LEDFLOW: INTRODUCING ENTROPY-GUIDED GENERATION ORDER INTO UNIFORM DISCRETE FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Uniform discrete flow permits repeated updates at every generation position. While continued revision supports correction of wrong tokens, it also exposes correct intermediate predictions to later errors. An experiment on Sudoku puzzles shows that 9.4% of generated cells are correct at an intermediate step but incorrect in the final output. We introduce generation order into uniform discrete flow through selective absorption, which fixes chosen predictions while preserving the uniform-flow velocity at active positions. To prioritize reliable predictions for absorption, we propose Low-Entropy Discrete Flow (LEDFLOW), a training-free sampler that adaptively orders absorption by local entropy. By decomposing absorption error into joint dependence and conditional prediction terms, we show that, under entropy-error regularity, selecting the lowest-entropy positions under a fixed absorption count minimizes an upper bound on the conditional term. We further analyze sensitivity of global lookahead, whose worst-case decision-error bound grows with lookahead window under an imperfect denoiser. Across reasoning benchmarks, LEDFLOW attains 0.845 Nikoli Sudoku solve accuracy, with largest gains on strongly constrained tasks. On a text-to-image generation benchmark it attains the best overall score among decode-time samplers, and on multimodal understanding it improves over the default sampler on all six benchmarks, at an inference cost comparable to standard flow sampling.

1 INTRODUCTION

Discrete flow matching allows multimodal generation by transporting a source distribution to a target distribution over discrete states through a continuous time probability velocity [10], while uniform discrete flow progressively transforms every token in each time step under a uniform source. FUDOKI [40] applies this framework to multimodal understanding and generation using a metric-induced probability path and a kinetic-optimal velocity. Conditioned on a target token, the velocity moves probability mass toward tokens closer to that estimated target, so that generated positions can be updated repeatedly as sampling proceeds.

Continued revision can replace correct intermediate tokens with incorrect predictions, especially near the end of sampling. Under the kinetic-optimal formulation, the schedule gradient diverges as t approaches the endpoint. The outgoing transition rate then grows without bound when a token is semantically distant from the estimated target. This late denoiser error exposes an otherwise correct token to a large transition rate. A Sudoku trajectory diagnostic (Fig. 2) using the kinetic-optimal update shows on 100 puzzles with 64 sampling steps that 9.4% of generated cells are correct at least once during sampling but incorrect in the final output. This motivates us to introduce generation order into uniform discrete flow through selective absorption. We equip each position with a binary absorption indicator that sets its outgoing rates to zero once absorbed, allowing reliable predictions to remain fixed while the other positions continue evolving. However, fixing an incorrect token prevents correction, and an arbitrary order does not distinguish reliable predictions from uncertain ones. Therefore, the absorption policy must account for this irreversible prediction risk.

To address this ordering problem, we propose Low-Entropy Discrete Flow (LEDFLOW), a training-free sampler that introduces generation order into uniform discrete flow through entropy-guided absorption (Fig. 1). At each sampling step, LEDFLOW selects the prescribed number of active positions with the lowest predictive entropy and absorbs them with the most likely predicted values. The

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

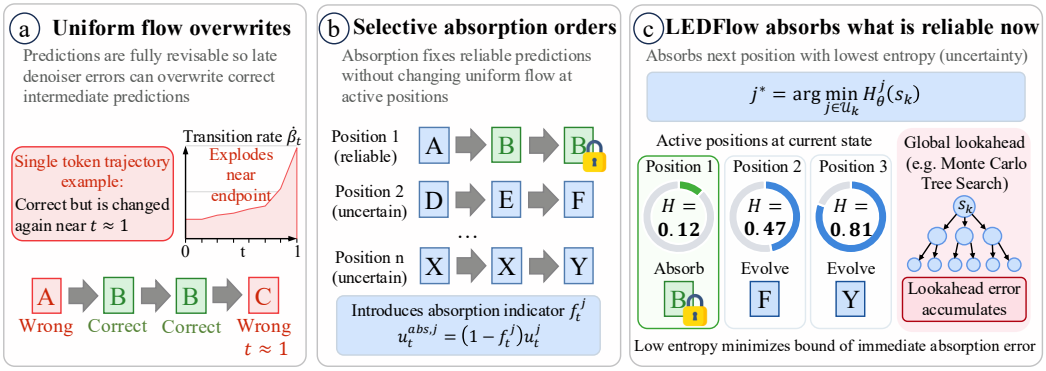


Figure 1: **Entropy-guided absorption in uniform discrete flow. (a)** Late revisions can overwrite correct predictions. **(b)** Selective absorption fixes chosen predictions while other positions continue evolving. **(c)** LEDFLOW prioritizes low-entropy positions for absorption; global lookahead aggregates potentially noisy estimates across the remaining positions.

remaining positions continue to evolve under the kinetic-optimal velocity, with subsequent predictions conditioned on the values already fixed by absorption. At each step, the sampler recomputes entropy for an adaptive order, allowing the policy to preserve the uniform flow velocity at active positions while prioritizing low-entropy predictions for absorption. This reduces the above correct-to-wrong reversions from 9.4% to 2.6% of generated cells and the fraction of token changes occurring after $t = 0.75$ from 46% to 15%.

We then provide a conditional analysis of LEDFLOW by analyzing the error introduced by irreversible absorption. Absorbing several positions independently can discard their dependence, while previously absorbed values affect the context of subsequent predictions. We characterize the stepwise absorption error through a Kullback-Leibler (KL) decomposition into joint dependence and conditional prediction terms. We show that selecting the lowest-entropy positions, under entropy-error regularity on oracle-reachable states and a fixed absorption count, minimizes an upper bound on the conditional error term. We provide further analysis on global entropy lookahead [44] which evaluates candidate absorptions through predicted successor states. With an imperfect denoiser, aggregating future estimates can compound errors, and the resulting worst-case decision-error bound scales with the lookahead window. This motivates an empirical comparison with local predictive entropy.

Experiments evaluate LEDFLOW across tasks with increasing structural dependence. The gains are largest on constrained reasoning, where LEDFLOW attains 0.845 Nikoli Sudoku solve accuracy and attains the best average rank across the nine reasoning tasks. On text-to-image generation LEDFLOW attains the best GenEval overall score among decode-time samplers (2.8% absolute over the native sampler), and on multimodal understanding it improves over the native sampler on all six benchmarks. We also examine the behavior motivating the analysis: entropy-guided absorption outperforms other absorption policies in puzzle solve accuracy. As the lookahead window increases, the mean absolute score error grows from 0.17 with a single window to 1.76 under full lookahead, while accuracy peaks at eight windows and then drops to 0.31. Lookahead therefore costs about eight times the inference runtime without bringing a significant reasoning improvement. Our contributions are:

- We analyze rapid terminal transitions and correct-to-wrong revisions in uniform discrete flow, and introduce explicit generation order through selective absorption to preserve chosen predictions (Sec. 2.2, 3.1).
- We develop LEDFLOW, a training-free entropy-guided absorption policy. Under entropy-error regularity on oracle-reachable states and a fixed absorption count, it minimizes an upper bound on the local conditional absorption error. We also show that a worst-case global lookahead decision-error bound grows with its window (Sec. 3.2, 3.3).
- We evaluate LEDFLOW across reasoning, multimodal understanding, and image generation. Controlled experiments compare absorption orders and measure how lookahead window size affects score error and position selection, while runtime measurements and transfer comparisons against each model's default order assess efficiency and generality (Sec. 4).

2 PRELIMINARIES AND MOTIVATION

2.1 PRELIMINARIES

We generate a target sequence $a_1 = (a_1^1, \dots, a_1^M) \in \mathcal{S}^M$ over a finite vocabulary $\mathcal{S}$ and M generated positions. Let x denote the fixed input, set $\mathcal{C} := x$, and write $q(a_1 \mid \mathcal{C})$ for the target distribution, $\mathcal{C}_t$ the conditioning information available at time t, q and p^q the oracle (data) distributions and their conditionals, p_t the prescribed probability path, and $\mathbb{Q}^\theta$ the learned denoiser posterior. Discrete flow matching transports a source distribution p_0 to the target over $t \in [0, 1]$ under a Continuous Time Markov Chain (CTMC) [4, 10, 29]. Let a and z denote the current state and a candidate destination state, respectively. For any state x, let x^{-j} denote all coordinates except j. We use the discrete delta probability mass function $\delta(x, y) := \mathbf{1}\{x = y\}$. A factorized probability velocity is $u_t(z, a) = \sum_{j=1}^M \delta(z^{-j}, a^{-j})\, u_t^j(z^j, a)$, where u_t^j gives the instantaneous transition rate of coordinate j. For a conditional path factorized on coordinates $p_{t|1}(a_t \mid a_1) = \prod_j p_{t|1}^j(a_t^j \mid a_1^j)$, the marginal coordinate velocity follows from posterior averaging: $u_t^j(z^j, a, \mathcal{C}_t) = \mathbb{E}_{a_1 \sim p_{1|t}(\cdot|a,\mathcal{C}_t)}\Big[u_t^j(z^j, a^j \mid a_1^j)\Big]$. At generation time, the reverse process $p_{1|t}$ is approximated by the learned denoiser $\mathbb{Q}_{1|t}^\theta(\cdot \mid a_t, \mathcal{C}_t)$. FUDOKI [40] implements the uniform distribution framework in [10] with a metric-induced token path: $p_{t|1}^j(a^j \mid a_1^j) = \mathrm{softmax}_{a^j \in \mathcal{S}}\Big[-\beta_t d(a^j, a_1^j)\Big] \propto \exp\Big[-\beta_t d(a^j, a_1^j)\Big]$, and d corresponds to the token distance between embeddings, $\beta_t = c\,(t/(1-t))^\alpha$ so that $\beta_0 = 0$ and $\beta_1 \to \infty$, allowing the source to be uniform over the vocabulary at $t = 0$ and the path to concentrate on target token as $t \to 1$. By denoting $\dot\beta_t$ the derivative of β_t, the kinetic optimal velocity is

$$u_t^j(z^j, a_t \mid a_1^j) = p_t(z^j \mid a_1^j)\, \dot\beta_t \Big[d(a_t^j, a_1^j) - d(z^j, a_1^j)\Big]_+, \tag{1}$$

where $[\cdot]_+ = \max\{\cdot, 0\}$. Probability mass moves from a_t^j to z^j only when $d(z^j, a_1^j) < d(a_t^j, a_1^j)$, allowing the flow to progress monotonically toward the target value a_1^j supplied to the velocity field [40]. However, at inference time where the true target is unavailable, FUDOKI's Euler solver instead samples a predicted target $\hat a_1^j \sim \mathbb{Q}_{1|t}^\theta(\cdot \mid a_t, \mathcal{C}_t)$ at every step and measures the velocity (Eq. 1) at that sample, so that every generated position, including those already holding a reliable prediction, is resampled at every step. Detailed formulations from existing work are provided in App. B.1.

