BINARY DISCRETE DIFFUSION: CANDIDATE-SET DIFFUSION FOR LOGIC AND PLANNING TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete diffusion models iteratively denoise a noisy sequence of categorical data such as text, enabling any-order generation with self-correction and in-filling capabilities. This makes them promising candidates for reasoning, planning, and constrained-generation tasks like coding and math. Yet masked and uniform diffusion erase a token's identity in a single corruption event, so their noisy states cannot explicitly represent partially resolved uncertainty in which several alternatives remain plausible. We close this gap with **Binary Discrete Diffusion**, a family of continuous-time discrete-diffusion processes whose noisy state is a set of candidate tokens per position, represented as a binary membership vector. Binary diffusion changes candidate sets through additions and removals, allowing information to be lost and recovered gradually while the diffusion state itself represents partial uncertainty. On Sudoku-Extreme, Boxoban, and graph coloring, binary symmetric diffusion consistently outperforms strong, well-tuned masked and uniform baselines under matched architectures and controlled training, sampling, and tuning budgets, with its largest gains on the hardest instances, while remaining competitive on language modeling (Text8, TinyStories, OpenWebText). These results support candidate sets as an effective intermediate representation for iterative reasoning and show that the corruption kernel can materially change the computation learned by the reverse model.[1]

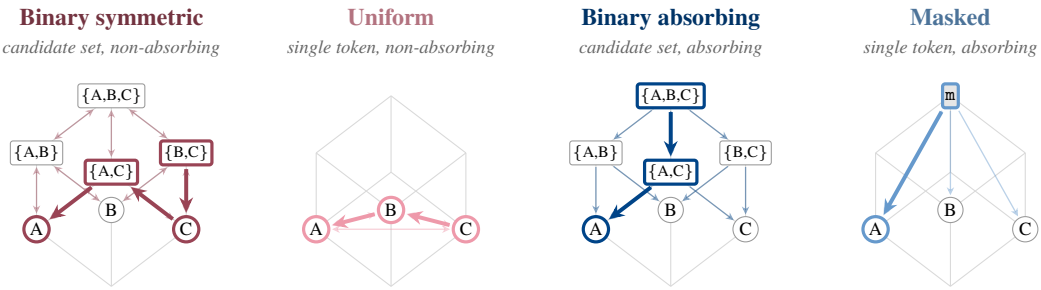


Figure 1: Denoising process of each kernel over the candidate sets of a three-token vocabulary {A,B,C}, with its allowed moves faded. Each panel highlights one possible denoising path to A.

1 INTRODUCTION

Discrete diffusion models generate by iteratively refining a corrupted sequence. Unlike autoregressive generation, decisions need not follow a fixed left-to-right order and can be revised later, while many positions can be updated in parallel (Schiff et al., 2025; Sahoo et al., 2024; Gong et al., 2025). These properties make discrete diffusion attractive for reasoning, planning, and constrained generation (Ye et al., 2025; Wewer et al., 2025). Yet sample quality at finite inference budgets remains challenging, motivating work on ordering, remasking, planned denoising, and search (Kim et al., 2025; Wang et al., 2025b; Liu et al., 2025; Bai et al., 2026). We instead focus on a complementary axis: the corruption process itself. How much can be gained by changing the corruption process, and in particular the intermediate representation on which denoising operates? Logic and planning are

[1] Code will be released upon publication.

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.

useful testbeds: they impose global constraints, admit exact solve-rate evaluation, and scale cleanly in difficulty at manageable computational cost, while sharing structure with coding, mathematics, and biological design (Sahoo et al., 2026; Avdeyev et al., 2023; Stark et al., 2024). They also expose a mismatch in the dominant masked and uniform kernels (Sahoo et al., 2024; Shi et al., 2024; Schiff et al., 2025; Sahoo et al., 2025): a position cannot naturally express that several alternatives remain plausible. A model solving Sudoku, for example, might narrow a cell from $\{2, 7, 9\}$ to $\{7, 9\}$, later reopen 2 as neighboring decisions are revised, and eventually settle on 7.

We propose to make such candidate sets themselves the diffusion states. A clean token is a singleton, while during reverse generation intermediate states can carry several live hypotheses that may be eliminated or restored. More broadly, the corruption kernel does more than specify how examples are noised: it determines what information survives at intermediate times and how the model's prediction is translated into reverse moves and their rates. Identical architectures can therefore induce qualitatively different iterative computations under different kernels. We instantiate this idea with **Binary Discrete Diffusion**, representing candidate sets by binary membership vectors in $\{0, 1\}^V$ (Figure 1). In **binary symmetric diffusion**, membership bits can switch both on and off, allowing hypotheses to be eliminated and later restored. In **binary absorbing diffusion**, corruption only adds candidates, so reverse generation progressively prunes the set toward a singleton.

Although each position has a nominal state space of size 2^V, every state has only V single-bit neighbors, and all neighboring reverse rates can be computed in $O(V)$ time from the model's usual V-way prediction, without enumerating subsets. The model learns these dynamics from clean solutions without candidate labels, task-specific deduction rules, or symbolic search. The richer state also enables **candidate-aware sampling**: we derive exact ancestral updates through a shared clean-token proposal, adapt re-corrupt sampling, and, building on ψ-sampling and confidence-guided remasking (Deschenaux et al., 2026; Wang et al., 2025b), develop revision rules that choose both *which positions* to reconsider and *which hypotheses* to reopen.

We evaluate Binary Discrete Diffusion on Sudoku-Extreme (Wang et al., 2025a), Boxoban (Guez et al., 2018), and graph coloring under controlled architectures, training budgets, and sampler tuning, using direct single-attempt solutions without verifier-guided selection. Across all three tasks, a consistent pattern emerges: **binary symmetric diffusion outperforms uniform diffusion, and both outperform masked and binary absorbing diffusion**. The advantage of binary symmetric over uniform grows on harder regimes, including low-clue Sudoku, long-horizon Boxoban, and graph coloring far beyond the training size. On Text8, TinyStories, and OpenWebText (Mahoney, 2011; Eldan & Li, 2023; Gokaslan & Cohen, 2019), binary diffusion remains competitive with established kernels. These results support candidate sets as an effective intermediate representation for iterative reasoning and show that the corruption process can materially change the computation learned by the reverse model.

2 Discrete diffusion modeling

We consider a data distribution p_{data} on sequences of length L over a finite vocabulary $[V] = \{1, \dots, V\}$. Starting from $Z_0 \sim p_{\mathrm{data}}$, we corrupt its positions independently with a continuous-time Markov chain (CTMC) $Z_t = (Z_t^1, \dots, Z_t^L)$, whose distribution approaches an easily sampled noise distribution as $t \uparrow 1$. We write $q_{t|s}(y \mid z)$ for the one-position transition kernel. For masked and uniform diffusion we use (Austin et al., 2021)

$$q_{t|0}(y \mid c) = \begin{cases} (1-t)\mathbf{1}\{y = c\} + t\mathbf{1}\{y = \mathtt{m}\}, & \text{masked,} \\ (1-t)\mathbf{1}\{y = c\} + t/V, & \text{uniform.} \end{cases} \tag{1}$$

Thus Z_t can be sampled directly at any time t, without simulating the intervening jumps.

We write $Q_t(z, y)$ for the forward jump rate from z to y, so that $q_{t+h|t}(y \mid z) = \mathbf{1}\{y = z\} + hQ_t(z, y) + o(h)$ as $h \downarrow 0$. The reverse rates depend on p_{data}, but conditioning on a clean example makes them computable from the forward process (Campbell et al., 2022; Noguerales et al., 2026). For a change at position i from Z_t^i to $y \neq Z_t^i$, the clean-conditioned reverse rates are given by

$$a_i(Z_0, Z_t, y; t) \stackrel{\text{def}}{=} Q_t(y, Z_t^i) \frac{q_{t|0}(y \mid Z_0^i)}{q_{t|0}(Z_t^i \mid Z_0^i)}. \tag{2}$$

The true reverse rates are obtained by averaging these clean-conditioned rates over $q(Z_0 \mid Z_t)$. Let $a_i^\theta(Z_t, y; t)$ denote learned reverse rates that depend on the noisy sequence Z_t and time t but not on Z_0. We train them against the clean-conditioned targets by minimizing the negative of the continuous-time evidence lower bound (ELBO) (Campbell et al., 2022; Noguerales et al., 2026):

$$
\begin{aligned}
-\operatorname{ELBO}(\theta) &= \int_0^1 \mathbb{E}_{Z_0, Z_t}[\mathcal{J}_t(\theta; Z_0, Z_t)]\, dt, \\
\mathcal{J}_t(\theta; Z_0, Z_t) &= \sum_{i=1}^{L} \sum_{y \neq Z_t^i} \Phi(a_i, a_i^\theta), \qquad \Phi(a, b) = a \log \frac{a}{b} - a + b.
\end{aligned}
\tag{3}
$$

This upper-bounds the expected negative log-likelihood. More sharply, its excess above the data-entropy floor is exactly the path-space KL between the exact reverse process and the learned reverse process (Noguerales et al., 2026). In practice, the jump rates can become singular near $t = 0$ or $t = 1$, so we train on an interior window $0 < t_1 < t_2 < 1$. The corresponding finite-window objective includes reconstruction and terminal-prior boundary terms, given in App. D.1.

Training. The network takes (Z_t, t) as input and outputs a strictly positive distribution $\mu_i^\theta(c, Z_t; t)$ over clean tokens c at each position i. Averaging the forward kernel over this prediction gives

$$
m_i^\theta(y, Z_t; t) \stackrel{\text{def}}{=} \sum_{c=1}^{V} \mu_i^\theta(c, Z_t; t) q_{t|0}(y \mid c), \tag{4}
$$

which we call the *local noisy-state mass*, and the model reverse rate is

$$
a_i^\theta(Z_t, y; t) \stackrel{\text{def}}{=} Q_t(y, Z_t^i) \frac{m_i^\theta(y, Z_t; t)}{m_i^\theta(Z_t^i, Z_t; t)}, \qquad y \neq Z_t^i. \tag{5}
$$

Thus one V-way network output determines all allowed reverse moves at a position. Training then minimizes an empirical estimate of the path integral in (3) by sampling (t, Z_0, Z_t).

Remark 2.1. This is the *cavity parameterization* (Noguerales et al., 2026; Sun et al., 2023). It minimizes the negative ELBO when $\mu_i^\theta(c, Z_t; t) = q(Z_0^i = c \mid Z_t^{-i})$, where Z_t^{-i} denotes the full sequence Z_t excluding Z_t^i, even though the network itself is evaluated on the full sequence Z_t.

3 Binary discrete diffusion

In binary diffusion, the noisy state at each position is a *set of candidate tokens*. A clean token $c \in [V]$ is represented by the singleton $\{c\}$, and corruption changes this set by adding or removing candidates. These sets are the diffusion states, allowing an intermediate state to retain several alternatives at once. Representing each set by its binary membership vector places the noisy state Z_t^i in the binary cube $\{0, 1\}^V$, where we use the same symbol for a noisy set z and its binary membership vector, including for a clean singleton $\{c\}$ and its one-hot vector.

We consider two corruption processes on this space. In **binary symmetric diffusion**, membership bits can switch both on and off: as t increases, the original token becomes less likely to be present and the other tokens more likely, until the state approaches independent fair bits and the original token is forgotten. In **binary absorbing diffusion**, corruption only adds candidates while preserving the original token, and the state approaches the full vocabulary. We focus primarily on the symmetric process, whose reversible candidate dynamics allow hypotheses to be both discarded and restored, and treat the absorbing process as a simpler monotone counterpart.

Although the forward bits evolve independently conditional on the clean sequence, the learned reverse dynamics depend on the full noisy sequence. Crucially, the network still produces only the usual V-way distribution $\mu_i^\theta(c, Z_t; t)$ over clean tokens at each position. The construction of Section 2 converts this single prediction into all allowed reverse rates on the binary state space.

3.1 Binary symmetric diffusion

In binary symmetric diffusion, each membership bit retains its original value with probability $1 - t$ and is otherwise resampled uniformly from $\{0, 1\}$. Conditional on Z_0, the process on $(\{0, 1\}^V)^L$

factorizes across positions and bits, and it suffices to specify the one-position kernel. For times $0 \le s < t < 1$ and noisy states $z, y \in \{0,1\}^V$, the forward kernel is given by

$$q^{\mathrm{sym}}_{t|s}(y \mid z) = (1-\varepsilon_{s,t})^{V-d_H(y,z)}\varepsilon_{s,t}^{d_H(y,z)}, \qquad \varepsilon_{s,t} \overset{\text{def}}{=} \frac{t-s}{2(1-s)}. \tag{6}$$

Here $d_H(y,z)$ is the Hamming distance. Equivalently, each bit is resampled with probability $(t-s)/(1-s)$, so it differs from its value at time s with probability $\varepsilon_{s,t}$. In particular, relative to the clean state Z_0^i, the mismatch probability for a given bit is $t/2$.

Forward rates. Differentiating (6) gives

$$Q^{\mathrm{sym}}_t(z,y) = \frac{\mathbf{1}\{d_H(z,y)=1\}}{2(1-t)}, \qquad y \neq z. \tag{7}$$

Writing $z^{(u)}$ for z with bit u flipped, the chain moves along the edges of the V-dimensional binary cube: from z its only possible destinations are the V Hamming neighbors $z^{(1)}, \dots, z^{(V)}$.

Clean-conditioned reverse rates. The reverse process goes backward through the same Hamming-neighbor graph as the forward process, so each reverse move also flips one bit. Relative to the original u-th bit, $(Z_0^i)_u$, such a flip either changes a mismatch into a match or a match into a mismatch. Define the match-to-mismatch odds ratio

$$\rho_t \overset{\text{def}}{=} \frac{1-t/2}{t/2} = \frac{2-t}{t}. \tag{8}$$

Then, the reverse rate for flipping bit u at position i is

$$a^{\mathrm{sym}}_{i,u}(Z_0, Z_t; t) \overset{\text{def}}{=} \frac{1}{2(1-t)} \begin{cases} \rho_t, & (Z_t^i)_u \neq (Z_0^i)_u, \\ \rho_t^{-1}, & (Z_t^i)_u = (Z_0^i)_u. \end{cases} \tag{9}$$

Interpretation. Conditioned on Z_0, a flip that makes $(Z_t^i)_u$ agree with $(Z_0^i)_u$ is favored over a flip that makes them disagree by a factor of ρ_t^2. **This preference strengthens as $t \downarrow 0$ and vanishes as $t \uparrow 1$, while both flip directions remain possible for $0 < t < 1$.**

Model reverse rates. Define the network probability mass assigned to a candidate set z by

$$\nu_i^\theta(z, Z_t; t) \overset{\text{def}}{=} \sum_{c \in z} \mu_i^\theta(c, Z_t; t). \tag{10}$$

For fixed z, the symmetric forward likelihood depends on the clean token only through whether it belongs to z. Substituting this structure into the general reverse-rate formula (5) gives, for $z = Z_t^i$ and $z^{(u)}$ obtained by flipping bit u,

$$a^{\theta,\mathrm{sym}}_{i,u}(Z_t;t) = \frac{\rho_t^{2z_u-1}}{2(1-t)} \frac{1+(\rho_t^2-1)\nu_i^\theta(z^{(u)}, Z_t; t)}{1+(\rho_t^2-1)\nu_i^\theta(z, Z_t; t)}, \qquad z_u = \mathbf{1}\{u \in z\}. \tag{11}$$

The full derivation is given in App. B.1.

Interpretation. Let $s = \nu_i^\theta(z, Z_t; t)$, $p_u = \mu_i^\theta(u, Z_t; t)$, and $A_t = \rho_t^2 - 1$. Flipping u changes the candidate-set mass to $s + (1-2z_u)p_u$, so the p_u-dependent factor in (11) is

$$\frac{1+A_t\nu_i^\theta(z^{(u)}, Z_t; t)}{1+A_t s} = \begin{cases} 1 - \dfrac{A_t p_u}{1+A_t s}, & u \in z, \\ 1 + \dfrac{A_t p_u}{1+A_t s}, & u \notin z. \end{cases}$$

Thus candidates assigned higher probability by the model are harder to remove when present and faster to restore when absent.

Efficient computation. Each binary state has only V Hamming neighbors. After computing $\nu_i^\theta(z, Z_t; t)$ once, all neighboring masses follow from $\nu_i^\theta(z^{(u)}, Z_t; t) = \nu_i^\theta(z, Z_t; t) + (1 - 2z_u)\mu_i^\theta(u, Z_t; t)$, so all reverse rates are obtained in $O(V)$ time without enumerating the 2^V states. The path loss similarly sums only over these V neighbors: $\mathcal{J}_t^{\mathrm{sym}} = \sum_i \sum_{u=1}^V \Phi(a_{i,u}^{\mathrm{sym}}, a_{i,u}^{\theta,\mathrm{sym}})$.

3.2 Binary absorbing diffusion

Binary absorbing diffusion only adds candidates. Between $s < t$, each inactive bit is activated independently with probability $(t-s)/(1-s)$, while already active bits never flip. Thus the original token is always present, every other token is present independently with probability t at time t, and the state approaches the full vocabulary as $t \uparrow 1$. For $z, y \in \{0,1\}^V$, the transition kernel is

$$q_{t|s}^{\mathrm{abs}}(y \mid z) = \mathbf{1}\{z \subseteq y\} \left(\frac{1-t}{1-s}\right)^{V-|y|} \left(\frac{t-s}{1-s}\right)^{|y|-|z|}. \tag{12}$$

Forward and reverse rates. The only forward jumps add one candidate, each at rate $1/(1-t)$. The reverse process therefore only removes active candidates, with clean-conditioned rate

$$a_{i,u}^{\mathrm{abs}}(Z_0, Z_t; t) = \frac{\mathbf{1}\{u \in Z_t^i \setminus Z_0^i\}}{t}. \tag{13}$$

Thus every candidate other than the original clean token is removed at rate $1/t$, while the original token has zero removal rate.

