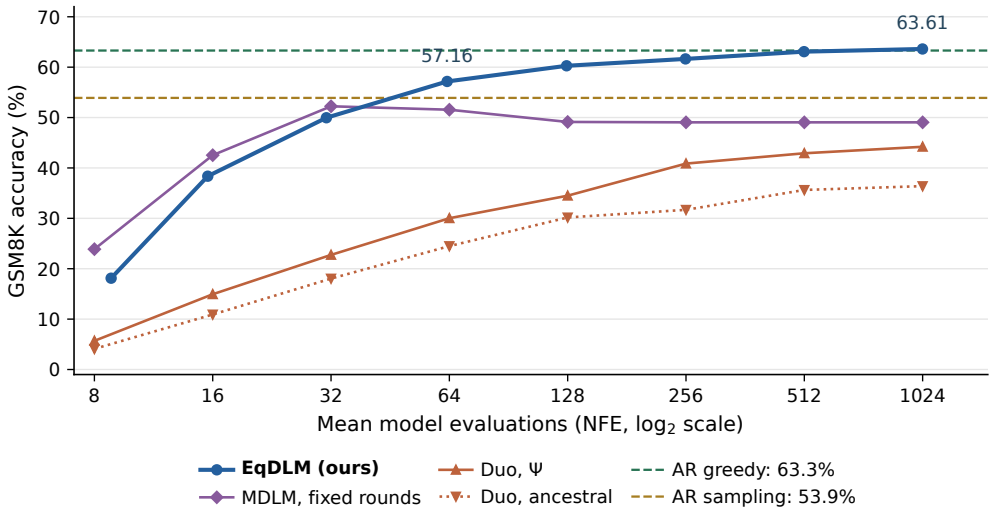

Figure 11: Updated GSM8K accuracy versus mean total NFE with self-conditioning. **EqDLM (ours)** uses annealed categorical sampling ($T = 0$, $\gamma = 0.125$), followed by categorical sampling at $T = 1$, with the e^{-1} stopping criterion checked in both phases. Each point evaluates all $1{,}319$ problems. The run with two annealing updates is omitted. Baseline curves retain the fixed-budget protocols of Figure 6, and dashed lines show published AR accuracies. Tick marks are powers of two, while EqDLM points are placed at their measured mean NFE.

MDLM fixed-round sampling. We evaluate the released S-FLM MDLM checkpoint without self-conditioning (Deschenaux & Gulcehre, 2026) and a separately trained 250k-step EMA checkpoint with self-conditioning. Both use the same fixed-round decoder. In each round, it fills at least $\lceil m/r \rceil$ of the most confident remaining positions, where m is the number of masks and r the number of rounds left, and also fills any position whose largest token probability is at least 0.999. Selected positions receive their most probable token and remain fixed. Every configured round is executed, including after all masks have been filled, so the reported NFE equals the fixed budget. Confidence-based iterative decoding was introduced for image tokens by MaskGIT (Chang et al., 2022). The threshold-plus-quota commitment rule follows FMLM+ posterior refinement (Agarwal et al., 2026). Here, confidence and token predictions come from MDLM's masked inputs and factorized marginals.

We use $T = 1$, native BF16 model computation, FP64 confidence probabilities, and disable TF32. Both variants use batches of 8, the same fixed permutation of all $1{,}319$ problems, and the same per-batch seeds as Duo below. With SC, the latent is carried once per call and reset for each batch. At 32 and 64 rounds, the SC model solves $689/1{,}319$ (52.24%) and $680/1{,}319$ (51.55%) problems, respectively. The SC grid covers 8–$1{,}024$ rounds. Without SC, the matched 32-round evaluation solves $579/1{,}319$ problems (43.90%), and 64 rounds solve $536/1{,}319$ (40.64%). Both complete grids span 8–$1{,}024$ rounds and peak at 32. Table 8 reports all measured values.

Duo sampling protocols. We evaluate the released S-FLM Duo checkpoint without SC (Deschenaux & Gulcehre, 2026) and a separately trained 250k-step Duo checkpoint with SC, using EMA weights in both cases. The ancestral sampler uses $T = 1$ without nucleus filtering. The Ψ sampler follows Deschenaux et al. (2026), using $T = 1$, top-$p = 0.9$, a linear time grid, and the released `max-rescale-0.05` setting throughout. Both include their final ancestral noise-removal call in the reported NFE. In particular, 64 NFE for Ψ means 63 reverse transitions and one final call. We retain the native model computation and output normalization, use FP64 categorical draws, disable TF32, and use batches of 8. The SC variant propagates its latent at every call and resets it between

batches. All comparisons use the same fixed permutation of the 1,319 problems and per-batch seeds. EqDLM's annealing rule is not used for either Duo variant.