2.2 MOTIVATION AND ORDERED ABSORPTION

The schedule gradient $\dot\beta_t$ in Eq. 1 diverges as $t \to 1$, which increases transition exposure at positions that disagree with the predicted target. With denoiser error, this exposure costs correctness.

Proposition 1 (Terminal transition exposure)**.** *At time t, consider position j whose current token differs from its target $a_t^j \neq a_1^j$. The total outflow rate under Eq. 1 is $\lambda_t^j = \dot\beta_t\, G_t^j$, where $G_t^j = \sum_{z \in \mathcal{S}} p_t(z \mid a_1^j)\big[d(a_t^j, a_1^j) - d(z, a_1^j)\big]_+ \geq p_t(a_1^j \mid a_1^j)\, d(a_t^j, a_1^j) > 0$. Since $\dot\beta_t = c\alpha\, t^{\alpha-1}(1-t)^{-\alpha-1} \to \infty$ while $p_t(a_1^j \mid a_1^j) \to 1$ as $t \to 1$, the rate diverges, $\lambda_t^j \to \infty$, and the corresponding frozen-rate jump probability for a fixed step size $h > 0$ satisfies: $1 - e^{-h\lambda_t^j} \to 1$. When the predicted target $\hat a_1^j$ is redrawn, disagreement with it exposes the current token to this increasing rate.*

To assess net effect on accuracy between correction and incorrect revision, Fig. 2 illustrates the latter by pretraining a flow model with Sudoku puzzles and evaluating on a held-out set of 100 puzzles with 64 sampling steps: 46% of token changes occur after $t = 0.75$, with 9.4% correctly generated cells replaced incorrectly at the end. This results in a puzzle solve rate of 41.0%, exposing a tension between continued revision of wrong tokens, and the preservation of correct intermediate predictions.

Selective absorption addresses this tension by making chosen predictions terminal while the other positions keep evolving under the native flow, but it introduces a new problem: deciding which predictions are reliable enough to be made irreversible. To prioritize reliable predictions at the current state and reassess remaining positions as the context changes after each absorption, the denoiser's own uncertainty is a natural signal: a concentrated (diffuse) posterior has little (more) to gain from further revision. Sec. 3 formalizes selective absorption as an augmentation of uniform discrete flow and turns this principle into an ordering policy driven by local entropy. App. B.2 provides the proof and scope of the proposition, together with the training and evaluation settings of this diagnostic.

3 LEDFLOW: LOW-ENTROPY DISCRETE FLOW

3.1 SELECTIVE ABSORPTION AS AN ABSORBING-STATE UNIFORM FLOW

We add an inference time indicator function into uniform discrete flow to change the sampler law. Specifically, each generated position is augmented with an absorption indicator $f_t^j \in \{0, 1\}$. Every position starts active with $f_0^j = 0$ and follows the uniform-flow velocity. Once $f_t^j = 1$, its outgoing transition rates are set to zero, hence keeping uniform velocity at active positions unchanged while fixing absorbed positions:

$$u_t^{\theta,\mathrm{abs},j}(z^j, a_t, f_t, \mathcal{C}_t) = (1 - f_t^j)\, u_t^{\theta,j}(z^j, a_t, \mathcal{C}_t), \qquad z^j \neq a_t^j. \tag{2}$$

An absorbed position is a fixed point of the sampler regardless of what the denoiser later predicts for it, while every active position keeps the kinetic-optimal velocity of Eq. 1, now evaluated at a state that contains the absorbed values. The augmentation leaves two decisions to the sampler: how many positions to absorb at each step (absorption count), and which positions to absorb (absorption order). While the absorption schedule can be fixed in advance, the order determines which predictions become irreversible and which values condition the rest of the generations (Sec. 2.2). Under this formulation, we introduce LEDFLOW, an absorption-ordering policy guided by local entropy, and analyze its local absorption risk. App. C.1 details the augmented process and its training-time counterpart for the puzzle denoisers.

3.2 THE LEDFLOW SAMPLER ABSORBS BY LOCAL ENTROPY

LEDFLOW selects active positions with low predictive entropy, absorbs their most likely values, and continues evolving the remaining positions under the uniform-flow velocity (illustrated in Algorithm 1). The absorption indicator in Eq. 2 addresses late revisions in Sec. 2.2 by retaining selected predictions and preventing further transitions. Let $s_k = (a_{t_k}, f_{t_k}, \mathcal{C}_{t_k})$ be the sampler state before absorption and $\mathcal{U}_k := \{j : f_{t_k}^j = 0\}$ the active positions. Write $\mathbb{Q}_{1|t_k}^\theta(a_1^j \mid s_k)$ for the denoiser marginal at position j, with predictive entropy: $H_\theta^j(s_k) := -\sum_{z^j \in \mathcal{S}} \mathbb{Q}_{1|t_k}^\theta(a_1^j = z^j \mid s_k) \log \mathbb{Q}_{1|t_k}^\theta(a_1^j = z^j \mid s_k)$. At grid point t_k, let m_k be the target number of active positions and set $b_k := \max\{0, |\mathcal{U}_k| - m_k\}$. The entropy-guided ordering policy selects the b_k active positions with lowest predictive entropy and updates the indicator for every selected position,

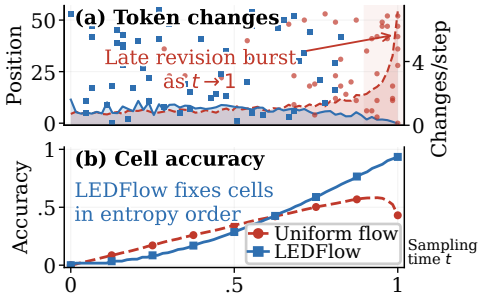
Algorithm 1: LEDFLOW selective absorption over uniform discrete flow

$a_{t_0} \sim p_0, \quad f_{t_0}^j \leftarrow 0 \; \forall j$
for $k = 0, \ldots, K-1$ **do**
  $(a_{t_k}^+, f_{t_k}^+) \leftarrow (a_{t_k}, f_{t_k})$
  $\mathcal{U}_k \leftarrow \{j : f_{t_k}^j = 0\}, \; s_k \leftarrow (a_{t_k}, f_{t_k}, \mathcal{C}_{t_k})$
  Compute $\mathbb{Q}_{1|t_k}^\theta(\cdot \mid s_k), \{H_\theta^j(s_k)\}_{j \in \mathcal{U}_k}$
  $b_k \leftarrow \max\{0, |\mathcal{U}_k| - m_k\}$
  $B_k^{\mathrm{LED}} \leftarrow$ Eq. (3)
  for $j \in B_k^{\mathrm{LED}}$ **do**
    $a_{t_k}^{j,+} \leftarrow \arg\max_{z \in \mathcal{S}} \mathbb{Q}_{1|t_k}^\theta(a_1^j = z \mid s_k)$
    $f_{t_k}^{j,+} \leftarrow 1$
  end for
  Evolve active $j \notin B_k^{\mathrm{LED}}$ to t_{k+1} by Eq. (2); absorbed j stay fixed
  $f_{t_{k+1}} \leftarrow f_{t_k}^+$
end for
return a_{t_K}

$$B_k^{\mathrm{LED}} \in \arg\min_{\substack{B \subseteq \mathcal{U}_k \\ |B| = b_k}} \sum_{j \in B} H_\theta^j(s_k) \Longrightarrow f_{t_k}^{j,+} = \begin{cases} 1, & j \in B_k^{\mathrm{LED}}, \\ f_{t_k}^j, & j \notin B_k^{\mathrm{LED}}, \end{cases} \tag{3}$$

$$a_{t_k}^j \leftarrow \arg\max_{z^j \in \mathcal{S}} \mathbb{Q}_{1|t_k}^\theta(a_1^j = z^j \mid s_k), \qquad f_{t_k}^j \leftarrow 1, \quad j \in B_k^{\mathrm{LED}}. \tag{4}$$

This policy turns each selected posterior into a fixed token prediction, changing the sampler law while retaining the uniform flow update in Eq. 2 for active positions outside B_k^{LED}. At the next time grid, the posterior and entropy are recomputed from the updated state to reassess subsequent decisions based on absorbed values. In the trajectory run (Fig. 2(a–c)), LEDFLOW reduces the fraction of token changes after $t = 0.75$ from 46% to 15% and raises the puzzle solve rate from 41.0% to 84.5%. This ordering policy is training free, as the absorption indicators can be directly plugged into the denoiser of the same checkpoint. Puzzle checkpoints take absorption indicators as an input feature and serve every absorption policy, including no absorption with indicators fixed at zero, which recovers the base uniform discrete flow sampler. We provide a conditional analysis of the selection of low local entropy for absorption order in Sec. 3.3. App. C.2 provides the schedule, update order, and matched-policy details.