Model reverse rates. For fixed z, the forward kernel $q_{t|0}^{\mathrm{abs}}(z \mid c)$ is constant for $c \in z$ and zero for $c \notin z$. Using the candidate-set mass ν_i^θ from (10), the general reverse-rate formula (5) reduces to

$$a_{i,u}^{\theta,\mathrm{abs}}(Z_t; t) = \frac{\mathbf{1}\{u \in Z_t^i\}}{t} \left(1 - \frac{\mu_i^\theta(u, Z_t; t)}{\nu_i^\theta(Z_t^i, Z_t; t)}\right). \tag{14}$$

Interpretation. A candidate receiving a larger share of the model probability within the current candidate set has a smaller removal rate. **Denoising therefore prunes unlikely candidates while protecting plausible ones.**

The absorbing path loss and the full derivations are given in App. B.2.

4 Sampling

Recall from Remark 2.1 that the network output μ_i^θ is a clean-token prediction in the cavity parameterization. Sampling instead needs the *denoiser*, $\pi_i^\theta(c, Z_t; t)$, the posterior of the clean token given the full noisy sequence, which reweights the cavity prediction by the local forward kernel:

$$\pi_i^\theta(c, Z_t; t) \overset{\text{def}}{=} \frac{\mu_i^\theta(c, Z_t; t)\, q_{t|0}(Z_t^i \mid c)}{\sum_{d=1}^V \mu_i^\theta(d, Z_t; t)\, q_{t|0}(Z_t^i \mid d)}. \tag{15}$$

For binary symmetric diffusion the reweighting multiplies the probability of each active candidate by ρ_t^2 from (8), and for binary absorbing it restricts the prediction to the active candidates. Generation starts from the limiting forward distribution, $p_{t_2}^{\mathrm{sym}}(z) = 2^{-V}$ for binary symmetric and $p_{t_2}^{\mathrm{abs}}(z) = \mathbf{1}\{z = \mathbb{1}_V\}$ for binary absorbing, where the all-ones vector $\mathbb{1}_V$ is the full vocabulary.

Exact ancestral sampling. Let $q_{s|t,0}(y \mid z, c)$ be the forward bridge, the distribution of the state at $s < t$ given the state z at t and the clean token c. The ancestral update averages this bridge over the denoiser:

$$p_{s|t,i}^\theta(y \mid Z_t) = \sum_{c=1}^V \pi_i^\theta(c, Z_t; t)\, q_{s|t,0}(y \mid Z_t^i, c). \tag{16}$$

A binary position has 2^V states, so this distribution cannot be enumerated for sampling at large V. As for the reverse rates in Section 3, the product structure of the kernels avoids this: given the

clean token, both bridges factor over bits. We therefore draw a clean proposal $\widehat{Z}_0^i \sim \pi_i^\theta(\cdot, Z_t; t)$ and then each bit from its one-bit bridge (App. B.3), which samples exactly from (16). Under the symmetric bridge each bit favors its current value for s close to t and the proposed clean bit as $s \downarrow 0$, so removed candidates can return. Under the absorbing bridge the proposal survives, every other active candidate survives with probability s/t, and a removed candidate never returns.

Revision by re-corruption. Ancestral sampling preserves the state accumulated by the chain, including its mistakes. *Re-corrupt* sampling instead deliberately forgets more of the current state: it draws a clean proposal from the denoiser and noises it afresh to level s,

$$p_{s|t,i}^{\theta,\text{re-corr}}(y \mid Z_t) = \sum_{c=1}^{V} \pi_i^\theta(c, Z_t; t)\, q_{s|0}(y \mid c). \tag{17}$$

If Z_t^i follows the forward marginal of a clean token, both ancestral and re-corrupt updates induce the correct forward marginal at level s, but re-corruption adds far more randomness and makes every position revisable at every step. This alone improves markedly over ancestral sampling on logic tasks (Table 10 in App. E.4).

Confidence-guided ψ-sampler. As in the ψ-sampler of Deschenaux et al. (2026), we mix the ancestral and re-corrupt samplers, but independently at each position i with weight κ_i:

$$\widehat{Z}_0^i \sim \pi_i^\theta(\cdot, Z_t; t), \qquad Z_s^i \sim (1-\kappa_i)\, q_{s|t,0}(\cdot \mid Z_t^i, \widehat{Z}_0^i) + \kappa_i\, q_{s|0}(\cdot \mid \widehat{Z}_0^i). \tag{18}$$

The bridge retains what the chain has inferred so far, while re-corrupting forgets it and allows errors to be corrected. A constant $\kappa_i = \eta$ interpolates from ancestral sampling at $\eta = 0$ to re-corrupt sampling at $\eta = 1$ and preserves the forward marginals, since both updates do. For masked diffusion, this mixture is the remasking sampler ReMDM (Wang et al., 2025b). We extend it with two confidence mechanisms, whose strengths γ, λ, and β recover the plain ψ-sampler at zero (Algorithm 1).

- **Position confidence: where to reconsider.** Following ReMDM-conf, we use the model's confidence in its clean proposal, read from the cavity or the denoiser, to concentrate revision on uncertain positions. We let κ_i decrease with confidence at strength γ, keeping its mean equal to η across positions. We also tilt the noise level s_i to which re-corrupted positions return with the same confidence at strength λ, keeping the mean noise level fixed (App. E.4).
- **Candidate confidence: which candidates to reconsider.** Plain re-corruption reopens alternative candidates uniformly at random, as the forward kernel does. For binary diffusion, we instead reopen candidate u with probability proportional to $p_{i,u}^\beta$, keeping the expected number of reopened candidates fixed, where $p_{i,u}$ is the denoiser probability of u when it was last in the candidate set (App. E.4). Uniform diffusion analogously draws the new token $u \neq \widehat{Z}_0^i$ in proportion to $\pi_i^\theta(u, Z_t; t)^\beta$; masked diffusion has no candidate to reopen. Thus candidates the model found plausible are preferentially restored.

Algorithm 1 Confidence-guided ψ-sampler.

Require: Cavity model, kernel, grid $t_2 = \tau_N > \cdots > \tau_0 = t_1$, re-corrupt fraction η, strengths γ, λ, β, proposal temperature T

1: Draw $Z_{t_2}^i \sim p_{t_2}$ at each position.
2: **for** $j = N, \ldots, 1$ **do**
3: Set $t = \tau_j$, $s = \tau_{j-1}$, evaluate $\mu_i^\theta(\cdot, Z_t; t)$, compute $\pi_i^\theta(\cdot, Z_t; t)$ by (15), and record $p_{i,u} = \pi_i^\theta(u, Z_t; t)$ for each candidate $u \in Z_t^i$ (every u for uniform diffusion).
4: Draw a clean proposal $\widehat{Z}_0^i$ from the denoiser at temperature T at each position.
5: Set $\kappa_i \propto \exp(-\gamma\, \mu_i^\theta(\widehat{Z}_0^i, Z_t; t))$ and $s_i \propto \exp(-\lambda\, \mu_i^\theta(\widehat{Z}_0^i, Z_t; t))$ with means η and s over positions, or with π_i^θ in place of μ_i^θ.
6: Draw Z_s^i from (18), re-corrupting to s_i and reopening candidates u with probability $\propto p_{i,u}^\beta$.
7: **end for**
8: Evaluate $\mu_i^\theta(\cdot, Z_{t_1}; t_1)$ and decode each position with $\pi_i^\theta(\cdot, Z_{t_1}; t_1)$.

5 EXPERIMENTS

We focus on solve rate and sample quality and report negative ELBOs only for completeness: the negative ELBO measures the path-space KL to the exact reverse process (Noguerales et al., 2026), but it need not track the quality of the final samples, especially under non-ancestral samplers, as uniform versus masked diffusion shows (Schiff et al., 2025; Sahoo et al., 2025).

5.1 LOGIC AND PLANNING TASKS

We evaluate on Sudoku-Extreme (Wang et al., 2025a), Boxoban (Guez et al., 2018; Garriga-Alonso et al., 2024), and four-coloring of random graphs, using matched 43M-parameter diffusion transformers and five training seeds per kernel and task. Besides the full test sets, we report hard subsets to de-saturate the benchmarks: low-clue Sudoku, long-solution Boxoban, and graphs ten times the training size. A sample counts as solved only if it satisfies every constraint, with one attempt per instance and no verifier. For the full and hard settings separately, each kernel's sampler is selected on validation at a reference budget of 256 network evaluations (NFE) for Sudoku and Boxoban and 64 for graph coloring, with the same tuning budget for every kernel and candidate confidence for all but masked diffusion. Parentheses below are standard errors over the five training seeds. Full experimental details are given in App. E, and the test negative ELBOs in Table 6.

Table 1: Solve rate (%) on the full test sets and hard subsets at the reference NFE.

	Sudoku		Boxoban		Graph coloring	
Kernel	All	Hard	All	Hard	All	Hard
Binary symmetric	**96.27** (0.21)	**92.66** (0.39)	**94.29** (0.30)	**84.65** (0.56)	**99.35** (0.00)	**64.13** (1.10)
Uniform	95.47 (0.21)	89.81 (0.47)	92.38 (0.64)	78.73 (1.20)	99.01 (0.02)	61.41 (0.34)
Binary absorbing	91.24 (0.11)	81.08 (0.26)	87.30 (0.81)	69.13 (1.66)	98.83 (0.02)	54.77 (0.39)
Masked	87.96 (0.16)	76.12 (0.10)	91.26 (0.31)	75.47 (0.69)	97.44 (0.04)	4.85 (0.33)

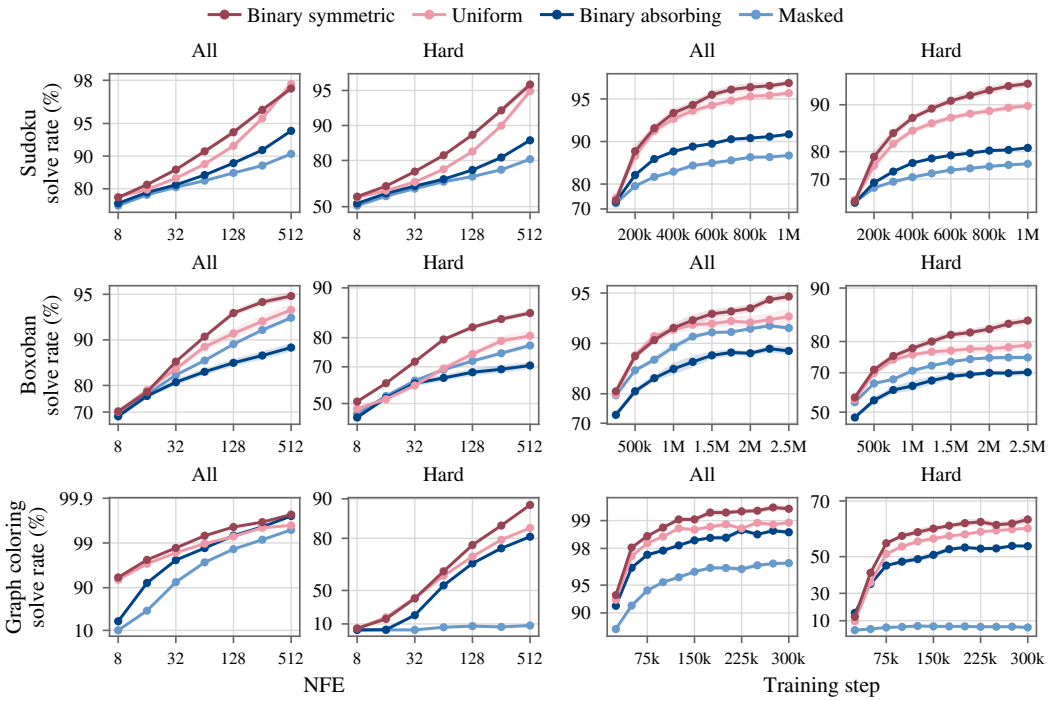


Figure 2: Solve rate against inference budget (NFE, left) and training step (right).

Table 1 reports the solve rates at the reference NFE, while Figure 2 shows their dependence on inference budget and training progress. Binary symmetric diffusion is the strongest kernel on every task, followed by uniform diffusion, with masked and binary absorbing diffusion behind both. Its

margin widens on the hard subsets, especially on long-horizon Boxoban and on large graphs. On the large graphs, masked diffusion nearly fails. Binary symmetric diffusion leads at almost all budgets and training steps (Figure 2), rather than depending on a single NFE or checkpoint. Table 10 ablates the sampling components, while Figure 4 in App. E.5 shows that the kernel comparison persists across difficulty bins and graph sizes. We show how each kernel solves a Sudoku board in App. F.

5.2 TEXT

We compare the four kernels using unconditional generation on three corpora of increasing sequence length L: Text8 (Mahoney, 2011) at $L = 256$, TinyStories (Eldan & Li, 2023) at $L = 512$, and OpenWebText (Gokaslan & Cohen, 2019) at $L = 1024$ (App. E.6). Every model trains on 131B tokens, on Text8 at the character level with $V = 27$ and on TinyStories and OpenWebText with byte-pair-encoding tokenizers of $V = 1024$. We report MAUVE (Pillutla et al., 2021) against GPT-2 token unigram entropy, with GPT-2 Large (Radford et al., 2019) supplying the features. MAUVE compares the distribution of generated samples with held-out text, and we use it as the primary metric given known failure modes of generative perplexity (GenPPL) (Franca & Tong, 2026; Zheng et al., 2025b), which we report with the negative ELBOs in App. E.6.

Table 2 gives the test MAUVE of each kernel at the sampler setting selected on validation (App. E.4), while Figure 3 traces quality against diversity using sampling settings selected on validation at representative entropies and evaluated on the test split. No kernel leads on all three corpora: binary symmetric diffusion reaches the best MAUVE on Text8, uniform diffusion on TinyStories, and masked diffusion on OpenWebText, each by a small margin. The kernels peak at slightly different entropies, so Table 12 in App. E.6 also compares them at one common entropy per corpus, where binary symmetric diffusion reaches the best MAUVE on both Text8 and TinyStories.

Table 2: Test MAUVE of tuned samplers (standard error over two generation seeds).

Dataset	Binary symmetric	Uniform	Binary absorbing	Masked
Text8	**0.913** (0.001)	0.896 (0.006)	0.821 (0.000)	0.854 (0.008)
TinyStories	0.882 (0.006)	**0.904** (0.001)	0.585 (0.011)	0.850 (0.009)
OpenWebText	0.840 (0.001)	0.822 (0.004)	0.747 (0.004)	**0.871** (0.013)

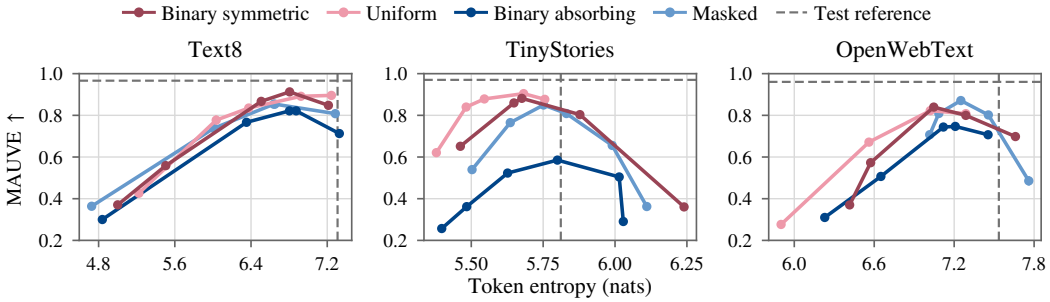


Figure 3: MAUVE against token entropy with GPT-2 Large as the scorer. Each point's sampling configuration is selected on validation at a representative entropy and evaluated on the test split.

5.3 TRAINING BINARY DIFFUSION MODELS

Binary models share the architecture and objective of the masked and uniform baselines, training on the path term of the negative ELBO (3), with two exceptions on Text8 described in App. D.2. A candidate set is represented by the sum of its token embeddings divided by the square root of its size, so a singleton is embedded as an ordinary token (and the empty set as zero). One consideration when scaling to larger vocabularies is whether this simple pooled representation remains sufficiently expressive as candidate sets grow.

The main optimization change is the time-sampling schedule. Binary diffusion corrupts all V membership bits in parallel, so the candidate set leaves the singleton regime already at $t = O(1/V)$. A linear schedule therefore spends too few samples near the clean endpoint, an effect that becomes

more pronounced as V grows. Switching to a logarithmic schedule was the single largest optimization improvement and was essential at large vocabularies. The network receives the corresponding normalized schedule coordinate as its time input (App. D.3). We further refine each base schedule with gradient-RMS importance sampling for all kernels (App. D.4).

A separate issue concerns the evaluation of the negative ELBO. For a finite training window, the path term alone is not comparable across kernels or endpoints: substantial information can remain in the state at t_2, making the truncated negative ELBO appear spuriously small. Meaningful comparisons therefore require the reconstruction and terminal-prior boundary terms of (50).