Table 8: Complete MDLM and Duo GSM8K evaluations (accuracy in %). Every entry evaluates all 1,319 problems at exactly the listed NFE. MDLM uses fixed rounds with threshold 0.999. Duo uses ancestral or the recommended Ψ sampler. Dashes denote budgets not evaluated for MDLM.

	Without self-conditioning			With self-conditioning		
NFE	MDLM	Duo anc.	Duo Ψ	MDLM	Duo anc.	Duo Ψ
1	–	0.00	0.00	–	0.00	0.00
2	–	0.23	0.15	–	0.00	0.00
4	–	1.21	1.14	–	0.76	0.61
8	20.09	5.84	5.84	23.88	4.09	5.69
16	37.68	8.87	12.89	42.53	10.92	14.94
32	43.90	12.96	19.18	52.24	18.04	22.74
64	40.64	15.47	19.26	51.55	24.49	30.02
128	37.07	17.06	23.88	49.13	30.17	34.50
256	36.92	17.59	23.20	49.05	31.69	40.86
512	36.92	17.51	26.46	49.05	35.63	42.91
1024	36.92	19.64	26.54	49.05	36.39	44.20

Published FMLM+ variants. Table 9 reproduces the GSM8K results from Table 8 of Agarwal et al. (2026). The three FMLM+ variants all use posterior refinement at inference. The from-scratch variant uses no pretrained MDLM. The MDLM-init variant starts from pretrained MDLM weights. The MDLM-distill variant uses a frozen MDLM teacher to provide training targets. In the authors' released TinyGSM training scripts, only the Init variant enables MDLM weight initialization, while Distill uses teacher targets with probability 0.25 and leaves weight initialization disabled. Thus, these rows distinguish pretraining and teacher supervision from changes to the sampling procedure.

Categorical sampling with self-conditioning. We also evaluate direct categorical updates on the same SC checkpoint with $p = 1$ in the training weight $w(t) = (1-t)^p$, so $w(t) = 1-t$, retaining the latent between calls. At $T = 1$ and 64 calls, this gives $1/1319$ correct answers (0.08%). Accuracy is zero at 1, 2, 4, 8, 16, and 32 calls. In the saved 64-call trajectories, 55.27% of eighth updates and 90.75% of final updates leave the entire generated token sequence unchanged, consistent with rapid trapping in incorrect programs. This evaluation uses eight processes, batches of eight, and rank seeds 1 through 8, with the native BF16 model and FP64 categorical draws. The different batching and random streams make this a separate sampling evaluation rather than a paired comparison with the main annealed categorical sampling schedule study.

Generation without self-conditioning. To check whether repeated corruption also helps a denoiser without latent memory, we compare samplers on the same no-SC checkpoint at 64 calls (Table 10). These runs use FP16 model computation, FP64 categorical draws, batch size 8, seed 1, the same problem order, and EMA weights. At $T = 1$, changing from direct categorical updates to annealed categorical sampling increases the number of correct programs from 6 to 351. Taking greedy denoiser proposals within annealed categorical sampling increases this to 504. The no-SC annealed categorical sampling runs use the linear schedule and no early stopping. Differences in training precision, loss normalization, and evaluation protocol prevent attributing the gap to the main SC results solely to latent propagation.

G Discrete Swiss-roll experiment

State space and data. The state space is $\mathcal{X} = \{0, \dots, 30\}^2$. We rasterize an Archimedean spiral centered at $(15, 15)$ by evaluating

$$(15 + r(u)\cos\phi(u),\ 15 + r(u)\sin\phi(u)), \qquad r(u) = 1.5 + 12u, \quad \phi(u) = 0.35\pi + 4.5\pi u,$$

at 8,000 equally spaced values of $u \in [0, 1]$. We round to integer coordinates, connect successive points with Bresenham's line algorithm, and remove duplicates. This gives a support $\mathcal{S}$ of 139 states. Clean examples are uniform on $\mathcal{S}$.

Table 9: Published GSM8K accuracy (%) from Agarwal et al. (2026), Table 8. All FMLM+ variants use posterior refinement. MDLM init explicitly uses pretrained MDLM weights, while MDLM distill uses an MDLM teacher.