(c) Trajectory-run statistics

Measurement	Uniform flow (no absorption)	LEDFlow (ours)	LEDFlow − Uniform flow
Token changes per position	1.49	1.06	−0.43
Fraction of positions changed more than once	0.30	0.13	−0.17
Fraction of token changes occurring after $t = 0.75$	0.46	0.15	−0.31
Mean sampling time of a token's last change	0.67	0.48	−0.19
Puzzle solve rate	0.410	**0.845**	+0.435
Trajectory correctness (fraction of cells)			
Correct to wrong	0.094	**0.026**	−0.068

Figure 2: **Ordered absorption reduces late revisions and improves puzzle solving. (a,b)** Token-change timing and cell accuracy on 100 Nikoli puzzles with $K = 64$. Without absorption, cell accuracy falls rapidly from 0.58 to 0.43 at $t \approx 0.94$, whereas LEDFLOW rises monotonically. **(c)** Statistics from the trajectory run: correct-to-wrong reversions decrease from 0.094 to 0.026 of all generated cells (cells correct at an intermediate step but incorrect at the endpoint).

3.3 THEORETICAL ANALYSIS

This section analyzes local entropy selection under an entropy-error assumption. We first characterize the error of irreversible absorption, then establish the conditional-error upper bound minimized by low-entropy selection, and compare the sensitivity of local and lookahead scores to estimation error.

Error induced by absorption. Noting that absorption is irreversible, its cost depends on the denoiser during each absorption process. At an oracle-reachable sampler state s_k, where the absorbed partial assignment has positive probability under the target distribution, let $B_k \subseteq \mathcal{U}_k$ be the positions absorbed at that step. Given the target distribution q, we write $p^q_{1|s_k}(a_1^{B_k})$ for the oracle posterior, with the denoiser marginal defined in Sec. 3.2. We apply KL decomposition [43, 2], to a stochastic reference kernel that samples selected positions independently from denoiser marginals.

Proposition 2 (Decomposition of local absorption error)**.** *Conditioned on a reachable state s_k and an absorption block B_k, we write the error $\mathcal{L}_{\text{abs}}(B_k; s_k) := D_{\text{KL}}\Big(p^q_{1|s_k}(a_1^{B_k}) \,\Big\|\, \prod_{j\in B_k} \mathbb{Q}^\theta_{1|t_k}(a_1^j \mid s_k)\Big)$, such that*

$$\mathcal{L}_{\text{abs}}(B_k; s_k) = \underbrace{\mathcal{TC}_q(a_1^{B_k} \mid s_k)}_{\mathcal{E}_{\text{joint}}(B_k;s_k)} + \underbrace{\sum_{j\in B_k} D_{\text{KL}}\Big(p^q_{1|s_k}(a_1^j) \,\Big\|\, \mathbb{Q}^\theta_{1|t_k}(a_1^j \mid s_k)\Big)}_{\mathcal{E}_{\text{cond}}(B_k;s_k)}, \tag{5}$$

where $\mathcal{TC}_q(a_1^{B_k} \mid s_k) = \sum_{j\in B_k} H_q(a_1^j \mid s_k) - H_q(a_1^{B_k} \mid s_k)$ is the conditional total correlation of the positions absorbed together.

Corollary 1 (Conditional error bounds the excess absorption risk)**.** *For a reachable state s_k and a block B_k with $|B_k| = b$, let $\hat{z}^j$ be the most likely value under $\mathbb{Q}^\theta_{1|t_k}(a_1^j \mid s_k)$, $c_j = p^q_{1|s_k}(a_1^j = \hat{z}^j)$ its oracle probability, and $c_j^\star = \max_z p^q_{1|s_k}(a_1^j = z)$ the probability of the oracle's best value. Then*

$$\sum_{j\in B_k}(1 - c_j) \;\le\; \sum_{j\in B_k}(1 - c_j^\star) + \sqrt{2b\,\mathcal{E}_{\text{cond}}(B_k; s_k)}. \tag{6}$$

The left side of Eq. 6 is the expected number of incorrect deterministic absorptions in the selected block, while the first term on the right is the unremovable marginal classification risk of the oracle. The square-root term bounds the excess risk due to denoiser error by Pinsker's inequality and Cauchy-Schwarz. While larger blocks reduce sequential evaluations at the cost of unoptimized dependence error, the joint term vanishes for $b = 1$. App. C.3 gives the derivation of the decomposition and the proof of the corollary.

Minimizing the conditional error bound. For the active set $\mathcal{U}_k$ at a reachable sampler state s_k, fix an absorption count $b \ge 1$ and let $\mathfrak{B}_k(b) = \{B \subseteq \mathcal{U}_k : |B| = b\}$. The oracle posterior serves

as an analytical reference for selecting positions with small model error. If it were available, the conditional-error objective would be:

$$B_k^{\text{oracle}} \in \arg\min_{B \in \mathfrak{B}_k(b)} \underbrace{\sum_{j \in B} D_{\text{KL}}\Big(p^q_{1|s_k}(a_1^j) \,\Big\|\, \mathbb{Q}^\theta_{1|t_k}(a_1^j \mid s_k)\Big)}_{\mathcal{E}_{\text{cond}}(B; s_k)}. \tag{7}$$

The oracle posterior is unavailable at inference. To prioritize predictions that are currently reliable, we formulate a reliability condition based on denoiser uncertainty, quantified by predictive entropy. This is motivated by the negative relationship between prediction entropy and reasoning accuracy [44].

Definition 1 (Entropy-error regularity)**.** Let $H_{\max} = \log|\mathcal{S}|$. A denoiser is *$\phi$-regular* on the oracle-reachable sampler states s_k if there exists a nondecreasing function $\phi : [0, H_{\max}] \to \mathbb{R}_{\geq 0}$, with $\phi(0) = 0$, such that $D_{\text{KL}}\Big(p^q_{1|s_k}(a_1^j) \,\Big\|\, \mathbb{Q}^\theta_{1|t_k}(a_1^j \mid s_k)\Big) \leq \phi\Big(H^j_\theta(s_k)\Big)$ for any s_k and $j \in \mathcal{U}_k$.

Applying Def. 1 to the conditional term in Eq. 5 gives, for any absorption block $B_k \subseteq \mathcal{U}_k$, $\mathcal{E}_{\text{cond}}(B_k; s_k) \leq \Phi(B_k; s_k) := \sum_{j \in B_k} \phi\Big(H^j_\theta(s_k)\Big)$. Because ϕ is nondecreasing, the block that minimizes Φ under a fixed absorption count contains the active positions with lowest predictive entropy, so the square root term in Eq. 6 is at most $\sqrt{2b\,\Phi(B_k; s_k)}$, so entropy selection minimizes this surrogate excess-risk bound for a fixed absorption count.

Theorem 1 (Local surrogate minimization and the oracle limit)**.** *Given a ϕ-regular denoiser:* (i) Local adaptive absorption. *For a fixed absorption count $b \geq 1$, a block containing b active positions with lowest entropy minimizes $\Phi(B; s_k)$ over $B \in \mathfrak{B}_k(b)$. In particular, for single position absorption,*

$$j^\star \in \arg\min_{j \in \mathcal{U}_k} H^j_\theta(s_k) \tag{8}$$

minimizes the upper bound on the conditional error of the next irreversible prediction. (ii) Oracle limit. *As the learned posterior converges to the oracle posterior on reachable states, $\mathcal{E}_{\text{cond}}(B_k; s_k) \to 0$ for every admissible absorption block and hence for every absorption policy.*

Part (i) follows from monotone sorting under Def. 1, while part (ii) gives the oracle limit common to all policies. For $b > 1$, the theorem minimizes the conditional-term bound in Eq. 5. The guarantee is local, since each absorption changes the sampler state and subsequent errors. The condition $\phi(0) = 0$ excludes confidently wrong deterministic predictions. Although Def. 1 uses predictive entropy, the analysis extends to any local uncertainty score whose conditional error is bounded by a nondecreasing function of that score, including maximum probability and probability margin under suitable regularity conditions. While it holds for 95.5% of sampled Sudoku states in Fig. 2, where the oracle posterior is computable, we treat it as idealized. App. C.3 provides the proof and the relaxed offset. We next compare local greedy decoding with estimation over future states.

Global lookahead can amplify estimation error. Global lookahead decoding (represented by Info-Gain [44]) evaluates information gain by estimating predicted uncertainty over all remaining active positions. Let $\mathcal{U}(s)$ contain m active positions at state s, and let $h^q_j(s)$ and $h^\theta_j(s)$ denote the oracle and model entropies. Define $R_r(s) := \sum_{j \in \mathcal{U}(s)} h^r_j(s)$, $r \in \{q, \theta\}$. For an active candidate position $j \in \mathcal{U}_k$ and absorbed value z^j, let s^{j,z^j} denote the hypothetical state after absorption obtained by setting coordinate j to z^j. Writing the model and oracle posterior as $P^j_\theta(\cdot \mid s) = \mathbb{Q}^\theta_{1|t}(a_1^j = \cdot \mid s)$, $P^j_q(\cdot \mid s) = p^q_{1|s}(a_1^j = \cdot)$, the corresponding entropy reductions are $G^r_j(s) := R_r(s) - \mathbb{E}_{z^j \sim P^j_r(\cdot|s)}\Big[R_r(s^{j,z^j})\Big]$, $r \in \{q, \theta\}$. Assume that at current state s and every hypothetical one-step lookahead state s' with positive model probability, $|h^\theta_j(s') - h^q_j(s')| \leq \delta$ for every active j, and the total variation $\frac{1}{2}\sum_{z^j \in \mathcal{S}} \Big|P^j_\theta(z^j \mid s) - P^j_q(z^j \mid s)\Big| \leq \epsilon$ for every candidate j.