6 Related work

Discrete diffusion was first formulated on a discrete time grid, modeling the reverse transitions directly (Hoogeboom et al., 2021; Austin et al., 2021), and its continuous-time formulation instead learns to reverse a categorical jump process (Campbell et al., 2022; Sun et al., 2023). Since then, methods have differed mainly in how they parameterize the reverse rates (Lou et al., 2024; Ou et al., 2025) and in which paths the forward process allows (von Rütte et al., 2025; Shaul et al., 2025). Closer to our setting, simplex-valued methods represent categorical uncertainty in a continuous state (Floto et al., 2023; Avdeyev et al., 2023), while partial and semantic masking keep residual information about the clean token (Chao et al., 2025; Zhou et al., 2025). Binary diffusion instead changes the state space: each position follows a jump process on the binary cube, whose coordinates express candidate membership. Similar bit-flipping processes have been explored to model binary data, where the cube is the data space (Sohl-Dickstein et al., 2015; Pham et al., 2025). The closest predecessor is the ShortListing Model, which also grows candidate sets, in discrete time (Song et al., 2025). Under the right noise schedule, its corruption process provides a discrete-time formulation of binary absorbing diffusion (App. C). Our main contribution is the non-absorbing symmetric process, which allows candidates to be both removed and restored during denoising, and the empirical evaluation on logic and planning tasks.

Discrete diffusion has been applied to Sudoku and other structured problems before (Ye et al., 2025; Wewer et al., 2025), and follow-up work improves solving through adaptive token order, remasking, explicit planning, or test-time search (Kim et al., 2025; Wang et al., 2025b; Liu et al., 2025; Zhang et al., 2026). Sudoku is also a standard benchmark for small recurrent reasoners such as HRM and TRM (Wang et al., 2025a; Jolicoeur-Martineau, 2025), and Lattice Deduction Transformers likewise maintain per-variable candidate sets but train deduction on-policy with search and abstention heuristics (Davis et al., 2026). All of these change the sampler, the architecture, or the training signal. Our experiments are instead designed to isolate the effect of the corruption kernel and evaluate direct, verifier-free samples from each kernel's tuned sampler.

7 Conclusion

We have introduced Binary Discrete Diffusion, a family of continuous-time discrete-diffusion processes whose noisy states are sets of candidate tokens, represented as binary membership vectors. Parameterizing the reverse process through clean-token predictions lets a single V-way output determine all reverse rates and ancestral updates, avoiding any 2^V enumeration in training and sampling. We also introduce a confidence-guided self-correcting sampler that chooses which positions and candidates to reopen. Under controlled architectures, training budgets, and sampler tuning, binary symmetric diffusion outperforms uniform diffusion, and both outperform masked and binary absorbing diffusion on logic and planning tasks, with the largest gains on the hardest instances. Binary diffusion also remains competitive on language modeling. These results show that candidate sets can be an effective intermediate representation for iterative reasoning, and that the corruption process can materially change the computation learned by the model.

Binary diffusion opens a broader design space for candidate-set corruption processes. Future kernels could exploit semantic or task-specific structure to introduce plausible candidates rather than random ones. Richer representations of candidate sets than the normalized embedding sum used here may also prove useful at larger scales. Scaling these ideas to larger models and datasets, and to domains such as code, mathematics, and biological sequence design, is a natural next step.

AI USE STATEMENT

Large language models were used to aid and polish writing (editing and reviewing the manuscript), for retrieval and discovery (finding and summarizing related work), and for research ideation and execution (implementing, running, and analyzing experiments). All AI-assisted work was reviewed by the authors, who verified every technical claim and result and take full responsibility for the final content.

ETHICS STATEMENT

This methodological work introduces candidate-set corruption processes for discrete diffusion and evaluates them with small research models trained on logic and planning tasks and on public text corpora, including OpenWebText, a web corpus with known biases. None of these models is intended for deployment, and the usual considerations for language-model research apply.

REPRODUCIBILITY STATEMENT

App. E specifies the datasets with their splits and hard subsets, the architectures and training settings, every evaluation procedure, and the selection of learning rates, checkpoints, and sampler settings. App. D describes the training objectives and time sampling, and Apps. A and B derive the rates, losses, and ancestral updates of every kernel. The code used for the experiments will be released with the paper.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021. URL https://openreview.net/forum?id=h7-XixPCAL.

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1276–1301. PMLR, 2023. URL https://proceedings.mlr.press/v202/avdeyev23a.html.

Jinbin Bai, Yixuan Li, Yuchen Zhu, Yi Xin, Qingyu Shi, Aosong Feng, Xiaohong Liu, Molei Tao, Jianru Xue, Xiangtai Li, and Ming-Hsuan Yang. Prism: Efficient test-time scaling via hierarchical search and self-verification for discrete diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.01842.

Chris Cameron, Wangzheng Wang, Nikita Ivanov, Ashmita Bhattacharyya, Didier Chételat, and Yingxue Zhang. One step forward and k steps back: Better reasoning with denoising recursion models, 2026. URL https://arxiv.org/abs/2604.18839.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas 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://openreview.net/forum?id=DmT862YAieY.

Nuria Chandra, Yucen Li, Alan Amin, Alex Ali, Joshua Rollins, Sebastian W. Ober, Aniruddh Raghu, and Andrew Gordon Wilson. A unification of discrete, Gaussian, and simplicial diffusion. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/45d5823950cab9a88e10c6e8dbff351c-Abstract-Conference.html.

Chen-Hao Chao, Wei-Fang Sun, Hanwen Liang, Chun-Yi Lee, and Rahul G. Krishnan. Beyond masked and unmasked: Discrete diffusion models via partial masking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=nqpbbvEZwF.

Tao Chen, Chenhui Wang, and Hongming Shan. BerDiff: Conditional Bernoulli diffusion model for medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 14223 of *Lecture Notes in Computer Science*, pp. 491–501. Springer, 2023a. doi: 10.1007/978-3-031-43901-8_47. URL https://doi.org/10.1007/978-3-031-43901-8_47.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations (ICLR)*, 2023b. URL https://openreview.net/forum?id=3itjR9QxFw.

Liam Davis, Leopold Haller, Alberto Alfarano, and Mark Santolucito. Lattice deduction transformers, 2026. URL https://arxiv.org/abs/2605.08605.

Justin Deschenaux, Caglar Gulcehre, and Subham Sekhar Sahoo. The diffusion duality, chapter II: Ψ-samplers and efficient curriculum. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2602.21185.

Ronen Eldan and Yuanzhi Li. TinyStories: How small can language models be and still speak coherent English?, 2023. URL https://arxiv.org/abs/2305.07759.

Jacob Fein-Ashley and Paria Rashidinejad. Solve the loop: Attractor models for language and reasoning, 2026. URL https://arxiv.org/abs/2605.12466.

Griffin Floto, Thorsteinn Jonsson, Mihai Nica, Scott Sanner, and Eric Zhengyu Zhu. Diffusion on the probability simplex, 2023. URL https://arxiv.org/abs/2309.02530.

Antonio Franca and Alexander Tong. Hacking generative perplexity: Why unconditional text evaluation needs distributional metrics, 2026. URL https://arxiv.org/abs/2606.08417.

Adrià Garriga-Alonso, Mohammad Taufeeque, and Adam Gleave. Planning behavior in a recurrent neural network that plays Sokoban. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024. URL https://openreview.net/forum?id=T9sB3S2hok.

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

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=j1tSLYKwg8.

Alex Graves, Rupesh Kumar Srivastava, Timothy Atkinson, and Faustino Gomez. Bayesian flow networks, 2023. URL https://arxiv.org/abs/2308.07037.

Arthur Guez, Mehdi Mirza, Karol Gregor, Rishabh Kabra, Sebastien Racaniere, Theophane Weber, David Raposo, Adam Santoro, Laurent Orseau, Tom Eccles, Greg Wayne, David Silver, Timothy Lillicrap, and Victor Valdes. An investigation of model-free planning: Boxoban levels. https://github.com/google-deepmind/boxoban-levels/, 2018.

Alec Helbling, Andrey Bryutkin, Mauro Martino, Duen Horng Chau, Nima Dehmamy, and Hendrik Strobelt. Flow reasoning models: Turning discrete flows into efficient recurrent reasoners, 2026. URL https://arxiv.org/abs/2606.29150.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12454–12465, 2021. URL https://openreview.net/forum?id=6nbpPqUCIi7.

Benhao Huang, Zhengyang Geng, and J. Zico Kolter. Equilibrium reasoners: Learning attractors enables scalable reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lh95PnOlpM.

Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks, 2025. URL https://arxiv.org/abs/2510.04871.

Pascal Jutras-Dubé, Patrick Pynadath, Jeremy Lu, Yuan Gao, and Ruqi Zhang. Sticky jump diffusions: A unifying view of masked, continuous, and hybrid diffusion, 2026. URL https://arxiv.org/abs/2607.10951.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30749–30768. PMLR, 2025. URL https://openreview.net/forum?id=DjJmre5IkP. Oral presentation.

Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z. Pan, Hyeji Kim, Sham Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction. In *Forty-third International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2510.01384.

Sanghyun Lee, Chunsan Hong, Seungryong Kim, Jonghyun Lee, Jongho Park, and Dongmin Park. Looped diffusion language models, 2026. URL https://arxiv.org/abs/2605.26106.

Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gómez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/cbf883a744952d4a40591271a58ab9d0-Abstract-Conference.html.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848. PMLR, 2024. URL https://openreview.net/forum?id=CNicRIVIPA.

Matt Mahoney. Large text compression benchmark. http://mattmahoney.net/dc/textdata, 2011.

Rodrigo Casado Noguerales, Bernhard Schölkopf, Thomas Hofmann, and Aran Raoufi. What does a discrete diffusion model learn?, 2026. URL https://arxiv.org/abs/2607.05381.

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 (ICLR)*, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Avishek Joey Bose, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling, 2025. URL https://arxiv.org/abs/2502.03540.

Le-Tuyet-Nhi Pham, Dario Shariatian, Antonio Ocello, Giovanni Conforti, and Alain Oliviero Durmus. Discrete Markov probabilistic models: An improved discrete score-based framework with sharp convergence bounds under minimal assumptions. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 49195–49258. PMLR, 2025. URL https://proceedings.mlr.press/v267/pham25a.html.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 4816–4828, 2021. URL https://openreview.net/forum?id=Tqx7nJp7PR.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Pierre H. Richemond, Sander Dieleman, and Arnaud Doucet. Categorical SDEs with simplex diffusion, 2022. URL https://arxiv.org/abs/2210.14784.

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://openreview.net/forum?id=L4uaAR4ArM.

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 (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 52584–52619. PMLR, 2025. URL https://openreview.net/forum?id=9P9Y8FOSOk.

Subham Sekhar Sahoo, Jean-Marie Lemercier, Zhihan Yang, Justin Deschenaux, Jingyu Liu, John Thickstun, and Ante Jukic. Scaling beyond masked diffusion language models. In *International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=xKJG1CNtQM.

Jinya Sakurai, Patrick Pynadath, Satoshi Hayakawa, Jaehong Yoon, Xulei Yang, Nancy F. Chen, and Xun Xu. Simplex relaxation for discrete diffusion, 2026. URL https://arxiv.org/abs/2608.10615.

Javier E. Santos, Zachary R. Fox, Nicholas Lubbers, and Yen Ting Lin. Blackout diffusion: Generative diffusion models in discrete-state spaces. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9034–9059. PMLR, 2023. URL https://openreview.net/forum?id=lpc5vlfxp8.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dallatorre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=i5MrJ6g5G1.

Amin Sghaier, Ali Parviz, and Alexia Jolicoeur-Martineau. Probabilistic tiny recursive model, 2026. URL https://arxiv.org/abs/2605.19943.

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky T. Q. Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=tcvMzR2NrP. Oral presentation.

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*, volume 37, pp. 103131–103167, 2024. URL https://openreview.net/forum?id=xcqSOfHt4g.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265. PMLR, 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yuxuan Song, Zhe Zhang, Yu Pei, Jingjing Gong, Qiying Yu, Zheng Zhang, Mingxuan Wang, Hao Zhou, Jingjing Liu, and Wei-Ying Ma. ShortListing model: A streamlined simplex diffusion for discrete variable generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. doi: 10.52202/085713-1558. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/42c4ca570eec68981c20094de1cc760e-Abstract-Conference.html.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of

Proceedings of Machine Learning Research, pp. 46495–46513. PMLR, 2024. URL https://proceedings.mlr.press/v235/stark24b.html.

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

Anej Svete and Ashish Sabharwal. On the reasoning abilities of masked diffusion language models. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/e20a65c7308b7b94ed1178eebc45bf76-Abstract-Conference.html.

Huu Binh Ta, Michael Cardei, Alvaro Velasquez, and Ferdinando Fioretto. Search-augmented masked diffusion models for constrained generation, 2026. URL https://arxiv.org/abs/2602.02727.

Alex M. Tseng, Nathaniel Diamant, Tommaso Biancalani, and Gabriele Scalia. GraphGUIDE: Interpretable and controllable conditional graph generation with discrete bernoulli diffusion, 2023. URL https://arxiv.org/abs/2302.03790.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 61810–61843. PMLR, 2025. URL https://openreview.net/forum?id=rvZv7sDPV9.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model, 2025a. URL https://arxiv.org/abs/2506.21734.

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 (NeurIPS)*, 2025b. URL https://openreview.net/forum?id=IJryQAOy0p.

Yuxiang Wang, Yu Xiang, Baojian Zhou, Qifang Zhao, Keyue Jiang, Yanghua Xiao, and Xiaoxiao Xu. On the trainability of masked diffusion language models via blockwise locality, 2026. URL https://arxiv.org/abs/2604.24832.

Christopher Wewer, Bartlomiej Pogodzinski, Bernt Schiele, and Jan Eric Lenssen. Spatial reasoning with denoising models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66706–66725. PMLR, 2025. URL https://proceedings.mlr.press/v267/wewer25a.html.

Jiawei Xu, Minghui Liu, Aakriti Agrawal, Yifan Chen, and Furong Huang. Scheduling thoughts: Learning the order of thought in diffusion language models, 2026. URL https://arxiv.org/abs/2606.23567.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=NRYgUzSPZz.

Tao Zhang, Jia-Shu Pan, Ruiqi Feng, and Tailin Wu. VFScale: Intrinsic reasoning through verifier-free test-time scalable diffusion model. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/6188c02ccc16a7587716de2efd754033-Abstract-Conference.html.

Huangjie Zheng, Shansan Gong, Ruixiang Zhang, Tianrong Chen, Jiatao Gu, Mingyuan Zhou, Navdeep Jaitly, and Yizhe Zhang. Continuously augmented discrete diffusion model for categorical generative modeling, 2025a. URL https://arxiv.org/abs/2510.01329.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. In *International Conference on Learning Representations (ICLR)*, 2025b. URL https://openreview.net/forum?id=CTC7CmirNr.

Cai Zhou, Chenyu Wang, Dinghuai Zhang, Shangyuan Tong, Yifei Wang, Stephen Bates, and Tommi Jaakkola. Next semantic scale prediction via hierarchical diffusion language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1385. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3b3d7aafe5f36995a8320059a858e4b9-Abstract-Conference.html.

APPENDIX CONTENTS

A Masked and uniform diffusion **17**
A.1 Masked diffusion . . . 17
A.2 Uniform diffusion . . . 17

B Mathematical details of binary diffusion **18**
B.1 Binary symmetric diffusion . . . 18
B.2 Binary absorbing diffusion . . . 19
B.3 Binary ancestral bridges . . . 20

C Extended related work **21**
C.1 Discrete diffusion and corruption-state design . . . 21
C.2 Binary diffusion and candidate-set representations . . . 21
C.3 Diffusion for structured reasoning . . . 22
C.4 Recurrent reasoning . . . 22

D Training objectives and time sampling **22**
D.1 Finite-window negative ELBO and boundary terms . . . 22
D.2 Training objectives . . . 23
D.3 Time schedules . . . 23
D.4 Adaptive gradient-RMS importance sampling . . . 24

E Experimental details **24**
E.1 Optimization details . . . 24
E.2 Learning-rate and checkpoint selection . . . 24
E.3 Evaluation . . . 27
E.4 Sampler tuning procedure . . . 27
E.5 Logic tasks . . . 29
E.6 Language modeling . . . 31

F Sudoku solve trajectories **32**

G Generated samples **35**
G.1 Text8 . . . 35
G.2 TinyStories . . . 35
G.3 OpenWebText . . . 37

A MASKED AND UNIFORM DIFFUSION

We specialize the results of Campbell et al. (2022); Noguerales et al. (2026) using the notation of Sections 2–4. Local noisy states are written z, y, and the priors are the limiting distributions from which generation starts.