Method	1	2	4	8	16	32	64
FMLM+, PR (from scratch)	0.0	0.1	2.9	8.7	13.4	19.0	19.1
FMLM+, PR (MDLM distill)	0.3	0.4	3.9	10.3	18.7	21.6	23.4
FMLM+, PR (MDLM init)	0.3	0.7	5.1	15.1	26.1	31.8	33.6

Table 10: Sampler comparison without self-conditioning. All rows use the same 250k EMA checkpoint and 64 calls on the full GSM8K test set. Annealed categorical sampling uses a linear schedule.

Sampler	T	Correct / total	Accuracy
Categorical	1	6/1319	0.45%
annealed categorical sampling	1	351/1319	26.61%
annealed categorical sampling	0	504/1319	38.21%

Corruption. Uniform-token corruption is defined by

$$q_{\mathrm{raw}}(z \mid x) = \int_0^1 \prod_{i=1}^{2} \left[(1-t)\mathbf{1}\{z_i = x_i\} + \frac{t}{31} \right] dt. \tag{36}$$

Accepting proposals at other valid points with probability ρ gives

$$q_\rho(z \mid x) = \begin{cases} q_{\mathrm{raw}}(z \mid x), & z \notin \mathcal{S}, \\ \rho q_{\mathrm{raw}}(z \mid x), & z \in \mathcal{S} \setminus \{x\}, \\ q_{\mathrm{raw}}(x \mid x) + (1-\rho) \displaystyle\sum_{y \in \mathcal{S} \setminus \{x\}} q_{\mathrm{raw}}(y \mid x), & z = x. \end{cases} \tag{37}$$

The parameter ρ controls acceptance of proposals at a different valid point. It is not the frequency of such proposals. Averaged over clean points, the probability of corruption into another valid point is 0.1077415ρ. Every grid point has positive training-input probability at every tested ρ.

Model and objective. We use a bidirectional two-layer DiT with hidden dimension 64, four attention heads, conditioning dimension 32, adaptive layer normalization, and zero dropout. The input consists of the two coordinate tokens, each with vocabulary size 31. The time-conditioning input is fixed to zero, and self-conditioning is disabled. The output factorizes over the two coordinates. We minimize

$$\mathcal{L}(\theta) = -\frac{1}{2|\mathcal{S}|} \sum_{x \in \mathcal{S}} \sum_{z \in \mathcal{X}} q_\rho(z \mid x) \sum_{i=1}^{2} \log p_{\theta,i}(x_i \mid z).$$

All sums are evaluated exactly at each optimizer step. The implementation collects terms by corrupted input using the exact conditional coordinate marginals. This is the same denoising CE expectation, without an additional regularizer. We train in FP32 for 2,000 AdamW updates with zero weight decay, default momentum coefficients $(0.9, 0.999)$ and numerical constant 10^{-8}. The learning rate decreases linearly from 10^{-3} to approximately 10^{-4}.

Sampling and exact evaluation. We evaluate the network on all 961 inputs and construct the transition matrix

$$P_\theta(z, y) = \prod_{i=1}^{2} p_{\theta,i}(y_i \mid z).$$

The initial distribution μ_0 is uniform on the 822 off-support states. We propagate $\mu_{k+1} = \mu_k P_\theta$ for 64 updates, with temperature one. The model's logits are computed in FP32, then converted to FP64 for normalization and matrix propagation. The reported validity is $\sum_{z \in \mathcal{S}} \mu_{64}(z)$, computed by matrix propagation rather than estimated from sampled trajectories. The stay probability is the diagonal entry $P_\theta(z, z)$.

For Figure 3b, with $\rho = 0$, we draw 8,192 independent states from μ_{64} and plot their unique coordinates. All 139 valid states appear, and this particular sample contains no invalid states. The

distribution still assigns approximately 2.60×10^{-5} probability to invalid points. The minimum stay probability on valid states is 0.9998626, and the maximum on invalid states is 0.1339384. Averaged uniformly over valid states, the probability of moving outside the support in one update is 1.13×10^{-5}. Leaving a particular valid state and leaving the support are different events, since a transition to another valid point preserves validity.