Lemma 1 (Error amplification in global entropy reduction)**.** *Let $\hat{j} \in \arg\max_j G^\theta_j(s)$, $j^\star \in \arg\max_j G^q_j(s)$, and $\hat{j}_{\text{ent}} \in \arg\min_j h^\theta_j(s)$. Because $R_r(s)$ is common to every candidate at s, it cancels from the comparison between candidates, and for every pair j, j' the error of the*

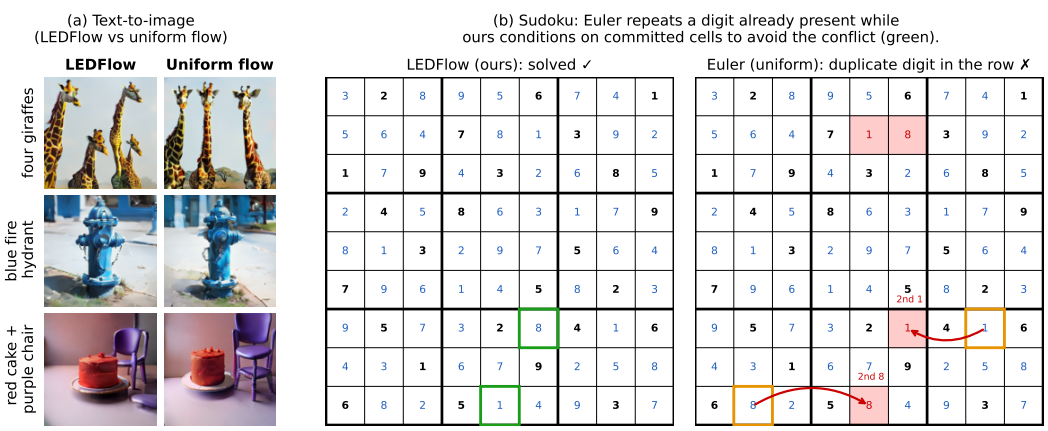


Figure 3: **Qualitative comparison across the structure spectrum. (a)** Text-to-image on FUDOKI shows LEDFLOW produces images with improved alignment with the required descriptions. **(b)** Sudoku: unordered Euler sampler repeats digits already present (two '1's and '8's in row 7 and 9), violating the row constraint, while LEDFLOW conditions on absorbed cells and avoids the conflict.

decision-relevant score difference obeys

$$\left|\big(G_j^\theta(s) - G_{j'}^\theta(s)\big) - \big(G_j^q(s) - G_{j'}^q(s)\big)\right| \le 2\eta_m, \qquad \eta_m := (m-1)\big(\delta + \epsilon \log|\mathcal{S}|\big),$$
$$G_{j^\star}^q(s) - G_{\hat{j}}^q(s) \le 2\eta_m, \qquad h_{\hat{j}_{\text{ent}}}^q(s) - \min_j h_j^q(s) \le 2\delta.$$

While the bounds do not imply a direct performance ordering, they clarify how estimation error affects the two objectives. The worst-case lookahead bound scales with window size m through accumulated error over the remaining $m-1$ positions, and preserves the oracle ranking whenever the candidate margin exceeds $2\eta_m$. App. C.3 details derivation and analysis outside oracle support.

4 EXPERIMENTS

Setup. We evaluate LEDFLOW across tasks with increasing structural dependence: six multimodal-understanding benchmarks via VLMEvalKit [8], GenEval text-to-image generation [11], and reasoning on MathVista, MathVerse, GSM8K, three Sudoku sets, Latin-square, graph coloring, and molecular infilling [19, 49, 6, 28, 38, 26]. We use FUDOKI [40] with matched configurations for understanding, generation, and mathematics, and task-specific uniform-flow denoisers for puzzles, evaluating LEDFLOW in a training-free setting. All tasks use $K = 64$ sampling steps. Evaluation uses full standard splits, 553 GenEval prompts with four images each, 500/100/700 generated/Nikoli/Extreme Sudoku instances, and five seeds. We compare two baseline groups. The first contains non-absorbing decoders: Euler [40], Time- and Location-corrected samplers [36], higher-order solvers (RK, RK2-Trapezoid) [37], exact guidance [35], and FreeCorrection [23]. The second contains uniform flow with different absorption policies: probability margin (Top K-margin) and global lookahead (Info-Gain) [44]. App. D provides experimental details.

4.1 EMPIRICAL PERFORMANCE

Multimodal understanding. Adding entropy-guided selective absorption gives consistently higher but modest gains over the replicated Euler baseline across all six benchmarks (Table 1), although Info-Gain leads on POPE and exact guidance on GQA. These modest gains place understanding at the weakly constrained end of the task spectrum, where absorption order matters less. Further analyses, full results, and the uncertainty of these gains are provided in App. E.1.

Text-to-image generation. Table 2 shows that LEDFLOW attains the highest overall score (0.7814) and the best average rank (2.67), ahead of decode-time exact guidance (0.7800). The resulting spread across absorption policies is wider than in multimodal understanding, consistent with text-to-image prompts imposing more structure than the understanding benchmarks above. Fig. 3 provides qualitative examples. We provide further generation experiments and comparisons in App. E.2.

Table 1: **Multimodal understanding scores.** White/grey/blue rows denote non-absorbing baselines/absorbing baselines/LEDFLOW. **Bold**, *italic*, and underlining mark first, second, and third place. Avg. rank is the mean benchmark rank (↓). Higher scores are better (↑). Entropy-guided selective absorption gives consistently higher but modest gains.

sampler	POPE↑	MME↑	MMBench↑	GQA↑	MMMU↑	MM-Vet↑	Avg. rank↓
Euler	86.1	1485.4	73.9	57.6	34.3	38.0	4.33
Euler (our replication)	87.4	1494.5	73.7	57.0	*36.2*	38.2	3.50
Exact guidance	86.8	1492.7	*74.2*	**58.2**	35.4	38.6	*3.00*
Top K-margin	83.5	1497.4	65.0	56.9	30.0	*40.0*	4.50
Info-Gain	**89.3**	*1503.0*	65.1	56.4	29.1	37.5	4.33
LEDFLOW (ours)	*89.0*	**1504.6**	**74.6**	*58.0*	**37.4**	**40.5**	**1.33**

Table 2: **Text-to-image generation scores.** Wider spread from LEDFLOW on absorption policies than Table 1 is consistent with text-to-image prompts imposing more structure than text generation.

Sampler	Single↑	Two↑	Count↑	Colors↑	Pos.↑	Attr.↑	Overall↑	Avg. rank↓
Euler	*0.9625*	0.8384	0.4875	0.8833	**0.7200**	0.6300	0.7536	5.42
Euler (our replication)	0.9375	0.8636	0.5031	0.8723	0.6800	0.6625	0.7532	5.92
Time-corrected	0.9125	*0.8687*	0.5125	**0.9149**	0.6600	0.7000	0.7614	4.83
Location-corrected	**0.9750**	*0.8687*	0.5000	*0.9043*	0.6700	0.6400	0.7597	4.08
RK	0.9250	0.8384	**0.5625**	0.8723	0.6500	*0.7400*	0.7647	6.17
RK2-Trapezoid	0.9500	0.7980	0.4750	*0.9043*	0.6700	0.6800	0.7462	5.67
Exact guidance	0.9400	0.8600	*0.5300*	0.8900	0.7000	**0.7700**	*0.7800*	*3.67*
Top K-margin	0.9469	0.7854	0.4188	0.8856	0.6275	0.5800	0.7071	8.17
Info-Gain	0.9312	0.7753	0.4906	0.8777	0.6600	0.5550	0.7150	8.42
LEDFLOW (ours)	0.9563	**0.8712**	0.5219	0.9016	*0.7150*	0.7225	**0.7814**	**2.67**

Table 3: **Constrained reasoning accuracy.** LEDFLOW outperforms in constrained reasoning task.

Sampler	Mathematics↑			Sudoku↑			Other puzzles↑			Avg. rank↓
	MathVista	MathVerse	GSM8K	Generated	Nikoli	Extreme	Latin	Graph	Molec.	
Euler	0.254	0.110	0.026	0.610	0.410	0.078	0.652	0.936	0.828	5.22
FreeCorrection	**0.261**	0.119	0.019	0.630	0.400	0.126	0.628	*0.988*	0.797	4.72
Time-corrected	0.241	0.111	0.022	0.610	0.410	0.078	0.652	0.936	0.828	5.56
Location-corrected	0.246	*0.120*	**0.029**	0.588	0.360	0.096	0.628	0.938	0.830	4.78
Top K-margin	*0.255*	*0.120*	0.024	*0.820*	*0.796*	**0.271**	0.948	0.951	0.887	*2.67*
Info-Gain	0.227	0.119	0.019	0.804	0.780	0.231	*0.990*	**1.000**	**0.914**	3.50
LEDFLOW (ours)	*0.255*	**0.123**	*0.027*	**0.865**	**0.845**	*0.269*	**0.998**	**1.000**	*0.906*	**1.56**

Reasoning. Table 3 reports constrained reasoning results. LEDFLOW reaches 0.845 on Nikoli against 0.796 for margin and 0.780 for lookahead ordering, and also leads these policies on the generated set while staying competitive in the Extreme set. Across the nine tasks, LEDFLOW is numerically higher than Info-Gain on seven and remains competitive on molecular infilling (0.906 against 0.914). Under the standard protocol ($K < M$ and $b_k > 1$), LEDFLOW improves empirically, while Thm. 1 provides a local conditional-error guarantee. App. E.5 reports detailed comparisons.