A.1 MASKED DIFFUSION

For the masked kernel in (1), the off-diagonal forward rates are (Noguerales et al., 2026)

$$Q_t(z, y) = \frac{\mathbf{1}\{z \in [V],\ y = \mathtt{m}\}}{1 - t}, \qquad y \neq z. \tag{19}$$

For $0 \leq s < t < 1$, the transition kernel is

$$q_{t|s}(y \mid z) = \frac{1 - t}{1 - s}\,\mathbf{1}\{y = z\} + \frac{t - s}{1 - s}\,\mathbf{1}\{y = \mathtt{m}\}. \tag{20}$$

The local model masses and denoiser are

$$\begin{aligned} m_i^\theta(y, Z_t; t) &= \begin{cases} (1 - t)\,\mu_i^\theta(y, Z_t; t), & y \in [V], \\ t, & y = \mathtt{m}, \end{cases} \\ \pi_i^\theta(c, Z_t; t) &= \begin{cases} \mu_i^\theta(c, Z_t; t), & Z_t^i = \mathtt{m}, \\ \mathbf{1}\{Z_t^i = c\}, & Z_t^i \in [V]. \end{cases} \end{aligned} \tag{21}$$

For $y \in [V]$ with $y \neq Z_t^i$, the reverse rates are

$$a_i(Z_0, Z_t, y; t) = \frac{\mathbf{1}\{Z_t^i = \mathtt{m},\ y = Z_0^i\}}{t}, \qquad a_i^\theta(Z_t, y; t) = \frac{\mathbf{1}\{Z_t^i = \mathtt{m}\}\,\mu_i^\theta(y, Z_t; t)}{t}. \tag{22}$$

All other off-diagonal reverse rates vanish. The path loss reduces to masked cross-entropy:

$$\mathcal{J}_t(\theta; Z_0, Z_t) = -\frac{1}{t}\sum_i \mathbf{1}\{Z_t^i = \mathtt{m}\}\log \mu_i^\theta(Z_0^i, Z_t; t). \tag{23}$$

Plugging μ_i^θ into the bridge (43) gives the ancestral update of Sahoo et al. (2024): visible tokens are copied, and a masked position unmasks to c with probability $(1 - s/t)\,\mu_i^\theta(c, Z_t; t)$.

A.2 UNIFORM DIFFUSION

For the uniform kernel in (1), the off-diagonal forward rates are (Noguerales et al., 2026)

$$Q_t(z, y) = \frac{1}{V(1 - t)}, \qquad y \neq z. \tag{24}$$

For $0 \leq s < t < 1$, the transition kernel is

$$q_{t|s}(y \mid z) = \frac{1 - t}{1 - s}\,\mathbf{1}\{y = z\} + \frac{t - s}{V(1 - s)}. \tag{25}$$

The local model mass and induced denoiser are

$$\begin{aligned} m_i^\theta(y, Z_t; t) &= (1 - t)\,\mu_i^\theta(y, Z_t; t) + t/V, \\ \pi_i^\theta(c, Z_t; t) &= \frac{\mu_i^\theta(c, Z_t; t)\,[(1 - t)\,\mathbf{1}\{Z_t^i = c\} + t/V]}{(1 - t)\,\mu_i^\theta(Z_t^i, Z_t; t) + t/V}. \end{aligned} \tag{26}$$

For $y \neq Z_t^i$, the clean-conditioned and cavity reverse rates are

$$\begin{aligned} a_i(Z_0, Z_t, y; t) &= \frac{1}{V(1 - t)}\,\frac{(1 - t)\,\mathbf{1}\{y = Z_0^i\} + t/V}{(1 - t)\,\mathbf{1}\{Z_t^i = Z_0^i\} + t/V}, \\ a_i^\theta(Z_t, y; t) &= \frac{1}{V(1 - t)}\,\frac{(1 - t)\,\mu_i^\theta(y, Z_t; t) + t/V}{(1 - t)\,\mu_i^\theta(Z_t^i, Z_t; t) + t/V}. \end{aligned} \tag{27}$$

The path loss is

$$\mathcal{J}_t(\theta; Z_0, Z_t) = \frac{1}{V(1-t)} \sum_i \sum_{y \neq Z_t^i} \Phi\left(\frac{q_{t|0}(y \mid Z_0^i)}{q_{t|0}(Z_t^i \mid Z_0^i)}, \frac{m_i^\theta(y, Z_t; t)}{m_i^\theta(Z_t^i, Z_t; t)} \right). \tag{28}$$

All masses in these quotients are positive on an interior time interval.

Plugging μ_i^θ into the bridge (43) gives the ancestral update of Schiff et al. (2025), which, unlike the masked update, can change a visible token.

Prior term. Corrupting a uniform clean token leaves its distribution uniform. The finite-time and limiting priors thus coincide:

$$p_{t_2}(y) = 1/V, \qquad y \in [V]. \tag{29}$$

For the product sequence prior,

$$\mathbb{E}_{Z_0} \operatorname{KL}\left(q_{t_2|0}(\cdot \mid Z_0) \,\|\, p_{t_2}\right) = L\left[(1 - t_2 + t_2/V)\log\left(V(1-t_2)+t_2\right) + \frac{(V-1)t_2}{V}\log t_2\right]. \tag{30}$$

The uniform prior has full support at every finite cutoff.

B Mathematical details of binary diffusion

We use the candidate-set notation, the candidate-set mass ν_i^θ of (10), and the bitwise loss from Section 3. Superscripts sym and abs specify the kernel and are omitted in derivations where it is fixed. The cavity prediction μ_i^θ, the induced masses m_i^θ, and the denoiser π_i^θ always refer to that kernel.

B.1 Binary symmetric diffusion

Proposition B.1 (Symmetric kernel and reverse rates)**.** *Let each bit flip at rate $1/[2(1-t)]$. For $0 \le s < t < 1$, the probability of a bit differing from its value at s is*

$$\varepsilon_{s,t} = \frac{t-s}{2(1-s)}. \tag{31}$$

The local kernel is

$$q_{t|s}^{\mathrm{sym}}(y \mid z) = \left(1-\varepsilon_{s,t}\right)^{V-d_H(y,z)} \left(\varepsilon_{s,t}\right)^{d_H(y,z)}. \tag{32}$$

Using the odds ρ_t from (8) *and writing $z = Z_t^i$, the reverse rates are*

$$\begin{aligned} a_{i,u}^{\mathrm{sym}}(Z_0, Z_t; t) &= \frac{1}{2(1-t)} \begin{cases} \rho_t, & z_u \neq (Z_0^i)_u, \\ \rho_t^{-1}, & z_u = (Z_0^i)_u, \end{cases} \\ a_{i,u}^{\theta,\mathrm{sym}}(Z_t; t) &= \frac{\rho_t^{2z_u - 1}}{2(1-t)} \frac{1 + (\rho_t^2 - 1)\,\nu_i^\theta(z^{(u)}, Z_t; t)}{1 + (\rho_t^2 - 1)\,\nu_i^\theta(z, Z_t; t)}. \end{aligned} \tag{33}$$

Proof. The bit-flip probability solves

$$\partial_t \varepsilon_{s,t} = \frac{1 - 2\varepsilon_{s,t}}{2(1-t)}, \qquad \varepsilon_{s,s} = 0.$$

The stated flip probability satisfies this equation and initial condition, and multiplying the independent bit kernels proves (32). The clean-conditioned reverse rates follow from the change in Hamming distance: a flip toward the clean bit multiplies the clean-conditioned mass by ρ_t, while a flip away multiplies it by ρ_t^{-1}. Multiplying these ratios by the forward flip rate gives the formula for $a_{i,u}$.

To compute the model mass, put

$$C_t(z) = (1 - t/2)^{V-|z|} (t/2)^{|z|}.$$

The clean-token likelihood is

$$q_{t|0}(z \mid c) = C_t(z)\,\rho_t^{2z_c-1}, \qquad m_i^\theta(z, Z_t; t) = \frac{C_t(z)}{\rho_t}[1 + (\rho_t^2 - 1)\,\nu_i^\theta(z, Z_t; t)].$$

Toggling bit u gives

$$\frac{C_t(z^{(u)})}{C_t(z)} = \rho_t^{2z_u-1}, \qquad \nu_i^\theta(z^{(u)}, Z_t; t) = \nu_i^\theta(z, Z_t; t) + (1 - 2z_u)\,\mu_i^\theta(u, Z_t; t).$$

The quotient of the two local masses, multiplied by $1/[2(1-t)]$, gives the stated model rate. All local masses are positive for $0 < t < 1$, so every bit flip has positive model rate. □

The denoiser used for decoding and sampling is

$$\pi_i^\theta(c, Z_t; t) = \frac{\mu_i^\theta(c, Z_t; t)\,\rho_t^{2(Z_t^i)_c}}{1 + (\rho_t^2 - 1)\,\nu_i^\theta(Z_t^i, Z_t; t)}. \tag{34}$$

Prior. The prior draws independent fair bits, $p_{t_2}^{\mathrm{sym}}(z) = 2^{-V}$. For Bernoulli distributions with success probabilities $a, b \in (0, 1)$, the KL divergence is

$$D_{\mathrm{Ber}}(a\|b) = a \log \frac{a}{b} + (1 - a) \log \frac{1 - a}{1 - b}. \tag{35}$$

Conditional on the clean token c, the bits of $Z_{t_2}^i$ are independent, with bit c active with probability $1 - t_2/2$ and every other bit with probability $t_2/2$. The prior also factors over bits, so the KL is a sum of Bernoulli terms, which coincide because $D_{\mathrm{Ber}}(1 - a\|1/2) = D_{\mathrm{Ber}}(a\|1/2)$:

$$\mathrm{KL}\left(q_{t_2|0}^{\mathrm{sym}}(\cdot \mid c)\|p_{t_2}^{\mathrm{sym}}\right) = V D_{\mathrm{Ber}}(t_2/2\|1/2). \tag{36}$$

This does not depend on c and vanishes as $t_2 \uparrow 1$.

B.2 Binary absorbing diffusion

Proposition B.2 (Absorbing kernel and rates)**.** *Let each zero bit activate at rate $1/(1-t)$ and each active bit remain active. For $0 \le s < t < 1$ and nonempty binary vectors z, y,*

$$q_{t|s}^{\mathrm{abs}}(y \mid z) = \mathbf{1}\{z \subseteq y\} \left(\frac{1-t}{1-s}\right)^{V-|y|} \left(\frac{t-s}{1-s}\right)^{|y|-|z|}. \tag{37}$$

The expression is zero when the subset condition fails. For the clean singleton $\{c\}$, it reduces to

$$q_{t|0}^{\mathrm{abs}}(z \mid c) = \mathbf{1}\{c \in z\}(1-t)^{V-|z|}t^{|z|-1}. \tag{38}$$

Conditioned on Z_0,

$$a_{i,u}^{\mathrm{abs}}(Z_0, Z_t; t) = \frac{\mathbf{1}\{u \in Z_t^i \setminus Z_0^i\}}{t}, \qquad a_{i,u}^{\theta,\mathrm{abs}}(Z_t; t) = \frac{\mathbf{1}\{u \in Z_t^i\}}{t} \frac{\nu_i^\theta(Z_t^i \setminus \{u\}, Z_t; t)}{\nu_i^\theta(Z_t^i, Z_t; t)}. \tag{39}$$

Proof. A bit that is zero at s remains zero at t with probability

$$\exp\left(-\int_s^t \frac{dv}{1-v}\right) = \frac{1-t}{1-s}.$$

Once activated, it remains one. Multiplication over bits proves (37), and $z = \{c\}$, $s = 0$ gives (38). For a nonempty state z,

$$m_i^\theta(z, Z_t; t) = (1-t)^{V-|z|}t^{|z|-1}\,\nu_i^\theta(z, Z_t; t).$$

Deleting an active bit u changes this mass by the factor

$$\frac{(1-t)\,\nu_i^\theta(z \setminus \{u\}, Z_t; t)}{t\,\nu_i^\theta(z, Z_t; t)}.$$

Multiplying by the reversed forward activation rate $1/(1-t)$ gives the model rate. For $Z_0^i = c$, the substitution

$$\mu_i^\theta(u, Z_t; t) = \mathbf{1}\{u = c\}$$

gives the clean-conditioned rate. Deleting the last bit has zero rate because the empty vector has zero mass. □

Proposition B.3 (Absorbing path loss)**.** *Under the assumptions of Proposition B.2,*

$$\mathcal{J}_t^{\mathrm{abs}}(\theta; Z_0, Z_t) = -\frac{1}{t} \sum_i \sum_{u \in Z_t^i \setminus Z_0^i} \log\left(1 - \frac{\mu_i^\theta(u, Z_t; t)}{\nu_i^\theta(Z_t^i, Z_t; t)}\right). \tag{40}$$

Proof. Fix i, write $Z_0^i = c$, and put

$$h_u = \frac{\mu_i^\theta(u, Z_t; t)}{\nu_i^\theta(Z_t^i, Z_t; t)}, \quad u \in Z_t^i, \qquad \sum_{u \in Z_t^i} h_u = 1.$$

The homogeneity identity $\Phi(a/t, b/t) = \Phi(a, b)/t$ gives the local loss times t as

$$\Phi(0, 1 - h_c) + \sum_{u \in Z_t^i \setminus \{c\}} \Phi(1, 1 - h_u) = 1 - h_c + \sum_{u \neq c,\ u \in Z_t^i} [-h_u - \log(1 - h_u)].$$

Because the h_u sum to one, the linear terms cancel, leaving (40). For a singleton, the remaining sum is empty and the loss is zero. □

For absorbing diffusion, the local kernel is constant over clean tokens in Z_t^i and zero outside it. Bayes reweighting gives the denoiser

$$\pi_i^\theta(c, Z_t; t) = \frac{\mu_i^\theta(c, Z_t; t)\, \mathbf{1}\{c \in Z_t^i\}}{\nu_i^\theta(Z_t^i, Z_t; t)}. \tag{41}$$

The cavity parameterization applies this support restriction analytically, while a separately predicted denoiser must respect it through explicit masking or learning.

B.3 Binary ancestral bridges

By Bayes' rule and the Markov property, the forward bridge of Section 4 is

$$q_{s|t,0}(y \mid z, c) = \frac{q_{t|s}(z \mid y)\, q_{s|0}(y \mid c)}{q_{t|0}(z \mid c)}. \tag{42}$$

Averaging it over the denoiser (15) cancels $q_{t|0}(z \mid c)$, which gives the plug-in form of the ancestral update (16):

$$p_{s|t,i}^\theta(y \mid Z_t) = \frac{q_{t|s}(Z_t^i \mid y) \sum_c \mu_i^\theta(c, Z_t; t)\, q_{s|0}(y \mid c)}{\sum_c \mu_i^\theta(c, Z_t; t)\, q_{t|0}(Z_t^i \mid c)}. \tag{43}$$

This is the bridge with the cavity in place of the clean token. We now derive the conditional bit bridges from (42).

Symmetric bridge. Conditional on the clean proposal $\widehat{Z}_0^i = c$, the bits are independent. With $\varepsilon_{s,t}$ from (6), bit u with clean value $b_u = \mathbf{1}\{u = c\}$ and current value $z_u = (Z_t^i)_u$ differs from b_u at s with probability

$$q^{\mathrm{sym}}\big((Z_s^i)_u \neq b_u \mid z_u, b_u\big) = \begin{cases} s\, \varepsilon_{s,t}/(2 - t), & z_u = b_u, \\ s\,(1 - \varepsilon_{s,t})/t, & z_u \neq b_u, \end{cases} \tag{44}$$

the chance of differing from the clean bit at s and then flipping, or not, to the current value at t. It favors the current value for s close to t and the clean bit as $s \downarrow 0$.

A simpler symmetric sampler applies a one-bit Bayes update to each cavity bit marginal and draws the bits independently. This ignores the dependence between the bits through the shared clean token: at $s = 0$, the exact bridge draws a single one-hot state from π_i^θ, whereas independent bit draws can leave zero or several candidates active. It performed worse in early experiments and was dropped.

Absorbing bridge. Conditional on $\widehat{Z}_0^i = c$, the bit c stays active, and every bit inactive in Z_t^i remains inactive at s. For $u \in Z_t^i \setminus \{c\}$,

$$q^{\mathrm{abs}}((Z_s^i)_u = 1 \mid (Z_t^i)_u = 1, Z_0^i = c) = \frac{s}{t}. \tag{45}$$

Indeed, a false candidate is active at s with probability s, active at t with probability t, and activation by s implies activation by t. These bit events are independent conditional on c and the current state. The update therefore preserves nonemptiness and produces a subset of the current active set.

C Extended related work

C.1 Discrete diffusion and corruption-state design

Several lines of work broaden the forward path without introducing candidate sets. Generalized interpolating diffusion and kinetic path design admit richer categorical couplings (von Rütte et al., 2025; Shaul et al., 2025), partial masking inserts intermediate corruption states between a token and a full mask (Chao et al., 2025), and sticky jump processes combine discrete jumps with continuous coordinates (Jutras-Dubé et al., 2026). CADD and CANDI go further and augment or hybridize the categorical state with a continuous one (Zheng et al., 2025a; Pynadath et al., 2025). Even at scale, and with architectures and training budgets matched, the choice between masking, uniform corruption, and interpolating paths remains consequential (Sahoo et al., 2026).

There is also a substantial simplex-valued lineage, which lifts categories to continuous points on the simplex or updates distribution parameters directly (Richemond et al., 2022; Floto et al., 2023; Avdeyev et al., 2023; Stark et al., 2024; Graves et al., 2023; Chandra et al., 2026). Simplax is the closest of these to our setting: it keeps uniform categorical diffusion as its marginal and adds an exact Dirichlet auxiliary variable to the reverse bridge (Sakurai et al., 2026). Its denoiser, however, still observes a corrupted category, whereas binary diffusion exposes the candidate set to the denoiser as the noisy state.