Retention bound under cross-support corruption. For $z \in \mathcal{S}$, define the clean posterior mass on the input itself as

$$a_\rho(z) = \Pr(X = z \mid Z = z) = \frac{p_{\mathrm{data}}(z) q_\rho(z \mid z)}{\sum_{x \in \mathcal{S}} p_{\mathrm{data}}(x) q_\rho(z \mid x)}.$$

Each optimal coordinate marginal assigns at least $a_\rho(z)$ probability to z_i. Hence, for a factorized denoiser of length L,

$$p^\star(\mathcal{S} \mid z) \geq p^\star(z \mid z) \geq a_\rho(z)^L. \tag{38}$$

Together with the off-support upper bound, the sufficient condition $\min_{z \in \mathcal{S}} a_\rho(z)^L > e^{-1}$ guarantees perfect classification by the stay probability. In our two-token experiment, this lower bound is at least 0.49598 even at $\rho = 1$. The exact minimum stay probability is 0.65527, compared with 0.65519 for the learned model. The maximum stay probability on invalid inputs is 0.13369 for the exact denoiser at every ρ, and at most 0.13476 across the learned models. For $\rho = 1$, the minimum probability of remaining anywhere in the support is 0.79958 for the exact denoiser and 0.78959 for the learned model. All extrema are evaluated over the entire finite state space.

Generation with threshold stopping. We compare 64 categorical updates with a stopped process using the same transition matrix and initial distribution. Before each update, a point with $P_\theta(z, z) > e^{-1}$ is returned and receives no further updates. To evaluate this exactly, we replace its transition row by a point mass:

$$\widetilde{P}_\theta(z, y) = \begin{cases} \mathbf{1}\{y = z\}, & P_\theta(z, z) > e^{-1}, \\ P_\theta(z, y), & \text{otherwise.} \end{cases}$$

We then compare $\mu_0 P_\theta^{64}$ and $\mu_0 \widetilde{P}_\theta^{64}$. The threshold selects exactly the support for both denoisers at every tested ρ. The computed off-support mass after 64 updates is below 2×10^{-14} in every stopped run. This is a finite-budget numerical result, rather than exact zero failure probability. No threshold tuning or support oracle is used for stopping. The final state is returned when the update budget is exhausted.

For Figure 3e,f, we use the learned denoiser at $\rho = 1$ and draw 1,024 states from each final distribution, with and without threshold stopping. We plot unique coordinates, with marker sizes independent of frequency, coloring valid points blue and invalid points brown. The valid fractions are computed from the full distributions, not from these displayed samples. Without stopping, validity is 74.29%. With stopping, it is approximately 100%. These panels differ from Figure 3b, which uses 8,192 draws at $\rho = 0$.

H Related Work

Denoising-based generation. Repeated denoising is an established generative mechanism. Generalized denoising autoencoders alternate corruption and learned denoising to define a generative Markov chain (Bengio et al., 2013). SUNDAE trains iterative text generation with step-unrolled cross-entropy, including inputs produced by the model itself (Savinov et al., 2022). Idempotent Generative Networks learn fixed-point structure in continuous spaces through reconstruction and idempotence objectives (Shocher et al., 2024). We study a complementary property of ordinary denoising CE: validity recognition at the factorized population optimum. This yields recognition and convergence guarantees, whose empirical counterparts we examine in learned discrete models.

Validity recognition and stopping. Recognizing when to stop is a related challenge. SAMM (Kornhardt et al., 2026) adds confidence and stopping heads, and PRISM (Kim et al., 2025) learns per-token quality in the same forward pass through an additional objective. Learned signals of this kind also guide token selection, density correction, and halting (Lezama et al., 2022; Neklyudov et al., 2019; Korolev et al., 2026). Whether a local score certifies the full sequence depends on the task. Our whole-sequence threshold instead follows from ordinary denoising CE at its factorized population optimum. Support Before Frequency (Müller et al., 2026) separates support and frequency information in small-noise, single-token reverse edits. We study full-state recognition and the dynamics of repeated reconstruction.

Discrete diffusion language models. Discrete diffusion learns a reverse process to recover p_{data}. Conventional masked sampling commits revealed tokens, as in MDLM (Sahoo et al., 2024), while ReMDM allows revision through remasking (Wang et al., 2025). Uniform-state Duo (Sahoo et al., 2025) permits revision through ancestral or Ψ sampling (Deschenaux et al., 2026). RADD learns time-independent clean-data conditionals within an absorbing diffusion process (Ou et al., 2025). Time independence of the network alone therefore does not distinguish our approach. We instead establish conditions under which denoising CE yields validity recognition and repeated generation with stopping or absorption.

Continuous and hybrid language models. CANDI combines discrete and continuous corruption (Pynadath et al., 2025), while FLM and FMLM use continuous flows and flow maps, respectively (Lee et al., 2026). S-FLM learns flows on the hypersphere (Deschenaux & Gulcehre, 2026). FMLM+ extends flow maps with masking-style schedules and posterior refinement (Agarwal et al., 2026). These provide complementary routes to parallel generation. Our analysis concerns factorized predictions on discrete candidates and their relation to solution validity.