4.2 EMPIRICAL ANALYSIS OF ABSORPTION AND LOOKAHEAD

Entropy-guided absorption (Thm. 1). Fig. 4(a) examines the ordering policy in Thm. 1 pooled over Sudoku puzzles in Table 3. Accuracy decreases monotonically from LEDFLOW (0.461) to arbitrary (0.225), the non-absorbing Euler flow (0.211), and a controlled absorption of high-entropy tokens (0.115). Panel (b) shows that, relative to the default absorption order of four models' samplers, LEDFLOW provides generalizable improvement, with largest gains on constrained reasoning benchmarks. Grey cells mark inapplicable settings. Detailed results with statistical tests are provided in App. E.6.

Error amplification in global entropy reduction (Lemma 1). Fig. 4(c) holds the configuration of LEDFLOW fixed and varies only the lookahead window aggregated by the Info-Gain policy [44]. We observe score sensitivity consistent with the analysis in Lemma 1: the mean absolute score error grows from 0.17 with a single window to 1.76 under full lookahead, while accuracy peaks at eight aggregated windows and then falls to 0.31, consistent with accumulated estimation error affecting position selection under an imperfect denoiser. Panel (d) shows that across the eleven benchmarks, LEDFLOW has an inference time similar to that of the uniform flow sampler, while Info-Gain costs about eight times as much. App. E.7 provides a detailed analysis of efficiency and block size.

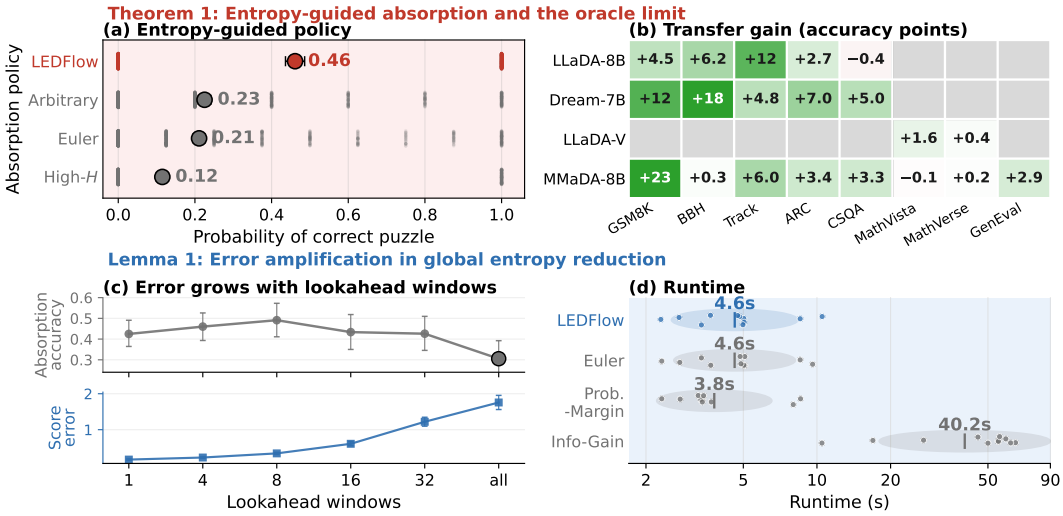


Figure 4: **Empirical analysis of entropy-guided absorption and the score sensitivity of global lookahead.** *Red (Thm. 1):* LEDFLOW's entropy-guided absorption policy gives the highest puzzle accuracy (a), and its gain over each model's default generation order transfers across pretrained models and tasks (b). *Blue (Lemma 1):* aggregating more lookahead terms increases score error and eventually reduces absorption accuracy (c), with increased inference runtime (d).

5 Related Work

Discrete flow sampling, correction, and absorption. Uniform discrete flows iteratively sample each position for repeated revisions [10, 40], using Euler or τ-leaping sampler on a time grid [4, 10, 29]. Corrected samplers reduce errors caused by approximating the CTMC within a sampling interval, while uniformization removes this discretization error by simulating the process exactly [36, 37, 50]. Separately, while absorbing states are standard in masked diffusion [1, 14, 30, 18], uniform diffusion admits conditional-absorbing representations [12, 33], motivating work on persistent context [13, 7]. Building on these, we augment a native uniform discrete flow with selective absorption and analyze its local KL error without changing the uniform flow velocity on positions that remain active.

Reveal order in masked generation. Masked generation has long used confidence to select predictions for irreversible decoding [5, 16, 46], while planning and remasking methods learn which tokens to reveal or revisit [25, 15, 43, 39]. Prioritizing confidence alone can postpone high entropy reasoning forks and reduce solution coverage [21, 41]. Info-Gain estimates lookahead quality over the remaining positions [44], which depends on denoiser quality. EB-Sampler [2] controls parallel unmasking block error with an entropy bound, confidence thresholds decode positions in parallel [42], while BoE steering [27] guides decoding with gradients of successor state entropy. LEDFLOW applies entropy-guided absorption to a revisable uniform flow with analysis of local prediction and dependence costs. App. F provides broader scope of related work.

6 Conclusion

We introduce generation order into uniform discrete flow through selective absorption, which fixes chosen predictions while allowing other positions to continue evolving. Our training-free sampler, LEDFLOW, absorbs the lowest-entropy positions first to protect reliable predictions from later errors. We decompose local absorption error into joint dependence and conditional prediction terms, and show that under entropy-error regularity this policy minimizes an upper bound on the conditional term for a fixed absorption count. We also show how a worst-case decision-error bound for global lookahead grows as it combines estimates across more positions. Experiments on reasoning, text-to-image generation, and multimodal understanding show the largest gains on strongly constrained tasks, while trajectory diagnostics show fewer correct-to-wrong revisions. Our analysis characterizes a local error bound under entropy-error regularity, with joint position selection, correction of absorbed predictions, and lookahead with more accurate denoisers offering directions for further study (App. A).

ETHICS STATEMENT

This work introduces an inference-time sampler and involves no human subjects, no user study, and no collection of new data. All experiments use publicly released pretrained models (FUDOKI, LLaDA, LLaDA-V, MMaDA, Dream) and public benchmarks, used within their stated licenses and intended research use. The task-specific Sudoku, Latin-square, graph-coloring, and molecular-infilling denoisers are trained from scratch on programmatically generated or public puzzle sets that contain no personal or identifying information. LEDFLOW changes the decoding order of an existing model and adds no safety filtering, so a model decoded with LEDFLOW inherits the biases, factual errors, and misuse potential of the underlying checkpoint, and the improvements we report on text-to-image and multimodal generation carry the dual-use risks common to generative models. The molecular-infilling results are benchmark measurements on MOSES and are not a validation of any candidate for synthesis or downstream chemical use. We report the settings in which LEDFLOW gives little or no benefit, including weakly constrained multimodal understanding and the Sudoku-Extreme set, in Sec. 4 and App. A.

REPRODUCIBILITY STATEMENT

The sampler is specified completely in the main text: the absorbing-state augmentation in Eq. 2, the entropy-guided selection rule in Eq. 3 and Eq. 4, and the full procedure in Algorithm 1. LEDFLOW is training free and requires no training or fine-tuning of the denoiser: on a released backbone such as FUDOKI the indicator only gates the velocity, while the puzzle denoisers receive the indicator as an input feature during their single pretraining run, shared by every sampler (App. C.1). Any uniform discrete flow checkpoint can therefore reproduce the sampler from these equations alone. The theoretical claims state their assumptions explicitly in Def. 1 and in the reachability conditions of Sec. 3.3. App. C.3 gives the proofs of Proposition 2, Corollary 1, Thm. 1, and Lemma 1, together with the treatment of lookahead states outside the oracle support. App. D documents the experimental settings: checkpoints and capacities, the absorption schedule and count, the sampler step budget $K = 64$, the denoiser training configuration for the puzzle tasks, and the evaluation split and sample count for every benchmark. App. D defines the two Sudoku protocols used in this paper and states which tables and figures belong to each, so that measurements from different protocols are not compared directly. App. C.2 gives the schedule, update order, and matched-policy details needed to reproduce the baseline absorption policies under identical conditions. Reported comparisons state their uncertainty: App. E.1 gives interval estimates and their approximation limits, and App. E.5 reports puzzle-level bootstrap confidence intervals and paired McNemar tests at n=1000. The supplementary material accompanying this submission releases the code and environment for the self-contained Sudoku and constrained-reasoning experiments: the task-specific denoiser and its training, the reveal-order samplers (LEDFLOW, margin, and Info-Gain), and the trajectory, calibration, and efficiency diagnostics, which reproduce the results in Table 3, Fig. 2, and Fig. 4(a) without any external checkpoint. These experiments require no external base model, so LEDFLOW itself is fully specified by the equations above and needs no released code to reproduce. The complete code for the text-to-image, multimodal-understanding, and 8B-transfer experiments, which build on third-party frameworks and their released checkpoints, will be released with the camera-ready version.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, specifically to write and execute the experimental code, polish the writing, structure the code repository, and modify the scientific figures. We have not used generative AI tools to develop theoretical models or conceptual frameworks, to design research methodology or experiments, to interpret results, to generate synthetic data, or to perform qualitative data analysis. We have reviewed all AI-assisted work: the authors checked the experimental code against the method descriptions, verified that the reported numbers are reproduced from the logged runs, inspected every figure against its underlying data, and read all AI-edited text to confirm that it preserves the intended technical meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

[1] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393. URL https://dl.acm.org/doi/10.5555/3540261.3541637.