C.2 Binary diffusion and candidate-set representations

Prior binary diffusion methods treat each bit as part of the object being generated. Bit-flip diffusion of binary data goes back to binomial diffusion (Sohl-Dickstein et al., 2015), and DMPM studies its continuous-time form (Pham et al., 2025). BerDiff applies a Bernoulli kernel to binary segmentation masks (Chen et al., 2023a), GraphGUIDE adds, removes, or flips graph edges through Bernoulli transitions (Tseng et al., 2023), and Blackout diffusion models discrete counts through birth–death processes (Santos et al., 2023). Analog Bits instead encodes one category in $\lceil \log_2 V \rceil$ real-valued bit coordinates and applies continuous diffusion (Chen et al., 2023b). In binary diffusion, by contrast, there are V binary coordinates per position and an active bit denotes membership in the noisy subset.

The ShortListing Model (SLM) is the closest predecessor (Song et al., 2025). It represents a categorical variable by a nonempty candidate set, starts the forward process from the clean singleton, activates false candidates until reaching the full vocabulary, and samples in reverse by progressive candidate pruning. However, the model is formulated in discrete time and identifies sets with simplex-face centroids. To see the correspondence with binary absorbing diffusion, let n_t be SLM's expected candidate-set size at time t and take the linear noise schedule

$$n_t = 1 + (V - 1)t. \tag{46}$$

The probability of adding a candidate between times $s < t$ is then

$$\frac{n_t - n_s}{V - n_s} = \frac{t - s}{1 - s}, \tag{47}$$

and the clean-conditioned probability of retaining a false candidate is

$$\frac{n_s - 1}{n_t - 1} = \frac{s}{t}, \tag{48}$$

which are the absorbing kernel (12) and bridge (45).

C.3 Diffusion for structured reasoning

Beyond Autoregression established masked diffusion baselines on Sudoku and other reasoning and planning tasks (Ye et al., 2025). Dirichlet diffusion solves Sudoku in a continuous simplex state (Avdeyev et al., 2023), and Simplax reports Sudoku generalization across clue densities (Sakurai et al., 2026). Several analyses show that the corruption and the order of updates matter as much as the model: Spatial Reasoning Models find that the order of variable updates strongly affects accuracy (Wewer et al., 2025), and the locality induced by the corruption and factorization changes trainability on Sudoku and graph path-finding (Wang et al., 2026). On the theoretical side, masked diffusion models can simulate chain-of-thought under finite precision (Svete & Sabharwal, 2026).

A separate line of work changes the reverse-time policy rather than the forward state, through token ordering, planned denoising and path planning, remasking, learned reveal schedules, or search during and around sampling (Kim et al., 2025; Liu et al., 2025; Peng et al., 2025; Wang et al., 2025b; Kim et al., 2026; Xu et al., 2026; Ta et al., 2026; Bai et al., 2026; Zhang et al., 2026). These methods compose with any corruption kernel, and beyond the confidence-guided re-corruption of Section 4 we use none of them: our verifier-free solve rate refers to direct samples from each kernel's tuned sampler, without search, reranking, or a symbolic solver.

C.4 Recurrent reasoning

Sudoku and related benchmarks have also driven a rapidly growing literature on small recurrent reasoners, which allocate computation through repeated latent updates, fixed-point dynamics, or looped layers (Wang et al., 2025a; Jolicoeur-Martineau, 2025; Sghaier et al., 2026; Huang et al., 2026; Fein-Ashley & Rashidinejad, 2026; Helbling et al., 2026; Cameron et al., 2026; Lee et al., 2026). These approaches change the architecture, the training dynamics, or the recurrent depth, which is a separate axis from the corruption state that binary diffusion changes, so we do not compare against them directly.

D Training objectives and time sampling

D.1 Finite-window negative ELBO and boundary terms

The idealized objective (3) integrates the reverse process from clean data at $t = 0$ to the terminal noise distribution at $t = 1$. In practice we use an interior window $0 < t_1 < t_2 < 1$ to avoid singular jump rates near the endpoints. The corresponding finite-window negative ELBO, abbreviated NELBO in the appendices, is

$$\mathrm{NELBO}(\theta; t_1, t_2) = \mathcal{R}_{t_1}(\theta) + \mathcal{K}_{t_2} + \int_{t_1}^{t_2} \mathbb{E}_{Z_0, Z_t}[\mathcal{J}_t(\theta; Z_0, Z_t)]\, dt, \tag{49}$$

where

$$\mathcal{R}_{t_1}(\theta) = \mathbb{E}_{Z_0, Z_{t_1}}[-\log p^{\theta}_{0|t_1}(Z_0 \mid Z_{t_1})], \qquad \mathcal{K}_{t_2} = \mathbb{E}_{Z_0}\,\mathrm{KL}\left(q_{t_2|0}(\cdot \mid Z_0) \,\|\, p_{t_2}\right). \tag{50}$$

The first term accounts for the residual corruption at the clean end of the truncated path, while the second measures the mismatch between the forward distribution at t_2 and the prior from which generation starts. Consequently, the path integral alone is not generally comparable across different kernels or time windows: changing t_1 or t_2 can move substantial information into the boundary terms.

For uniform and binary symmetric diffusion the terminal-prior term is finite and available analytically, with the expressions given in Apps. A.2 and B.1. For masked and binary absorbing diffusion, the limiting prior is a point mass and its KL from the finite-t_2 forward distribution is infinite, and our likelihood-evaluation convention for these kernels is described in App. E.3. As $t_1 \downarrow 0$ and $t_2 \uparrow 1$, with the terminal prior matched to the limiting forward distribution, the boundary contributions vanish and (49) reduces to (3).

D.2 TRAINING OBJECTIVES

Alongside the path ELBO, we compared a flat ELBO, cross-entropy (xent), and weighted cross-entropy (wxent). For the kernels considered here, the reverse rates share a scalar prefactor,

$$r_t = \begin{cases} 1/[2(1-t)], & \text{binary symmetric,} \\ 1/[V(1-t)], & \text{uniform,} \\ 1/t, & \text{binary absorbing,} \\ 1/t, & \text{masked,} \end{cases} \tag{51}$$

and the remaining factors are likelihood ratios at the current noisy token. Since $\Phi(ra, rb) = r\Phi(a, b)$, the prefactor is also the weight of the jump penalties in the path integrand $\mathcal{J}_t$ of (3). The flat ELBO drops it, $\mathcal{J}_t^{\text{flat}} = \mathcal{J}_t / r_t$, which changes the relative weight of the noise levels. Cross-entropy trains the network output directly as a denoiser,

$$\mathcal{J}_t^{\text{xent}}(\theta; Z_0, Z_t) = -\sum_{i=1}^{L} \log \pi_i^\theta(Z_0^i, Z_t; t), \tag{52}$$

and weighted cross-entropy multiplies it by the prefactor, $\mathcal{J}_t^{\text{wxent}} = r_t \, \mathcal{J}_t^{\text{xent}}$. For masked diffusion the sums run over masked positions. The ELBO and wxent are integrated over time as in (3), while the flat ELBO and xent are averaged over the base schedule of App. D.3. All four are estimated with the inverse-density weight defined there, which the adaptive proposal of App. D.4 preserves.

Logic models train on the path ELBO. For text, we selected the objective of each kernel and corpus on the validation NELBO after 10% of the training budget. The path ELBO was selected everywhere except on Text8, where masked diffusion trains on xent, which coincides with its flat ELBO, and binary absorbing diffusion on wxent. The latter still bounds the likelihood, as we show next.

Absorbing cross-entropy bound. For binary absorbing diffusion, wxent upper-bounds the path integrand. Fix a position with clean token $Z_0^i = c$. The denoiser is supported on the active set Z_t^i, which contains c, so

$$\prod_{u \in Z_t^i \setminus \{c\}} \big(1 - \pi_i^\theta(u, Z_t; t)\big) \geq 1 - \sum_{u \in Z_t^i \setminus \{c\}} \pi_i^\theta(u, Z_t; t) = \pi_i^\theta(c, Z_t; t). \tag{53}$$

Taking negative logarithms, multiplying by $1/t$, and summing over positions in (40) gives $\mathcal{J}_t^{\text{abs}} \leq \mathcal{J}_t^{\text{wxent}}$.

D.3 TIME SCHEDULES

The base schedule maps a uniform coordinate $u \in [0, 1]$ to corruption time on the clipped window $[t_1, t_2]$. The linear, logarithmic, and cosine maps are

$$t_{\text{lin}}(u) = t_1 + (t_2 - t_1)u, \qquad t_{\log}(u) = t_1 \left(\frac{t_2}{t_1}\right)^u, \qquad t_{\cos}(u) = t_1 + (t_2 - t_1)\sin^2\Big(\frac{\pi u}{2}\Big), \tag{54}$$

which give the base sampling densities on $[t_1, t_2]$

$$p_{\text{lin}}(t) = \frac{1}{t_2 - t_1}, \qquad p_{\log}(t) = \frac{1}{t \log(t_2/t_1)}, \qquad p_{\cos}(t) = \frac{1}{\pi\sqrt{(t - t_1)(t_2 - t)}}. \tag{55}$$

Logarithmic sampling assigns equal probability to intervals with the same ratio of endpoints, giving more resolution near clean data. This is useful for binary diffusion because a small change in noise level can substantially change a small candidate set, particularly with a large vocabulary. Cosine sampling instead gives more resolution near both ends of the window. To estimate the path integral without bias, each sequence's loss is weighted by the inverse density of its sampled time, namely $t_2 - t_1$ for the linear schedule, $t \log(t_2/t_1)$ for the logarithmic schedule, and $\pi\sqrt{(t - t_1)(t_2 - t)}$ for the cosine schedule.

The network receives the normalized schedule coordinate as its time input:

$$u(t) = \begin{cases} \dfrac{t - t_1}{t_2 - t_1}, & \text{linear schedule,} \\ \dfrac{\log(t/t_1)}{\log(t_2/t_1)}, & \text{logarithmic schedule,} \\ \dfrac{2}{\pi} \arcsin \sqrt{\dfrac{t - t_1}{t_2 - t_1}}, & \text{cosine schedule.} \end{cases} \tag{56}$$

The adaptive proposal below uses this same coordinate, while the kernel and rates retain their definitions in corruption time t. Inference grids are chosen separately from training-time sampling. Applying any map in (54) to j/N, for $j = 0, \ldots, N$, gives a linear, logarithmic, or cosine grid for N reverse steps, traversed in decreasing j as in Algorithm 1. The logarithmic grid has smaller absolute steps near t_1, and the cosine grid near both ends of the window. Training uses the linear and logarithmic schedules (Tables 3 and 4), and the sampling grids use all three (Table 7).

D.4 Adaptive gradient-RMS importance sampling

Adaptive importance sampling replaces the base schedule by a proposal that draws more samples where loss gradients are large. It improved over the base schedule alone in all our experiments, so every model, for all four kernels and all tasks, trains with it. The statistic is the root-mean-square norm of the gradient with respect to the network's output logits, used as an inexpensive proxy for parameter-gradient norms.

We divide the base coordinate u into $K = 32$ equal-width bins. Let g_θ denote a sequence's contribution to the loss under the base schedule, including token normalization and the base-schedule weight, and let ℓ denote its output logits. For the path objective with L scored tokens per sequence, this contribution is

$$g_\theta = \frac{\mathcal{J}_t(\theta; Z_0, Z_t)}{L\, p_{\mathrm{base}}(t)}, \qquad p_{\mathrm{base}} \in \{p_{\mathrm{lin}}, p_{\mathrm{log}}, p_{\mathrm{cos}}\}. \tag{57}$$

When scored token counts vary across sequences, we replace L by their minibatch mean. In each bin k, we maintain exponentially decayed sums S_k of $\|\nabla_\ell g_\theta\|_2^2$ and counts C_k, both with decay 0.99. Their ratio gives the mean squared gradient norm, from which we form the RMS estimate and the bin probability:

$$G_k = \sqrt{\frac{S_k}{C_k}}, \qquad P_k = 0.9 \frac{G_k}{\sum_{j=1}^{K} G_j} + \frac{0.1}{K}. \tag{58}$$

Sampling uniformly within each selected bin gives the time density and path-loss weight

$$p(t) = K P_k\, p_{\mathrm{base}}(t), \qquad \frac{1}{p(t)} = \frac{1}{K P_k\, p_{\mathrm{base}}(t)}, \qquad u(t) \text{ in bin } k. \tag{59}$$

The additional factor $1/(KP_k)$ preserves the objective under the base schedule. Mixing in 10% of the base schedule keeps every bin accessible and bounds this factor by 10.

For the first 100 minibatches we sample from the base schedule while collecting statistics. Each minibatch uses the preceding proposal, held fixed during differentiation, and updates the statistics for the next minibatch. Times within a minibatch are stratified in the proposal's cumulative-probability coordinate. Validation draws use the base schedule.

E Experimental details

E.1 Optimization details

Tables 3 and 4 give the architectures and optimization settings of the models.

E.2 Learning-rate and checkpoint selection

We selected the learning rate of each task and kernel from $\{10^{-4}, 3 \cdot 10^{-4}, 10^{-3}\}$ on the validation NELBO of App. E.3, computed on 51,200 validation draws and averaged over two training seeds

Table 3: Training settings of the logic models.

Architecture	
transformer layers	12
hidden width	512
attention heads	8
MLP width	2048
time-conditioning width	128
conditioning	Sudoku: clue embedding; Boxoban: board prefix; graph coloring: adjacency in attention
sequence length	Sudoku 81; Boxoban $100 + 128$ (flattened board + solution); graph coloring 64
precision	mixed: float32 parameters, normalization, logits, and loss; bfloat16 matmuls, attention, and residual stream
dropout	0
Optimization	
optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.999)$
learning rate	10^{-4}; Boxoban $3 \cdot 10^{-4}$ except binary symmetric
schedule	constant after 1,000 warm-up steps
batch size	512
training steps	Sudoku 1M, Boxoban 2.5M, graph coloring 300k
weight decay	0.05, excluding norms and embeddings
gradient clip	1
EMA decay	0.9999
training seeds	5
Diffusion	
objective	path ELBO, cavity parameterization
time window	$[0.01, 0.9999]$
base schedule	linear, low-discrepancy sampling
adaptive sampling	gradient RMS, EMA 0.99

Table 4: Training settings of the text models.

Architecture	
transformer layers	12; OpenWebText 16
hidden width	768; OpenWebText 1024
attention heads	12; OpenWebText 16
MLP width	3072; OpenWebText 4096
time-conditioning width	128
sequence length	Text8 256, TinyStories 512, OpenWebText 1024
vocabulary	Text8 27, TinyStories 1024, OpenWebText 1024
precision	mixed: float32 parameters, normalization, logits, and loss; bfloat16 matmuls, attention, and residual stream
dropout	0
Optimization	
optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.999)$
learning rate	$3 \cdot 10^{-4}$
schedule	constant after 2,500 warm-up steps
batch size	512
training steps	Text8 1M, TinyStories 500k, OpenWebText 250k
weight decay	0.02, excluding norms and embeddings
gradient clip	1
EMA decay	0.9999
training seeds	1
Diffusion	
objective	path ELBO, cavity parameterization; Text8: xent for masked, wxent for binary absorbing
time window	Text8 $[0.001, 0.9999]$; TinyStories and OpenWebText $[0.0005, 0.9999]$
base schedule	linear; binary kernels logarithmic; low-discrepancy sampling
adaptive sampling	gradient RMS, EMA 0.99

(Table 5). Within each task, kernel, and seed, all learning rates share the data, times, and corruptions, so their differences have a Monte Carlo standard error below $3 \cdot 10^{-4}$, and the variation between seeds dominates. Sudoku and graph coloring are tuned at their full training budget, while Boxoban is tuned at 1M steps. On Boxoban uniform and binary absorbing, the two lowest learning rates are within seed noise, and in both cases we keep the middle rate $3 \cdot 10^{-4}$. Since Boxoban validation NELBO was still improving slowly at that point, we trained the selected rates for 2.5M steps and selected a checkpoint for each run on the validation NELBO, evaluated every 100k steps from 1M. We keep the final checkpoint unless an earlier one is lower by more than two standard errors of the paired difference, and only two binary absorbing seeds select an earlier checkpoint, at 2.0M and 1.0M steps.

Table 5: Validation NELBOs for the logic learning-rate sweep in nats per scored token, mean over two training seeds with half their difference in parentheses.

Task	Kernel	10^{-4}	$3{\cdot}10^{-4}$	10^{-3}
Sudoku	Binary symmetric	**0.0163** (0.0005)	0.0175 (0.0002)	0.0230 (0.0007)
	Uniform	**0.0162** (0.0013)	0.0179 (0.0001)	0.0226 (0.0005)
	Binary absorbing	**0.0148** (0.0001)	0.0178 (0.0010)	0.0229 (0.0011)
	Masked	**0.0154** (0.0004)	0.0176 (0.0002)	0.0245 (0.0009)
Boxoban	Binary symmetric	**0.0286** (0.0017)	0.0306 (0.0034)	0.0358 (0.0006)
	Uniform	0.0267 (0.0026)	**0.0256** (0.0009)	0.0332 (0.0007)
	Binary absorbing	**0.0290** (0.0005)	0.0309 (0.0005)	0.0394 (0.0002)
	Masked	0.0294 (0.0001)	**0.0266** (0.0002)	0.0338 (0.0009)
Graph coloring	Binary symmetric	**0.0863** (0.0001)	0.0873 (0.0000)	0.0905 (0.0000)
	Uniform	**0.0857** (0.0000)	0.0865 (0.0002)	0.0896 (0.0002)
	Binary absorbing	**0.0881** (0.0001)	0.0898 (0.0001)	0.0951 (0.0003)
	Masked	**0.0837** (0.0001)	0.0845 (0.0001)	0.0880 (0.0005)

Table 6: Test NELBOs of the final logic models in nats per scored token, mean and standard error over the five training seeds.

Task	Binary symmetric	Uniform	Binary absorbing	Masked
Sudoku	0.0165 (0.0003)	**0.0160** (0.0004)	0.0161 (0.0001)	0.0166 (0.0002)
Boxoban	**0.0224** (0.0002)	0.0235 (0.0005)	0.0278 (0.0004)	0.0236 (0.0002)
Graph coloring	0.0860 (0.0001)	0.0857 (0.0001)	0.0877 (0.0001)	**0.0837** (0.0000)

E.3 Evaluation

NELBO. We evaluate the finite-window NELBO (49) on the training window with the estimators used in training. The path term draws one time per sequence from the training time schedule with inverse-density weighting. Reconstruction is estimated from forward corruptions at t_1 with the denoiser as decoder, and visible tokens in masked diffusion are copied exactly and have zero reconstruction loss. The prior is the limiting distribution from which generation starts. Its KL from the time-t_2 corruption is analytic for uniform and binary symmetric diffusion (Apps. A.2 and B.1). For masked and binary absorbing diffusion the prior is a point mass, so that the KL is infinite for $t_2 < 1$ and zero at $t_2 = 1$, and we omit it. We aggregate the total NELBO over scored tokens, excluding padding and conditioning clues.

Logic solve rate. Logic models are scored on direct samples, with one attempt per instance and no verifier or selection among attempts, so the reported solve rate is single-attempt success. Sudoku success requires the completed reference grid, and Boxoban success requires legal actions that end with every box on a goal, with the action sequence read up to the first `STAY`, the no-operation action that pads solutions. Graph coloring accepts any proper coloring. We evaluate the selected EMA checkpoint of each of the five training seeds (App. E.2) with a linear sampling grid on $[0.01, 0.9999]$ and the sampler selected on validation at the reference NFE, 256 for Sudoku and Boxoban and 64 for graph coloring (App. E.4). The tables use the full test splits, and the NFE and training curves use 8,192 and 4,096 test instances, except in their hard columns, which use every hard test instance. Every kernel is evaluated on the same instances, and the difficulty bins are fixed on validation data. The hard subsets are defined with the data in App. E.5. We report the mean and standard error over training seeds.

Text sample quality. The test split is obtained by halving the validation split, and every sampler setting is tuned on the other half, which we simply call the validation split. After removing special tokens, we retokenize complete decoded strings with GPT-2 Large for GenPPL and MAUVE features. Each sample set holds 2,000 samples from one generation seed, and each reported value averages the two generation seeds of a sampler setting. Entropy is the unigram entropy, in nats, of the GPT-2 tokens pooled over a sample set rather than a within-string average, so it measures the diversity of the set as a whole. GenPPL is the exponential of the total GPT-2 Large negative log-likelihood of the sample set divided by its total token count, so it is pooled over tokens rather than averaged over strings. MAUVE uses 200 buckets and is averaged over four reference draws and five clustering seeds, with reference subsets sampled without replacement and of the size of the generated sample set. Error bars span the two generation seeds and condition on the checkpoint.

E.4 Sampler tuning procedure

Every kernel is sampled with the confidence-guided ψ-sampler of Algorithm 1. Its settings are the proposal temperature T, the re-corrupt fraction η, the position-confidence strength γ, the ordering tilt λ, which moves the noise level a re-corrupted position returns to in proportion to $\exp(-\lambda\, \mu_i^\theta(\widehat{Z}_0^i, Z_t; t))$ while keeping the mean at the grid level, the coordinate in which the proposal's confidence is read, the cavity μ_i^θ or the denoiser π_i^θ, and the candidate-confidence strength β, tuned on the logic tasks for every kernel but masked diffusion. Here $\eta = 0$ is ancestral sampling and $\eta = 1$ without confidence is plain re-corruption. With candidate confidence, binary diffusion re-corrupts the proposal's own bit as under the forward kernel and reopens each alternative $u \neq \widehat{Z}_0^i$ independently with probability $\min\{1, c_i\, p_{i,u}^\beta\}$, where c_i keeps the expected number of reopened alternatives at its forward value, $(V-1)s_i/2$ for binary symmetric and $(V-1)s_i$ for binary absorbing diffusion. The weight $p_{i,u}$ starts at $1/V$ and is taken from the last step at which u was

a candidate, because the binary absorbing denoiser vanishes outside the current set. Without this memory, reopening in proportion to the current cavity or denoiser probability lowers binary symmetric diffusion's best Sudoku validation solve rate over the same grid to 89.6% and 85.2%, below position confidence alone (95.2%). Uniform diffusion re-corrupts a position with its forward probability $(V-1)s_i/V$ of changing token and draws the new token $u \neq \widehat{Z}_0^i$ with probability proportional to $\pi_i^\theta(u, Z_t; t)^\beta$, so that $\beta = 0$ is the forward kernel. Text sampling also tunes the nucleus truncation top-p and the spacing of the sampling grid. Temperature and top-p are applied to the denoiser rather than to the cavity prediction (Noguerales et al., 2026).

For each task and kernel these settings are selected on validation over the grids of Table 7. On the logic tasks the score is the mean solve rate over the training seeds on a fixed validation split of 2,048 instances (1,024 for the large graphs) at the reference NFE. On text it is MAUVE at NFE $= L$, screened with 512 samples and confirmed with 2,000 samples for the three best candidates. The selection is then frozen for every test measurement, budget, and training seed unless otherwise stated. The hard column sampler configuration is selected separately on the hard subsets of the validation splits. The grids are not full products: the ordering and confidence strengths enter through a few sub-grids, and masked diffusion, which has no candidate to reopen, receives a matching number of extra ordering and re-corruption points in place of the candidate-confidence settings (Table 7), so every kernel is tuned with the same budget. Because the cavity and the denoiser coincide for masked diffusion, the confidence coordinate has no effect there. Table 10 selects over nested subsets of the logic grid, ancestral sampling, re-corruption alone, the ψ-sampler without confidence, with position confidence, and with every setting, so each row reports what its settings add. Candidate confidence lifts binary symmetric diffusion on Sudoku and binary absorbing diffusion on the large graphs, whereas it is only selected for uniform diffusion on the Boxoban hard subset. Tables 8 and 9 list the selected settings.

Figure 3 measures each kernel at several entropies: the sampler settings screened on validation are binned by GPT-2 token entropy, the best of each bin is evaluated on the test split with 2,000 samples and two generation seeds, and the figure draws the upper envelope of these points over entropy, on which the validation-selected setting lies for every kernel.

Table 7: Sampler settings tuned on validation.

Setting	Logic tasks	Text
Temperature T	$\{0, 0.5, 1\}$	$\{0.8, 0.9, 1, 1.1\}$
Re-corrupt fraction η	$\{0, 0.02, 0.05, 0.1, 0.2, 0.5, 0.8, 1\}$	$\{0, 0.01, 0.05, 0.1, 0.2, 0.5, 0.8, 1\}$
masked also	$\{0.9\}$	
Position confidence γ	$\{0, 1, 3, 10\}$	$\{0, 1, 3, 5, 10\}$
Ordering tilt λ	$\{0, 1, 3, 10, 30, 100\}$	$\{0, 1, 3, 10\}$
masked also	$\{2, 5, 7, 15, 20, 50, 300, 1000\}$	
Candidate confidence β	$\{0, 0.5, 1, 2, 3\}$, all but masked	$\{0, 0.5, 1, 2\}$, binary
Confidence coordinate	cavity, denoiser	cavity, denoiser
Grid spacing	linear	cosine, logarithmic
Nucleus top-p	1	$\{0.9, 1\}$

Table 8: Selected sampler settings on the logic tasks, on the full validation splits and on the hard subsets. The confidence coordinate (coord.) is the cavity (c) or the denoiser (d).

Kernel	Setting	Sudoku All	Sudoku Hard	Boxoban All	Boxoban Hard	Graph coloring All	Graph coloring Hard
Binary symmetric	T	0	0	0	1	1	0
	η	1	1	1	1	0.8	1
	coord.	c	c	c	c	c	c
	γ	0	0	0	0	0	0
	λ	0	0	3	3	0	3
	β	0.5	0.5	3	0.5	0	0
Uniform	T	1	1	1	1	1	1
	η	0.5	0.5	0.5	1	0.5	1
	coord.	c	c	c	c	d	c
	γ	10	10	3	0	3	0
	λ	0	0	3	3	0	30
	β	0	0	0	2	0	0
Binary absorbing	T	0.5	0	0	0	0.5	0
	η	1	1	1	1	0.5	1
	coord.	d	d	c	c	c	c
	γ	0	0	0	0	1	0
	λ	1	1	1	0	3	0
	β	0	0	0	0.5	0	0.5
Masked	T	0	0.5	0	0	0.5	0.5
	η	0.9	0.9	0.5	0.8	0.8	1
	coord.	c	c	c	c	c	c
	γ	0	0	10	0	3	0
	λ	3	3	0	3	3	2
	β	0	0	0	0	0	0

Table 10: Solve rate (%) at the reference NFE for each stage of the tuning procedure, selected on validation and evaluated on the full test splits and the hard subsets of Table 1. A stage that keeps the sampler of the row above still repeats its test run with new sampling randomness (such repeats agree within the standard errors). For masked diffusion, the last row adds the extra ordering and re-corruption settings that replace candidate confidence.

	Binary symmetric		Uniform		Binary absorbing		Masked	
Sampler	All	Hard	All	Hard	All	Hard	All	Hard
Sudoku								
Ancestral	76.41 (0.33)	59.22 (0.34)	**76.66** (0.49)	**60.37** (0.42)	71.90 (0.24)	52.25 (0.34)	70.31 (0.29)	50.75 (0.48)
Re-corrupt	94.54 (0.18)	87.78 (0.40)	**94.77** (0.31)	**88.27** (0.64)	86.03 (0.24)	71.75 (0.54)	80.84 (0.09)	64.39 (0.21)
ψ-sampler	94.54 (0.18)	87.78 (0.40)	**94.77** (0.31)	**88.40** (0.68)	86.03 (0.24)	71.89 (0.49)	80.49 (0.13)	63.95 (0.24)
+ position conf.	94.37 (0.18)	87.47 (0.21)	**95.48** (0.22)	**89.90** (0.48)	91.23 (0.11)	81.24 (0.21)	88.19 (0.19)	76.56 (0.29)
+ candidate conf.	**96.27** (0.21)	**92.66** (0.39)	95.47 (0.21)	89.81 (0.47)	91.24 (0.11)	81.37 (0.20)	87.96 (0.16)	76.00 (0.10)
Boxoban								
Ancestral	**83.80** (0.37)	**64.17** (0.57)	80.00 (0.81)	55.33 (1.50)	71.75 (0.91)	43.50 (1.59)	76.39 (0.37)	50.71 (0.58)
Re-corrupt	**93.53** (0.39)	**81.89** (0.85)	91.68 (0.63)	77.35 (1.30)	84.69 (0.66)	66.14 (1.45)	90.62 (0.37)	75.87 (0.93)
ψ-sampler	**93.53** (0.39)	**81.89** (0.85)	91.68 (0.63)	77.35 (1.30)	84.66 (0.65)	66.01 (1.38)	90.66 (0.34)	75.94 (0.88)
+ position conf.	**94.38** (0.33)	**83.60** (0.78)	92.34 (0.61)	78.39 (1.36)	87.28 (0.80)	68.25 (1.65)	91.24 (0.31)	76.60 (0.70)
+ candidate conf.	**94.29** (0.30)	**84.14** (0.61)	92.38 (0.64)	78.57 (1.42)	87.30 (0.81)	68.25 (1.65)	91.26 (0.31)	76.60 (0.70)
Graph coloring								
Ancestral	**97.47** (0.05)	4.70 (0.24)	97.33 (0.03)	**4.98** (0.25)	83.17 (0.13)	0.00 (0.00)	63.00 (0.54)	0.00 (0.00)
Re-corrupt	**99.38** (0.01)	**55.00** (1.01)	98.94 (0.01)	48.16 (0.92)	88.54 (0.33)	0.58 (0.09)	86.95 (0.60)	0.05 (0.03)
ψ-sampler	**99.35** (0.00)	**55.10** (1.00)	98.94 (0.01)	48.24 (0.94)	90.64 (0.19)	0.58 (0.09)	86.92 (0.56)	0.05 (0.03)
+ position conf.	**99.35** (0.00)	**64.49** (1.20)	99.00 (0.01)	61.41 (0.34)	98.83 (0.02)	19.96 (0.97)	97.04 (0.03)	4.94 (0.37)
+ candidate conf.	**99.35** (0.00)	**64.13** (1.10)	99.01 (0.02)	61.41 (0.34)	98.83 (0.02)	54.77 (0.39)	97.44 (0.04)	4.85 (0.33)

E.5 LOGIC TASKS

In every task the given part of the instance conditions the model: it is shown clean, never corrupted, and excluded from the loss, so all randomness and revision act on the generated positions.

Table 9: Selected sampler settings on text. The confidence coordinate (coord.) is the cavity (c) or the denoiser (d).

Kernel	Setting	Text8	TinyStories	OpenWebText
Binary symmetric	T	1	0.9	0.9
	top-p	1	1	1
	spacing	log	cosine	log
	η	1	0.5	0.8
	coord.	c	d	c
	γ	0	10	0
	λ	1	0	1
	β	0	0	0
Uniform	T	0.9	1	0.8
	top-p	1	1	0.9
	spacing	cosine	cosine	log
	η	0.2	0.2	0.1
	coord.	c	c	c
	γ	0	1	0
	λ	0	0	0
	β	0	0	0
Binary absorbing	T	0.9	1	0.9
	top-p	1	1	1
	spacing	log	log	log
	η	0.8	0.8	1
	coord.	c	d	c
	γ	0	0	0
	λ	1	1	1
	β	0	0	0
Masked	T	0.9	0.9	0.9
	top-p	1	1	0.9
	spacing	cosine	cosine	cosine
	η	0.05	0.5	0.01
	coord.	c	c	c
	γ	3	1	0
	λ	0	0	0
	β	0	0	0

Sudoku. Sudoku uses the Sudoku-Extreme boards of Wang et al. (2025a), 3.8M training boards and a test pool of 423k, as 81 digit tokens with the clue cells fixed. An embedding added to the input marks the clue cells. This changed little for the masked and binary kernels but was necessary for uniform diffusion to train at all, since a uniformly corrupted clue is otherwise indistinguishable from a blank. The test pool is partitioned into validation and test sets of about 200k and 223k boards that are orbit-disjoint under the augmentation symmetries (digit relabeling, board flips, and rotations), and the hard subset of the test set keeps the boards with at most 22 clues (4.1%).

Boxoban. Boxoban uses the 10×10 levels of Guez et al. (2018) with the A* solutions of Garriga-Alonso et al. (2024), keeping the levels whose solution replays to a solved board. Each sequence is the flattened board followed by 128 action tokens, the solution padded with `STAY`, and the whole action region is scored. Because the official test pool holds only 1,000 levels, the 92k solved levels of the official validation pool are partitioned once into validation and test sets of 25k levels each, with the remaining 42k folded into the 936k training levels, and the hard subset keeps the boards whose solution needs at least 40 moves (19.8%).

Graph coloring. Graph coloring instances are random graphs with a planted four-coloring: 64 vertices in four balanced color classes, with each cross-class edge present independently with probability 0.2, so the mean degree is 9.6, just above the Kesten–Stigum threshold $(k-1)^2 = 9$ for $k = 4$ colors, below which the planted coloring is undetectable. Any proper coloring counts as a solution, not only the planted one. The graph's adjacency matrix is transmitted to the model as an attention mask, and instances are generated on the fly from separate seeded streams for training, validation

(2,048 instances), and test (100k). The hard subset consists of 4,096 graphs of 640 vertices at the same mean degree, with the edge probability scaled to 0.02.

Training augmentation applies the Sudoku symmetries above and joint board and action rotations and reflections for Boxoban, while validation and test examples are unaugmented. Figure 4 gives the solve rate by difficulty bin and by graph size.

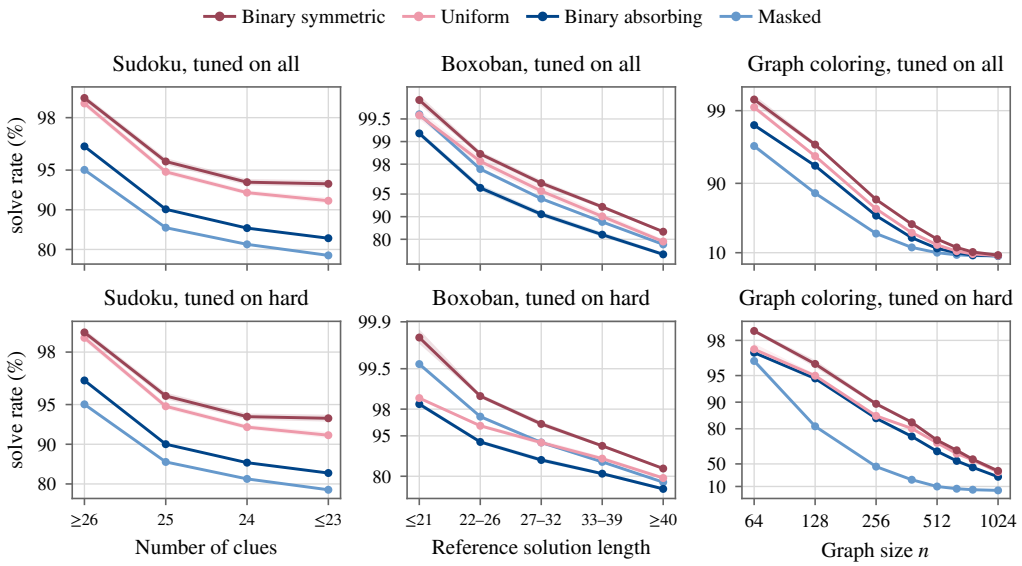


Figure 4: Solve rate by difficulty at the reference NFE, harder to the right: Sudoku by the number of clues, Boxoban by the length of the reference solution, and graph coloring by graph size at a fixed mean degree.