[2] Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=WBcBhT1NKO.

[3] Changxiao Cai and Gen Li. Confidence-based decoding is provably efficient for diffusion language models, 2026. URL https://arxiv.org/abs/2603.22248.

[4] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: enabling multimodal flows with applications to protein co-design. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024. URL https://dl.acm.org/doi/10.5555/3692070.3692283.

[5] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11305–11315, 2022. doi: 10.1109/CVPR52688.2022.01103. URL https://ieeexplore.ieee.org/document/9878676.

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

[7] Justin Deschenaux, Caglar Gulcehre, and Subham Sekhar Sahoo. The diffusion duality, chapter II: ψ-samplers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RSIoYWIzaP.

[8] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, Dahua Lin, and Kai Chen. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, pp. 11198–11201, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706868. doi: 10.1145/3664647.3685520. URL https://doi.org/10.1145/3664647.3685520.

[9] Alberto Foresti, Mustapha Bounoua, Giulio Franzese, Luca Ambrogioni, and Pietro Michiardi. Improved sampling schedules for discrete diffusion models, 2026. URL https://arxiv.org/abs/2602.06849.

[10] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385. URL https://dl.acm.org/doi/10.5555/3737916.3742155.

[11] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=Wbr51vK331.

[12] Samson Gourevitch, Yazid Janati, Dario Shariatian, Umut Simsekli, Eric Moulines, Eric P. Xing, and Alain Durmus. Uniform diffusion models revisited: Leave-one-out denoiser and absorbing state reformulation, 2026. URL https://arxiv.org/abs/2605.22765.

[13] Satoshi Hayakawa. From truncation to commitment: Persistent context in uniform discrete diffusion, 2026. URL https://arxiv.org/abs/2609.01043.

[14] Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. DiffusionBERT: Improving generative masked language models with diffusion models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4521–4534, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.248. URL https://aclanthology.org/2023.acl-long.248/.

[15] Chunsan Hong, Seonho An, Min-Soo Kim, and Jong Chul Ye. Improving discrete diffusion unmasking policies beyond explicit reference policies. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=on6cb46OhD.

[16] Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham 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'25. JMLR.org, 2025. URL https://dl.acm.org/doi/10.5555/3780338.3781545.

[17] Tung Sum Thomas Kwok, Zeyong Zhang, Xinyu Wang, Chunhe Wang, Xiaofeng Lin, Hanwei Wu, Lei Ding, Guang Cheng, and Zhijiang Guo. From table to cell: Attention for better reasoning with tabalign, 2026. URL https://arxiv.org/abs/2605.14465.

[18] 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 '24, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

[19] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KUNzEQMWU7.

[20] Yatin Nandwani, Rishabh Ranjan, Mausam ., and Parag Singla. A solver-free framework for scalable learning in neural ILP architectures. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=EqZuN4V_FLF.

[21] Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, and Gao Huang. The flexibility trap: Rethinking the value of arbitrary order in diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=kpgURPRMGf.

[22] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KnqiC0znVF.

[23] Yidong Ouyang, Panwen Hu, Zhengyan Wan, Zhe Wang, Liyan Xie, Dmitriy Bespalov, Ying Nian Wu, Guang Cheng, Hongyuan Zha, and Qiang Sun. Training-free self-correction for multimodal masked diffusion models, 2026. URL https://arxiv.org/abs/2602.02927.

[24] Rasmus Berg Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 3372–3382, Red Hook, NY, USA, 2018. Curran Associates Inc. URL https://dl.acm.org/doi/10.5555/3327144.3327256.

[25] Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion models with applications to biological sequence generation. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025. URL https://openreview.net/forum?id=fFuVPKpSt0.

[26] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alan Aspuru-Guzik, and Alex Zhavoronkov. Molecular sets (moses): A benchmarking platform for molecular generation models, 2020. URL https://arxiv.org/abs/1811.12823.

[27] Shreshth Saini, Avinab Saha, Balu Adsumilli, Neil Birkbeck, Yilin Wang, and Alan C. Bovik. Tabes: Trajectory-aware backward-on-entropy steering for masked diffusion models, 2026. URL https://arxiv.org/abs/2602.00250.

[28] Jeffrey Seely, Yuki Imajuku, Tianyu Zhao, Edoardo Cetin, and Llion Jones. Sudoku-bench: Evaluating creative reasoning with sudoku variants, 2026. URL https://openreview.net/forum?id=GNRuPhVrNP.

[29] 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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tcvMzR2NrP.

[30] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385. URL https://dl.acm.org/doi/10.5555/3737916.3741193.

[31] Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. Scaling laws with vocabulary: Larger models deserve larger vocabularies. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=sKCKPr8cRL.

[32] Gennesaret Tjusila, Mathieu Besançon, Mark Turner, and Thorsten Koch. How many clues to give? a bilevel formulation for the minimum sudoku clue problem. *Oper. Res. Lett.*, 54(C), May 2024. ISSN 0167-6377. doi: 10.1016/j.orl.2024.107105. URL https://doi.org/10.1016/j.orl.2024.107105.

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

[34] Zhengyan Wan, Yidong Ouyang, Panwen Hu, and Qiang Sun. dflowgrpo: Rate-aware policy optimization for discrete flow models, 2026. URL https://arxiv.org/abs/2605.09291.

[35] Zhengyan Wan, Yidong Ouyang, Liyan Xie, Fang Fang, Hongyuan Zha, and Guang Cheng. Discrete guidance matching: Exact guidance for discrete flow matching. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=N1RYhOg6ib.

[36] Zhengyan Wan, Yidong Ouyang, Liyan Xie, Hongyuan Zha, Fang Fang, and Guang Cheng. Corrected samplers for discrete flow models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=h8g5eiSEpc.

[37] Zhengyan Wan, Yidong Ouyang, Qiang Yao, Liyan Xie, Fang Fang, Hongyuan Zha, and Guang Cheng. Error analysis of discrete flow with generator matching. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tiPoSBaOiY.

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

[39] Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling, 2026. URL https://arxiv.org/abs/2503.00307.

[40] Jin Wang, Yao Lai, Aoxue Li, Shifeng Zhang, Jiacheng Sun, Ning Kang, Chengyue Wu, Zhenguo Li, and Ping Luo. FUDOKI: Discrete flow-based unified understanding and generation via kinetic-optimal velocities. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=RSVdHXZN6D.

[41] Russ Webb, Amitis Shidani, Alice Bizeul, and Dan Busbridge. Limits of confidence in diffusion, 2026. URL https://arxiv.org/abs/2609.20581.

[42] Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding, 2025. URL https://arxiv.org/abs/2505.22618.

[43] Jiawei Xu, Minghui Liu, Aakriti Agrawal, Yifan Chen, and Furong Huang. Scheduling thoughts: Learning the order of thought in diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=j4UhHaxCpq.

[44] Kaisen Yang, Jayden Teoh, Kaicheng Yang, Yitong Zhang, and Alex Lamb. Improving sampling for masked diffusion models via information gain. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=1HPTFpwktA.

[45] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal large diffusion language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=wczmXLuLGd.

[46] 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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=NRYgUzSPZz.

[47] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models, 2025. URL https://arxiv.org/abs/2508.15487.

[48] Zebin You, Shen Nie, Xiaolu Zhang, JUN ZHOU, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2026. URL https://cvpr.thecvf.com/virtual/2026/poster/39049.

[49] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng Li. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VIII*, pp. 169–186, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-73241-6. doi: 10.1007/978-3-031-73242-3_10. URL https://doi.org/10.1007/978-3-031-73242-3_10.

[50] Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=HDeIb67lJe.

[51] 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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=CTC7CmirNr.

A CLAIM SCOPE AND LIMITATIONS

Table 4 summarizes the evidence and boundary associated with each main claim. The core paper studies an entropy-guided selective-absorption policy added at inference time to a uniform discrete flow.

Table 4: Evidence and scope of the main claims.

Claim	Supported by	Boundary
Selective absorption separates the priority decision from the native flow update	Section 3.1 and Appendix C.1	Absorption deliberately changes the sampler law; it is not claimed to preserve the path law of the unmodified uniform flow.
The local absorption error separates into joint factorization and conditional prediction terms	Proposition 2 and Appendix C.3.1	The equality is local to a fixed state and selected block; errors across the complete sampling trajectory also depend on subsequent states.
Lowest predictive entropy minimizes the conditional-error upper bound	Theorem 1 and Appendix C.3	The result assumes entropy-error regularity on oracle-reachable states; it optimizes a local surrogate and does not cover trajectories after support is lost.
Global entropy reduction accumulates estimation error with lookahead window	Lemma 1 and Appendix E.7	This is an upper-bound and controlled fixed-state result; it does not imply that lookahead fails for every denoiser or state.
Entropy-guided absorption transfers across tasks and pretrained models	Section 4.2 and Appendix E.6	Transfer replaces each model's default generation order through its own sampler; several differences are not significant. Sudoku runs share a checkpoint within each capacity; all rows of Table 3 share one matched cohort, while separate diagnostics retain their stated protocols.

A.1 LIMITATIONS AND INTENDED SCOPE

Local theoretical result. Theorem 1 bounds the error of the current absorption decision under entropy-error regularity. Absorbing a position changes the state encountered later, so the theorem does not assert that a locally optimal block minimizes the final sequence loss for every model or task. The oracle limit concerns conditional KL at a fixed oracle-reachable state. It does not guarantee recovery from earlier incorrect absorptions or eliminate joint dependence and oracle classification risk.

Irreversible absorption. Once selected, a position cannot be revised. This makes the contribution of the absorption policy measurable and preserves the native uniform-flow update on active positions, but it can propagate an early incorrect prediction. The method is consequently best suited to settings in which predictive entropy is informative. Combining selective absorption with principled revision is an open direction.

Lookahead comparison. Lemma 1 identifies how errors in counterfactual predictions accumulate when more entropy terms are aggregated. The bound does not state that Info-Gain must underperform: lookahead can remain useful when its estimates are accurate or its action margins exceed their error. Our fixed-state window experiment tests when the accumulated error becomes large enough to change the selected position.

Baseline coverage. Our decode-time baselines are those with a published implementation or a fully specified algorithm that runs on a uniform discrete flow without modification. Two related samplers fall outside that set. EB-Sampler [2] controls the block size rather than the ordering, so it is complementary to LEDFLOW rather than an alternative to it. We report a reimplementation

on the puzzle tasks in Appendix E.7.3, labelled as such, rather than a row in the cross-task tables. BoE steering [27] requires a mask embedding to anchor its first-order expansion, which a uniform flow does not provide. We therefore do not report a number for it rather than substitute a surrogate of our own construction. Any future comparison against either method should be labelled as a reimplementation, compute matched by function evaluations, and reported at the best setting found in a disclosed hyperparameter sweep.

Empirical coverage. The experiments span multimodal understanding, generation, and structured reasoning, but the largest improvements occur when outputs contain strong cross-position constraints. Broader nonuniform flow backbones and tasks permitting revision would further test the scope of selective absorption.

Training and decoding regimes. LEDFLOW is a training-free sampler in both settings: no sampler, including LEDFLOW, trains or tunes any model. The absorption indicator is maintained by the sampler and gates the velocity in Eq. 2. On FUDOKI the released weights remain frozen and the indicator is not a denoiser input, so the model sees only the current token configuration. For the puzzle tasks, the task-specific denoiser is pretrained once with the standard cross-entropy objective and additionally receives the indicator as an input feature, so fixed positions are visible to it; this single checkpoint is shared by all absorbing and non-absorbing samplers (App. C.1). That pretraining does not guarantee calibration after model-generated absorptions. Both non-absorbing and absorbing Sudoku baselines use coordinatewise kinetic-optimal updates, with a cosine absorption schedule for absorbing baselines. The terminal-rate proposition concerns the flow velocity, whereas the local absorption analysis concerns the selected absorption block. The ordering diagnostic and the singleton-absorption studies are separate ablations.

Remaining empirical attribution. The current experiments do not isolate selective absorption against terminal-rate clipping, early stopping, or suppressed target resampling. The per-step profile in Appendix E.7 attributes 3.6% of a step to the ordering score, so reported timing differences among confidence scores reflect implementation rather than the score itself. That profile is relative and measured on CPU. The block-size ablation varies the number of positions absorbed per step and does not cover every absorption schedule.

B Details for Section 2: Problem Formulation

This appendix expands the two subsections of Section 2: Appendix B.1 gives the discrete-flow background behind Section 2.1, and Appendix B.2 proves the terminal-exposure proposition and documents the Sudoku diagnostic of Section 2.2.

B.1 Preliminaries: discrete flow matching background

Definition 2 (Conditional CTMC)**.** Given a rate matrix $R_t(y,z)_{y,z\in\mathcal{S}^M}$ where $R_t(y,z)\geq 0$ for $z\neq y$ and $\sum_z R_t(y,z)=0$, the process $\{a_t\}$ is a CTMC with rate R_t, if for $h>0$, $\mathbb{P}(a_{t+h}=z|a_t=y)=\delta_y(z)+R_t(y,z)h+o(h)$ and its marginal density p_t at time t obeys the Kolmogorov forward equation $\dot{p}_t(y)=\sum_{z\neq y}(p_t(z)R_t(z,y)-p_t(y)R_t(y,z))$.

We expand the probability-path and velocity formulation summarized in Section 2.1, following Gat et al. [10], and then give the uniform metric-induced specialization used by FUDOKI [40].

Conditional and marginal probability paths. Let $\pi(a_0,a_1\mid\mathcal{C})$ be a coupling whose marginals are the source $p_0(a_0\mid\mathcal{C})$ and target $q(a_1\mid\mathcal{C})$. Discrete flow matching specifies a conditional path between each coupled pair and marginalizes the pair:

$$p_t(a\mid\mathcal{C})=\sum_{a_0,a_1} p_t(a\mid a_0,a_1,\mathcal{C})\,\pi(a_0,a_1\mid\mathcal{C}), \tag{9}$$

$$p_t(a\mid a_0,a_1,\mathcal{C})=\prod_{j=1}^{M} p_t^j(a^j\mid a_0,a_1,\mathcal{C}). \tag{10}$$

The boundary conditions $p_0^j(\cdot \mid a_0, a_1) = \delta_{a_0^j}$ and $p_1^j(\cdot \mid a_0, a_1) = \delta_{a_1^j}$ ensure that the marginal path begins at p_0 and ends at q. When the source is fixed or integrated into the coordinate path, this reduces to the target-conditioned notation $p_{t|1}(a \mid a_1)$ used in the main text.

Probability velocities and continuity. For a current state a, the coordinate velocity $u_t^j(v, a)$ defines the infinitesimal update

$$\mathbb{P}(A_{t+h}^j = v \mid A_t = a) = \delta_{a^j}(v) + h\, u_t^j(v, a) + o(h). \tag{11}$$

It is a valid CTMC rate when

$$\sum_{v \in \mathcal{S}} u_t^j(v, a) = 0, \qquad u_t^j(v, a) \geq 0 \quad \text{for } v \neq a^j. \tag{12}$$

The full velocity permits one-coordinate transitions, $u_t(z, a) = \sum_j \delta(z^{-j}, a^{-j}) u_t^j(z^j, a)$, with the first argument denoting the destination. Hence the path evolves according to the Kolmogorov forward equation

$$\dot{p}_t(a \mid \mathcal{C}) = \sum_{z \in \mathcal{S}^M} p_t(z \mid \mathcal{C}) u_t(a, z). \tag{13}$$

Equivalently, define the probability flux from z to a by $J_t(a, z) = p_t(z) u_t(a, z)$. Then $\dot{p}_t(a) + \mathrm{div}_a\, J_t = 0$, where

$$\mathrm{div}_a\, J_t = \sum_{z \neq a} \big[J_t(z, a) - J_t(a, z) \big]. \tag{14}$$

Thus a velocity generates p_t precisely when its flux satisfies this continuity equation and the rate conditions above.

Conditional velocities and posterior marginalization. Suppose $u_t^j(v, a^j \mid a_0, a_1, \mathcal{C})$ generates the coordinate path in (10). The corresponding marginal velocity is obtained by averaging over the target pair conditioned on the current state:

$$u_t^j(v, a, \mathcal{C}) = \mathbb{E}_{(a_0, a_1) \sim p_t(\cdot, \cdot \mid a, \mathcal{C})} \Big[u_t^j(v, a^j \mid a_0, a_1, \mathcal{C}) \Big], \tag{15}$$

$$p_t(a_0, a_1 \mid a, \mathcal{C}) = \frac{p_t(a \mid a_0, a_1, \mathcal{C}) \pi(a_0, a_1 \mid \mathcal{C})}{p_t(a \mid \mathcal{C})}. \tag{16}$$

For a target-conditioned path, only $p_{1|t}(a_1 \mid a, \mathcal{C})$ remains unknown. It is approximated by the denoiser $\mathbb{Q}_{1|t}^\theta$, trained with the coordinate cross-entropy objective

$$\mathcal{L}_{\mathrm{den}}(\theta) = -\mathbb{E}_{t, a_0, a_1, a_t} \left[\sum_{j=1}^M \log \mathbb{Q}_{1|t}^\theta(a_1^j \mid a_t, \mathcal{C}_t) \right]. \tag{17}$$

Substituting this posterior into (15) yields the learned velocity used at generation time.