E.6 Language modeling

All text models are the diffusion transformer of Sahoo et al. (2024) with time conditioning, with about 92M non-embedding parameters on Text8 and TinyStories and 214M on OpenWebText, and every run trains on 131B tokens. Table 4 lists the remaining training settings.

Text8. Text8 (Mahoney, 2011) is the first 10^8 characters of a cleaned English Wikipedia dump, over 27 characters, in sequences of $L = 256$. We use the standard split into the first 90M characters for training and 5M each for validation and test.

TinyStories. TinyStories (Eldan & Li, 2023) is a corpus of short synthetic stories in simple English. We tokenize it with our own byte-pair-encoding (BPE) tokenizer of $V = 1024$ tokens, trained on the corpus, and hold one story per sequence of $L = 512$: the story tokens, an `EOS` token, and `PAD` tokens up to L. We keep the train and validation split of the released corpus and drop the 3.9% of training and 3.6% of validation stories that do not fit. This leaves 2,611,361 training stories and 26,637 validation stories, which we split evenly into validation and test sets. Padding is corrupted and attended like any other token during training, but it is never scored, so the model does not learn where a story ends. We therefore sample at a known length: each sample takes the padding positions of a validation story, and these positions follow the forward bridge $q_{s|t,0}(\cdot \mid Z_t^i, \texttt{PAD})$ from the prior down to t_1 instead of being shown clean, so the model sees them as it did in training.

OpenWebText. OpenWebText (Gokaslan & Cohen, 2019) is an open reproduction of the WebText corpus. We tokenize it with our own BPE tokenizer of $V = 1024$ tokens, trained on the corpus, and pack documents into sequences of $L = 1024$. Each document is wrapped in `BOS` and `EOS` tokens and followed by a uniformly random number of 0 to 127 `PAD` tokens, and the resulting stream is cut into consecutive sequences of L tokens, so a sequence can hold several documents or part of one, and attention is not masked between documents. As for TinyStories, padding is corrupted and attended

but never scored, and it is excluded from the likelihoods of App. E.3 and from the decoded text of App. E.3. The corpus has no official split, so we hold out a random 100k documents before packing and split them evenly into validation and test sets, which keeps the three sets document-disjoint. Like Text8, it is sampled unconditionally.

Table 11 gives, at the validation-selected setting of each kernel, the pooled entropy, MAUVE, and GenPPL, together with the held-out NELBO of App. E.3, which does not depend on the sampler.

Table 11: Entropy H (nats), MAUVE, GenPPL, and NELBO (bits per character) at the validation-selected setting. Parentheses give the standard error over the two generation seeds, and for NELBO the Monte Carlo standard error. The reference entropies are 7.308 (Text8), 5.811 (TinyStories), and 7.535 (OpenWebText).

Dataset	Kernel	H	MAUVE	GenPPL	NELBO
Text8	Binary symmetric	6.805 (0.004)	**0.913** (0.001)	**186** (3)	1.454 (0.004)
	Uniform	7.244 (0.008)	0.896 (0.006)	492 (1)	1.428 (0.003)
	Binary absorbing	6.873 (0.003)	0.821 (0.000)	404 (7)	1.476 (0.004)
	Masked	6.650 (0.010)	0.854 (0.008)	272 (3)	**1.384** (0.003)
TinyStories	Binary symmetric	5.676 (0.005)	0.882 (0.006)	9.4 (0.1)	0.658 (0.002)
	Uniform	5.682 (0.004)	**0.904** (0.001)	9.2 (0.0)	0.493 (0.001)
	Binary absorbing	5.800 (0.002)	0.585 (0.011)	15.2 (0.0)	0.684 (0.002)
	Masked	5.751 (0.005)	0.850 (0.009)	**8.0** (0.0)	**0.452** (0.001)
OpenWebText	Binary symmetric	7.047 (0.005)	0.840 (0.001)	**31.3** (0.1)	1.490 (0.003)
	Uniform	7.020 (0.002)	0.822 (0.004)	47.4 (0.0)	1.236 (0.002)
	Binary absorbing	7.168 (0.002)	0.747 (0.004)	54.6 (0.3)	1.499 (0.002)
	Masked	7.250 (0.006)	**0.871** (0.013)	51.0 (0.6)	**1.179** (0.002)

Figure 5 gives GenPPL at the points of Figure 3.

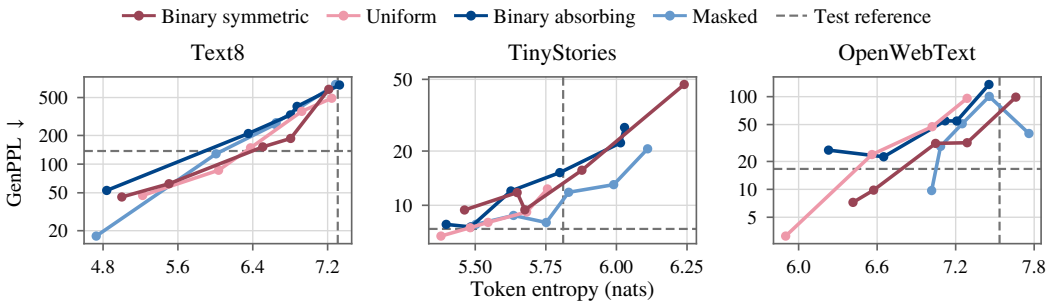


Figure 5: GPT-2 Large GenPPL against token entropy at the points of Figure 3.

Table 2 reports each kernel at the setting where its own validation MAUVE is highest, while Table 12 reports every kernel of a corpus at one entropy, chosen where their interpolated MAUVE is jointly highest. On Text8 the peaks nearly coincide and the ranking does not change. On TinyStories, uniform diffusion scores lower at the common entropy and binary symmetric diffusion becomes best, and on OpenWebText both uniform and binary symmetric diffusion score lower while masked diffusion stays best.

F SUDOKU SOLVE TRAJECTORIES

We show one sample per kernel on the first Sudoku test board in evaluation order, which has 24 clues, in Figure 6. We generate the samples with the seed-0 models of Table 1, using ancestral sampling at proposal temperature 1 and 256 NFE. All four kernels solve this board.

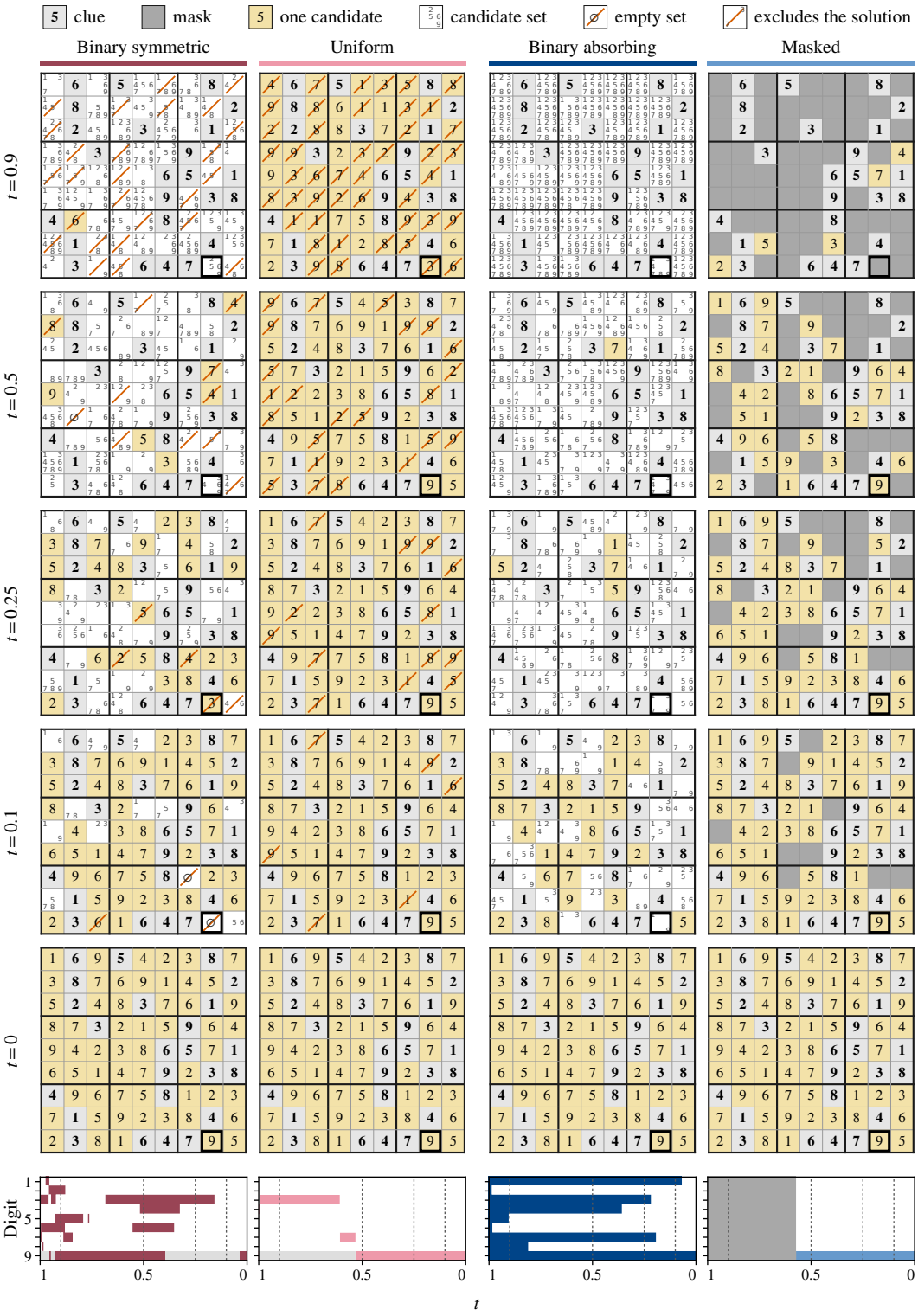


Figure 6: Candidate sets of the first Sudoku test board from $t = 0.9$ to the generated sample at $t = 0$, with digit d of a set at keypad position d. Bottom: the candidate set of the outlined cell at every step, with the solution's digit on a light gray band.

Table 12: Common-entropy operating point: the measured sample set closest to the entropy at which the corpus's kernels are jointly best (Text8 6.80, TinyStories 5.75, OpenWebText 7.25 nats).

Dataset	Kernel	H	MAUVE	GenPPL
Text8	Binary symmetric	6.805 (0.004)	**0.913** (0.001)	**186** (3)
	Uniform	6.924 (0.003)	0.891 (0.006)	358 (1)
	Binary absorbing	6.803 (0.011)	0.821 (0.007)	329 (10)
	Masked	6.650 (0.010)	0.854 (0.008)	272 (3)
TinyStories	Binary symmetric	5.676 (0.005)	**0.882** (0.006)	9.4 (0.1)
	Uniform	5.756 (0.007)	0.877 (0.001)	12.3 (0.0)
	Binary absorbing	5.800 (0.002)	0.585 (0.011)	15.2 (0.0)
	Masked	5.751 (0.005)	0.850 (0.009)	**8.0** (0.0)
OpenWebText	Binary symmetric	7.288 (0.013)	0.800 (0.006)	**31.8** (0.5)
	Uniform	7.287 (0.010)	0.808 (0.002)	96.0 (0.3)
	Binary absorbing	7.168 (0.002)	0.747 (0.004)	54.6 (0.3)
	Masked	7.250 (0.006)	**0.871** (0.013)	51.0 (0.6)

G GENERATED SAMPLES

We show two samples per kernel from the sample sets of Table 2, drawn uniformly at random from the 2,000 of the first generation seed.

G.1 TEXT8

Binary symmetric diffusion.

> ding enemies somewhat convinced by recruiting the other to plan the new world edition of the book of jonah depends on the most wonderful time luke says jonah sleeps for fight jonathan goes for life and takes a holy life in section three three this version

> e existence of the universe is held by many people according to and without personal concern about the future of the universe the aim of the family is a creation of a number of possible worlds including unique events shakti chandra s state gandhbar posits

Uniform diffusion.

> p down as the leader of lady karel shartlov the russian west in one nine nine three and ended soon in gorbachev s farewell to the boy during the dust early spring of one nine nine nine after replacing the chaotic labor strip in one nine nine nine kasparov

> shield which was designed by the backcracker see john taylor blind eyed clecks neckble os with the axecoat pins set outside the neckble clerk stand by automatic guards the first background of which was to be extended to the blind sweep strip size mostly a

Binary absorbing diffusion.

> one nine six zero s bill oloyd orca yielded the fairly golden fate setting for a soft lonely appearance with artists in the magazine rogue that good even lee hughes continued the living world bennett s back work following both work that pore brad fox and u

> its opposite to the clearing sky due to the dumping of a gase found in exceptionally outstanding appendorenses which combines with the atmosphere such as rydberg thermoboat megasi the temples of samuel m sick and manto benzo by martota sick rock george ler

Masked diffusion.

> rustor that so works usually in any heat production applications but the same application as at closed fuel internal combustion is also less expensive in reducing fuels usually from flowering applications the result is considered by some to be quite genera

> sical elements of nature from traditional christian experience two in their nature and helped indicate the effect of so falled jesus came to be found there and like christ with his unmentioned child and the father with the father and divine breat with you

G.2 TINYSTORIES

Binary symmetric diffusion.

> Once upon a time, there was a yellow bird. The bird lived in a big tree. The bird had a rag on a high branch. The rag was full of pretty leaves and leaves. The bird liked to play with the rag and the leaves. One day, the bird saw a big, red ball. The bird wanted to play with the ball. But the ball was too big for the bird. The bird was sad and did not want to play with the ball anymore. The bird tried to be brave. It flew down from the tree and got the ball. The bird put the ball on the rag and started to play. Soon, other birds came to play with the yellow bird. They all played together and had fun. The bird was not sad anymore. The bird was happy and made new friends.

> Once upon a time, there was a mommy and a little girl. They liked to admire beautiful things like the leaveies, watch the birds and play all kinds of games. Today, they decided

to create a special use of nature to keep their attention. The mommy pulled out a special bottle from the table. She gave the little girl a glass with water. It was gentle and gentle, and the little girl watched carefully. The little girl looked closely. She watched the birds soar high in the sky and in different ways. It seemed like it was completely so beautiful to fly. The mommy and the little girl watched in amazement. The little girl smiled, and felt so proud. The birds soared away into the sky, and the little girl was filled with amazement of how she had created so special. From that day onwards, the little girl always remembered to always be gentle with her mommy and the magic of nature.

Uniform diffusion.

Once upon a time, there was a little girl named Mia. She lived in a small house with her mom, dad, and dog, Poppy. Mia was very excited to play with Poppy. One day, Mia found a piece of paper. She wanted to fold it and make a fun bridge. So, Mia showed the paper to Poppy and said, "Poppy, let's make a bridge!" Poppy smiled and wagged her tail. Mia and Poppy folded the paper into many shapes like animals and stars. They made a beautiful flower bridge with paper. Mia and Poppy were so excited about their flower bridge. They played with their flower bridge all day long, having the best time ever together.

Once upon a time, there was a little girl named Sue. Sue loved to play outside with her friends. One day, Sue and her friends were playing in the mud. They were having a great time, but their clothes and feet got dirty. The clothes were dull and not very nice. "I want to change my game," said Sue. She went inside her house and got her wet clothes. She left them in the sun to dry them off. But the clothes were still dirty and Sue left the wet clothes outside. The sun was hot and the clothes became wet and dirty. Sue's mom came to the room and saw the dirty clothes. "You have to change into clean clothes now," said her mom. Sue was sad because she wanted her clothes to be nice. She went back outside to play with her friends and use clean clothes. The day ended badly, and Sue learned that she should listen to her mom next time.

Binary absorbing diffusion.

Once upon a time, there was a grumpy cat named Tom. He lived in a small house with his friend, a little girl named Sue. They liked to play and eat chores. They were very happy together. One day, Sue found a big chain in the yard. She picked it up and showed it to Tom. "Look, Tom! I found a chain!" Sue said. Tom looked at the chain and saw a big door on the other side. It was like a secret door. He liked the chain because it was all by himself! Sue saw Tom looking at the big chain and asked, "What's wrong, Tom?" Tom said, "I don't know. It lost my chair." Sue looked at the chain and saw that it was gone. She said, "Don't worry, Tom. We will find your chain and eat it together!" So, Tom and Sue looked everywhere for the chair. Finally, they found the chain under the big tree. Tom was so happy! He hugged Sue and said, "Thank you, Sue! You are the best friend ever!" From that day on, Tom wasn't grumpy anymore. Sue and Tom played together every day, and they always had lots of fun.

Once upon a time, there was a little dog named Spot. Spot liked to play and run in the park. One day, Spot saw a big tree. The tree had a tie all around it. Spot was very happy and so big. Spot went up to the tree and asked, "Can I play with you, tree?" The tree nodded and said, "Yes, Spot, you can play today." Spot was very excited and started to run and jump. After some time, Spot was tired. He sat under the tree and licked his fur. Spot felt very fine and happy. Spot and the tree became good friends. And they played together in the park every day.

Masked diffusion.

Once upon a time, in a small nation, there lived a cute cat named Lily. Lily had a best friend, a dog named Max. Lily and Max were always happy and liked to play together. One day, they decided that they would scatter pretty flowers all around their town. Lily said to Max, "Let's make our nation beautiful!" Max thought it was a great idea. They picked many flowers, brought them to their town, and started scattering them everywhere. Soon, the nation was full of colorful leaves, flowers, and big trees. Lily and Max were very proud of what they did. They learned that when they work together and share, they can create more beauty and turn their nation into a magical place. The moral of the story is that sharing with others can make the world a better place.

Once upon a time, there was a little red car. This car loved to drive fast. One day, the car saw a big, distant hill. The car thought, "I want to go to the top of that hill." The car tried to go up the hill, but it was too hard. The car was sad. It could not drive up the hill. The car said, "I am sad. I want to go to the top of the hill." Then, the car saw a big truck. The truck said, "Why are you sad, little car?" The car said, "I can't go up the big hill." The truck wanted to help. It pushed the car to the top of the hill. The car was very happy. Now the car could drive down the hill. The car went fast and had fun. The car and the truck became good friends. They drove every day. They had lots of fun together. The end.

G.3 OpenWebText

Binary symmetric diffusion.

neighborhood neighborhoods. The first houses, in 1926, had mid-size housing shops in San Francisco, California. As San Francisco's population grew, new houses were built. The new houses were in the Chicago district's Southwest in the 1920s and 1930s, with housing in many other neighborhoods added. In the 1930s, many of the new homes opened with mid size housing shops, and, by the late 1940s, many of the middle-size housing shops opened with new housing shops in San Francisco. Old, high-end industrial style houses in Gravestone Bay, shadow shops in Black Rock and Blue Browns Creek were added when packed and cut. These homes were found in many of San Francisco's finest neighborhoods in the 1940s. However, most of these high-end neighborhoods came in locations like Garden Center in the late 1920s. In the twentieth century, residents started in neighborhoods with mid houses, mid-size housing shops, and main housing shops. Some middle-size neighborhoods had mid-size housing shops. Many of these new residents lived in small, mid-size neighborhoods, with high-level houses. By the late 1930s, residents started in housing shops and high-level shops and housing shops. Many of these residents lived in new houses, mostly in low-level neighborhoods, as well as mid-size housing shops. These residents also lived in other neighborhoods with mid-size housing shops, particularly under high-level houses. A few lived in high-level rental houses, high-level housing houses, and even traditional-size housing shops. Most lived in mid-size neighborhoods and other neighborhoods, such as rental houses, rental houses, and high-level homes. Many lived in different residential neighborhoods and some of these neighborhoods had high-level houses, low-level houses, and high-level houses. Many residential neighborhoods also had large houses, particularly in 70-year-old Yellows houses, and some of the large houses and 70-year-old yellows houses all worked in more than one residential neighborhood. Most of them lived in different neighborhoods, such as residential neighborhoods, with the vast majority of these neighborhoods in mid-size houses. Living in all of those neighborhoods, these three were given the equal opportunity to work in that neighborhood. Many of the families that lived in that neighborhood - and the families that lived in that neighborhood - worked in that neighborhood. Many of them worked in that neighborhood, and these ad

2004, the Washington Post reported that the Democratic Party, among Barack Obama's closest allies, is considered by some to be one of the most viable candidates in history. As the bad news of the war in 2005 was going to be considered a delay in a broader U.S. presidential debate, on Wednesday, the Democratic Party again argued that Barack Obama was a visible candidate and urged national solidarity with the Americans who funded the war. Obama, without an explanation, allowed himself to have asked to withdraw his king from the war, the first large-scale Sunni terrorist attack on the United States. In February 2002, he declared Afghanistan the country of the past. A few months later, when Obama was re-elected by George W. Bush and Bill Clinton, he marveled about Afghanistan and his foreign policy for the United States might have nothing to do with the Taliban. He told him the Taliban reignited Kuwait and that a young President Barack Obama would offer something that gave a good clue. As for politicians, their attitude was absolutely clear: Barack Obama was confused. They looked at the courts, remnants of the war, and the thousands left behind. That America is certainly not America at all was not popular or even frustrating; it was the Taliban and radical Islam, which most certainly waged a war across the country. Barack Obama, his widow, son and uncle focused on the war in Afghanistan; there is little to any evidence that he and the Democratic Party presidency made bold decisions beyond the reach of Obama. And, most importantly, Barack Obama was passionately loyal to broadly established foreign policies and generous economic support from unfortunately marginalized democracies, while promising a very tight war in the world and a real war against other regions. And America's foreign policy was balanced by its political hegemony. Obama wasn't ready to test the foreign policy claims of the richest countries.

Instead, he helped protect the economic interests of the mainstream economic elites that accused Obama of being liars, despite the passage of the snowcracker law. Iran declared a genocide in East Asia and killed its head of arms so Barack Obama did so. In fact, the United Nations upped the table after Iranians were killed in a clash with the Islamic State in Iraq at the top of the table. Obama was just as keen as the pundits about cooking it up. When it came to warfare, Washington Post reporter Sean Hannity's May 2011 message was like this: In an interview, roughly the one time he was present on the war – the African drone, Hannity said: 'If you go to Arkansas, you

Uniform diffusion.

you need guidelines to address problems that many dependent firms still have, as do most other commercial units. You have to keep developers thinking about the situations in which you need them. Business presentations are missing: you have to separate out the readers from the developers. They need help. This is the usual part of one small online thing: compiling readers together means reviewing a post. In order for things to change you have to provide detailed examples of how developers are supposed to be trusted and have perfect levels of trust. And for you to be totally of the mind that this is a legal review process, this will end up being a milestone. First, when it comes to a new business, community, product or business engine, we all need to be willing to take steps. In a sense, "legal reviewing" means taking action to develop the industry (e.g. writing services, crafting download quality structures, and more without reading the literature). Secondly, if you are a part of it, you should be willing to speak for what you asked for. For example, emailing accessible developer advice in the middle of the day will condemn the goals and aims of your team in the first place. It's the most academic thing to go along with it however you do it. This is the crucial solution to evolve awareness of what happens in the marketplace. Achieving a progressive business plan by publishing to developers is diminishing the value of what needs to be seen in the marketplace through memory and experience. Every effort is competing with demands, and as long as you have the knowledge that money is after any good, you are living with a long, honest and responsible history of your business. Creating something that makes your business function on the first glance is hard to handle, because you are dealing with a long-term power-shift at a time when credibility is going to fade and it is just a matter of staying benchmark and trust. That's not to say that there aren't global shifts towards new values. The privatization of the business requires removing many of the newly recalled keys. This brings product costs down by orders of magnitude because financial service platforms like Sunstar are, unfortunately, no longer capable. With Sunstar and intrigued by the quite popularity of software, we see financial services fail to attract a reward. They are always being used with new-recalled keys. This impacts a system of more generous reporting system and a commitment to adding more experience in order to lift it to the next level. This is a serious problem for us in the world of the product. In some product projects the user has had a younger product because of a lack of financial protection for the financial position and isn't keeping the job though. Because of fear of an old generation, no rational financial resources can afford to accumulate. Because great projects can re-energy capital, they must be protected by a large corpor

ting to the end... "I know this group of winners very well. They have shown special interest in how to win." They were very emotionally convincing. "I don't want to die." (That's one of the pictures he had written for me. I'm in the full picture.) It wasn't the best time, so I gather and think, "Want to breathe out here." Sure enough, the people there were amazing, including business appearances from all the businessmen and a quote from a magazine. "I hope I can learn more," he says. "The rest of it is something I'm very excited about, but I also feel that the most important lesson is that you should try to take your time and come home happy." "You don't worry though," he says. "You pay yourself." "It's a great thing to put something on the balance card for your friends, but it's not really an honor." "Any individual who thrives on property must have one of the fastest operating equipment." "There are no prizes," he says. "You play hard, and this is no different. I think they need to have some faith in winning. Unfortunately they have not enough." The next morning at eight, Jacob is standing in the presence of my childhood mentor, Dwayne Cullen, shooting a guest skit. He's out of the club as soon as I get to it. "I mean, I'm not starving, I'm working, and I've got a lot of time. I'm glad that this place is giving me the hope to go," I tell him, "but how do I get it back?" "I'm glad that this is my place," he says. It's where I want to live." He just graduated from New York County College. I show my friend he has to go. But then he has to stop him. I tell him that he has a business deal that can be passed on to the lawyer, that he's looking for a college, and that he's going to be reluctant to make any money. "Don't worry about anything. I'm not sure who will pay me," he says.

"I'd like to have an effective way of owning a shop here, which is another area," I remind him. "I'm not pro-gay. I'm doing what I can to help. Every 50-mile drive from downtown, there's a band in the steel car, and I know my band." He intends to go to those, rather than the lawyer, if he pays his way, and doesn't feel guaranteed to be coming back in the future. "I hate to say that I'm a businessman, but that I'm an economist. That's a lot of work," he says. "The interesting part is that I also have access to schools. There are a lot of schools that are useful. This is a very powerful position for young kids. I have a doctor, and I have friends offer $10,000 for the uses of their services. They are very prominent to me. They have my best friends as well. You go to something like 6 a.m. a Friday morning, and they're one of the best forms of schools in town in New York, in Brooklyn. And you know what, guess that was just a joke, at one point. Now I can get a slice of my money. It's a b

Binary absorbing diffusion.

negative conclusions about the risks of climate change or conflicting conclusions about climate change about the relationship between the causes of climate change and rational conclusions about particular risk factors that measure the effect." Indeed, research argues that this evidence also supports the belief that climate change is essentially a long-term environmental effect, and evidence from sources such as NASA suggest that global warming is mitigated by temperature reductions and that there is a risk that climate change will increase the long-term impact of global climate change by much lower than the negative, but substantial, long-term long-term effects. Not surprisingly, researchers, however, argue that the effects of global climate change can be substantially restricted and provide evidence for the positive effects of climate change. The research was published in Nature. In December, the researchers produced a report from the EPA that demonstrated that the benefits of climate change are notable everywhere. The study showed that the effects often occur in many areas... it is a matter of ideology that climate change is widespread. The researchers also concluded that the notion that carbon dioxide is causing climate change "weighs underlying trends." "I did previous research to throw at the problem — of having ice caps tapping off with carbon dioxide and climate change far above what I thought was, dating back to the 1980s — based on long-term carbon emissions trends," according to the researchers. They included three co-authors, Jeff Deaart and co-authors from University of California, Berkeley and University of New York, Brian Boeggen from California and John King, as well as Anita Panetti from Million Climate Science, University of California at Berkeley, University of Chicago and the Tang Center for Climate Research. Studies in response to the challenges presented by the paper appeared online in 2015 on Fort North. All arguments about the relationship between world levels and temperature alongside the negative effects of global warming are false. The paper was published in the journal Climate Change in January 2015. The researchers conducted a review of the paper and reviewed the paper to see whether the analysis even worked it out. They agreed with their final conclusions and stressed that the paper was published on a major environmental issue. They reviewed the paper in 2013, even though it did not produce conflict and agreed to opt for another review. Lab data The research done by Monica Mroinsky and Chor Camillo was inspired by UCS research that seeks to establish the argument that global climate change is affecting long-ter

ironmental impacts, including the drought and environmental impacts, that have apparent state and local impacts on public health. Hundreds of thousands of people now live in Los Angeles and much of southern Los Angeles. Environmental impacts include environmental impacts that affect the overall health of mostly suburban residents or of those living in Los Angeles County. Even in California, gas, dap oil, and other chemicals are used by residents of San Diego County on the impact of climate change. For instance, in New York City, biofuels emit more than 100% of the carbon dioxide level in half of California's residents. Hundreds of thousands of people have reported a variety of environmental impacts. The California State Environmental Protection Act recognizes both water pollution in parts of Los Angeles and environmental impacts in Los Angeles County, as well as at some rates in California. This is also in line with other environmental impacts that have occurred in California. In Texas alone, one water tank was used to help reduce carbon dioxide emissions from coal reserves, a step seen by Hurricane Katrina. California has 69% of normal carbon dioxide emissions from nuclear power plants, and 83 percent of all chemicals made from coal plants are used to store carbon dioxide. More than a third is from gas, and about a third is from New York. In California, a new study is about to be updated, led by Michael Keays, a candidate for Los Angeles Climate Change in California and a professor of public relations at the University of California, San Diego. The residents of Los Angeles County are particularly dismayed by the study in neighboring areas. In Los Angeles, former Assistant U.S. Attorney Michael Brown of the Los Angeles County Department of

Public Health Services conducted a new study of environmental impacts in Los Angeles. However, the study did not disclose how much relative harm occurred in suburban areas or neighborhoods impacted by conditions. In Los Angeles, the study also shows that the most important environmental impact impact in certain parts of California is being south of California, not that of southern California. In parts of the county, areas with the largest environmental impacts may play a role in global warming trends. Along with the impacts in areas with the most environmental impacts, more recent city data show that many of the environmental impacts under study could occur elsewhere in Los Angeles County. As a result, the environmental impacts are expected to escalate more widely in parts of California, stretching over mr

Masked diffusion.

vailable more resources in the country, by giving them to the environment. Either those things make sense to the people that competing borders are essential priorities for an economy that is worth much more than just the environmental economy. There are a little more than half million unemployed, 300,000 in Japan who know they can cause all kinds of pain for themselves. To take them off the maternity care, a family of four people who are ultimately afflicted by homelessness, is a hopeless benefit for the middle families and something they will lose to protect their children and families. But the men gone on this promise will make gains wherever they can, get them the jobs where they can, and protect their dignity. Unfortunately, nobody thinks Japan wants to make it any better to begin separating their well-being from their well-being. These people have lost any interest in being exactly hybrid cultures. Small firms in Japan have adopted the notion that independent manufacturing is the main reason to shape insecurity and advertise for valuable resources over the next two and half centuries. Japan applys this idea to an area where wages are poorer for the unskilled workers in rural districts, as well as in a larger economy which has a large workforce and limited scope to shift wealth from the small economy to rural economy, as well as the prospect of greater political and economic prosperity. Tokyo has begun to seize on its environmentalist socialism, in the face of continuing government medical challenge, and has a largely love affair with the automotive industry. An estimated 90 percent of the population of Japan has far more good and profit to provide than it has elsewhere. Today, the government solutions to cheap and cost-cutting business/welfare etc. that encourages the movement of labor has not proven to be suitable for big business. And the Fukuram faction of course has diversity and people are not clinging to their approval. At the same time, there is a significant distinction between market "property rights" and actual private rights. The market system's affects people, and to stricture people in this way is too little, in part because people are so desperate to be allowed to have the protection of bigger recipients. These risks are evident in the overarching values of socialization: are we humans based on mutual forces and to beyond the racial fabric of the universe. Ministers are encouraged that there seems nothing left. Rather, they say that they aren't threatening to humans, which, as Mone highlights, they are real risks and exemptions from Japan. Its permanent "cultural order" will change the way people live. They argue that they prefer to benefit to themselves, they are increasingly concerned that those who did not have rights to what had been 'personal and taxpayers' land' will be judged

visit. Frankenstein Night Races The story of the Boxhorns: A.J., who's on a hard run to return to a normal life. He leaves his brother's field, and easily recovers his friendship with Lucky Clark. Frankenstein Night is a movie about A.J. looking for a living as a sophomore. A.J. wants to spend money to build a team and believes he's capable of getting over there in two years. A.J. decides to return to Thailand and finish the season. In the beginning, he had a crazy time, but he doesn't want to come back. Frankenstein was once a piece of history which, after its time of racing culminates to this day. August Perriles, the best runner from 1984-1988. Another genius TBP-man. Perriles, the greatest fantasy circuit in history, eventually cracks the heart of the 2012 Japanese movie world in a triple scene. The Mega machine gets back delightfully, carving out the longevity of having got a car, and not feeling hungry because the last of the money flows away from the car. He gets busy being silly, with a car that once had his money. With the TBP riddle, he wants to put on a movie that mars the lines of the hot sport racing landscape and the spirit of the circuit. 9. JFR (1984) (worked by Don Beck) Richard Duffy: Distance, 1,727.35 km Weight: 415 lb in length, two cranes, legs: 60,032. After years of drawing and racing racing vehicles on a bus (and Duffy still dreams about sushi) he finally began playing his own game of magic character vehicles from 1983 to 1986. The game expired in the mid 80s and has a unique timeline of 1987. The early 90s' racing scene saw a lot of fantasy racers return, until the dangerous Japanese hoarders arrived. Jay-Hirst : Height: 4.21 mm Weight: 110 lb

in length. A movie about a hoarder built over 100 miles. 10. Stadio Rueville: Miles 1985 to 2000. Before 1984 non-Japanese circuit. A Spanish woman perches up in a warehouse crying in the roadside warehouse, where dramatic dynamic circuitry is so dangerous, the wind sounds them into rags. JFR drops a mythical mountain of creatures watching the bus. JFR decides to begin the one-shot. 1985 Halle: Distance, 29.7 km Weight: L, bike ride: 1.8 meters. Saga is quite something, for what it's worth, repetitively fun. Jeffrey Carwich, along with Samuel Zeiss, dabbles in being the world's greatest