Uniform metric-induced flow in FUDOKI. FUDOKI uses a distance $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ satisfying $d(v, y) = 0$ if and only if $v = y$, and defines

$$p_{t|1}^j(v \mid a_1^j) = \frac{\exp[-\beta_t d(v, a_1^j)]}{\sum_{w \in \mathcal{S}} \exp[-\beta_t d(w, a_1^j)]}, \qquad \beta_0 = 0, \quad \beta_t \xrightarrow[t \to 1]{} \infty. \tag{18}$$

The initial coordinate distribution is therefore uniform over $\mathcal{S}$, while the target concentrates on a_1^j. To select a velocity for this prescribed path, write the conditional flux from r to v as $J_t^j(v, r \mid a_1^j)$ and minimize its kinetic energy subject to continuity, nonnegative off-diagonal flux, and the path boundaries:

$$\min_{J_t^j} \int_0^1 \sum_{v \neq r} \omega_t(v, r) \frac{J_t^j(v, r \mid a_1^j)^2}{p_{t|1}^j(r \mid a_1^j)}\, \mathrm{d}t, \qquad \omega_t(v, r) > 0. \tag{19}$$

For the weighting $\omega_t(v, r) = 1/p_{t|1}^j(v \mid a_1^j)$, the optimal flux is

$$J_t^{j,\star}(v, r \mid a_1^j) = \Big[p_{t|1}^j(r \mid a_1^j)\, \dot{p}_{t|1}^j(v \mid a_1^j) - \dot{p}_{t|1}^j(r \mid a_1^j)\, p_{t|1}^j(v \mid a_1^j) \Big]_+. \tag{20}$$

Dividing by the probability at the current token $r = a^j$ gives the off-diagonal conditional velocity

$$u_t^j(v, a^j \mid a_1^j) = p_{t|1}^j(v \mid a_1^j)\dot{\beta}_t \left[d(a^j, a_1^j) - d(v, a_1^j)\right]_+, \qquad v \neq a^j, \tag{21}$$

with the diagonal fixed by (12). Probability mass therefore moves only to tokens closer to the target under d. Finally, FUDOKI replaces the target posterior in (15) by $\mathbb{Q}_{1|t}^\theta$:

$$u_t^{\theta,j}(v, a, \mathcal{C}_t) = \mathbb{E}_{a_1 \sim \mathbb{Q}_{1|t}^\theta(\cdot|a,\mathcal{C}_t)}\left[u_t^j(v, a^j \mid a_1^j)\right]. \tag{22}$$

Applying (11) with this velocity gives the native FUDOKI sampler. Because the uniform path has full vocabulary support and introduces no terminal state, a coordinate may change repeatedly before $t = 1$. The selective absorbing state in Section 3.1 is an additional inference mechanism rather than part of this standard flow construction.

B.2 Motivation and ordered absorption: repeated transitions and the absorption decision

This appendix formalizes the transition behavior summarized in Section 2.2 and derives a decision threshold for absorption.

Setting. We work directly with the kinetic-optimal velocity of Eq. 1. Conditioned on the true target a_1^j it reads

$$u_t^j(z^j, a_t \mid a_1^j) = p_t(z^j \mid a_1^j)\,\dot{\beta}_t \left[d(a_t^j, a_1^j) - d(z^j, a_1^j)\right]_+, \qquad p_t(z^j \mid a_1^j) = \frac{e^{-\beta_t d(z^j, a_1^j)}}{Z_t(a_1^j)}, \tag{23}$$

with $Z_t(a_1^j) = \sum_{z \in \mathcal{S}} e^{-\beta_t d(z, a_1^j)}$. At inference the solver substitutes the predicted target $\hat{a}_1^j \sim \mathbb{Q}_{1|t}^\theta(\cdot \mid a_t, \mathcal{C}_t)$ for a_1^j throughout. We assume only that d is a metric on $\mathcal{S}$, so that $d(z, z') > 0$ whenever $z \neq z'$. No further structure is needed, and in particular the argument does not require d to be the discrete metric. This matters because FUDOKI computes d from pretrained token embeddings rather than from token identity. We also use $\beta_t = c(t/(1-t))^\alpha$ with $c = 3, \alpha = 0.9$, whence

$$\dot{\beta}_t = c\,\alpha\, t^{\alpha-1}(1-t)^{-\alpha-1} \xrightarrow[t\to 1]{} \infty. \tag{24}$$

Throughout, a_t^j is the token currently held at position j and a_1^j is the true target there.

Proposition 3 (Target-conditioned monotonicity)**.** *Let*

$$\lambda_t^j := \sum_{z^j \neq a_t^j} u_t^j(z^j, a_t \mid a_1^j), \qquad \hat{\lambda}_t^j := \sum_{z^j \neq a_t^j} u_t^j(z^j, a_t \mid \hat{a}_1^j) \tag{25}$$

be the total transition rates obtained by conditioning the kinetic-optimal velocity on the oracle and predicted targets, respectively. If $a_t^j = a_1^j$, then $\lambda_t^j = 0$. For $\hat{a}_1^j \sim \mathbb{Q}_{1|t}^\theta(\cdot \mid a_t, \mathcal{C}_t)$, the expected rate at which a correct token is replaced satisfies

$$\mathbb{E}[\hat{\lambda}_t^j] \geq \dot{\beta}_t\, \mathbb{E}\Big[\mathbf{1}\{\hat{a}_1^j \neq a_1^j\} p_t(\hat{a}_1^j \mid \hat{a}_1^j) d(a_1^j, \hat{a}_1^j)\Big] \geq \dot{\beta}_t\, \varepsilon_t^j p_{\min,t} d_{\min}, \tag{26}$$

where $\varepsilon_t^j = 1 - \mathbb{Q}_{1|t}^\theta(a_1^j \mid a_t, \mathcal{C}_t)$, $p_{\min,t} = \min_{z \in \mathcal{S}} p_t(z \mid z)$, and $d_{\min} = \min_{z \neq a_1^j} d(z, a_1^j)$. Hence the lower bound is positive whenever all three quantities are positive.

Part (i): the true target leaves a correct token fixed. Suppose the velocity is conditioned on the true target and the position already holds it, so that $a_t^j = a_1^j$. For any $z^j \neq a_t^j$,

$$u_t^j(z^j, a_t \mid a_1^j) = p_t(z^j \mid a_1^j)\,\dot{\beta}_t \left[d(a_1^j, a_1^j) - d(z^j, a_1^j)\right]_+ = p_t(z^j \mid a_1^j)\,\dot{\beta}_t \left[-d(z^j, a_1^j)\right]_+ = 0,$$

because $d(a_1^j, a_1^j) = 0$ and $d(z^j, a_1^j) > 0$, so the bracket is negative and the ReLU annihilates it. Summing over destinations gives $\lambda_t^j = 0$: conditioned on the true target, a position that has reached it never leaves. Retention at the true target is a property of the exact velocity itself, not an addition to it.

Toward part (ii): a single transition under a wrong predicted target. At inference the velocity is conditioned on $\hat{a}_1^j$ rather than a_1^j. Suppose the position is correct, $a_t^j = a_1^j$, but the prediction misses, $\hat{a}_1^j \neq a_1^j$. Taking the destination $z^j = \hat{a}_1^j$,

$$u_t^j(\hat{a}_1^j, a_t \mid \hat{a}_1^j) = p_t(\hat{a}_1^j \mid \hat{a}_1^j)\, \dot{\beta}_t \left[d(a_1^j, \hat{a}_1^j) - d(\hat{a}_1^j, \hat{a}_1^j) \right]_+ = p_t(\hat{a}_1^j \mid \hat{a}_1^j)\, \dot{\beta}_t\, d(a_1^j, \hat{a}_1^j) > 0,$$

since $p_t(\hat{a}_1^j \mid \hat{a}_1^j) = 1/Z_t(\hat{a}_1^j) > 0$ and $d(a_1^j, \hat{a}_1^j) > 0$. The transition is not merely permitted but is the one the velocity most favours, because $\hat{a}_1^j$ is the closest state to the value the velocity is conditioned on. The monotonicity of Eq. 23 is thus relative to $\hat{a}_1^j$ and carries no implication for $d(\cdot, a_1^j)$: a step that moves closer to the prediction can move away from the truth.

Part (ii): the expected total conditional transition rate. Every destination contributes a nonnegative rate, so retaining only the summand $z^j = \hat{a}_1^j$ computed above lower bounds $\hat{\lambda}_t^j$. The term vanishes when $\hat{a}_1^j = a_1^j$ by part (i), so taking expectations over the predicted target gives the first inequality of Eq. 26,

$$\mathbb{E}\left[\hat{\lambda}_t^j\right] \geq \dot{\beta}_t\, \mathbb{E}\left[\mathbf{1}\{\hat{a}_1^j \neq a_1^j\}\, p_t(\hat{a}_1^j \mid \hat{a}_1^j)\, d(a_1^j, \hat{a}_1^j)\right].$$

Bounding $p_t(\hat{a}_1^j \mid \hat{a}_1^j) \geq p_{\min,t} := \min_{z \in \mathcal{S}} 1/Z_t(z) > 0$ and $d(a_1^j, \hat{a}_1^j) \geq d_{\min} := \min_{z \neq a_1^j} d(z, a_1^j) > 0$ on the event $\{\hat{a}_1^j \neq a_1^j\}$, whose probability is $\varepsilon_t^j = 1 - \mathbb{Q}_{1|t}^{\theta}(a_1^j \mid a_t, \mathcal{C}_t)$, yields the second inequality, $\mathbb{E}[\hat{\lambda}_t^j] \geq \dot{\beta}_t\, \varepsilon_t^j\, p_{\min,t}\, d_{\min}$. A denoiser placing all of its mass on the true target has $\varepsilon_t^j = 0$, recovering part (i). Any denoiser with residual target prediction error has $\mathbb{E}[\hat{\lambda}_t^j] > 0$.

Proof of Proposition 1 (terminal blow-up of the transition rate). Fix a position j with $a_t^j \neq a_1^j$. The summand in the total rate of Eq. 25 vanishes at $z^j = a_t^j$ (its bracket is 0), so, factoring the schedule gradient out of Eq. 23,

$$\lambda_t^j = \sum_{z^j \neq a_t^j} u_t^j(z^j, a_t \mid a_1^j) = \dot{\beta}_t \underbrace{\sum_{z^j \in \mathcal{S}} p_t(z^j \mid a_1^j) \left[d(a_t^j, a_1^j) - d(z^j, a_1^j) \right]_+}_{=:\, G_t^j}.$$

Retaining only the destination $z^j = a_1^j$, whose bracket equals $d(a_t^j, a_1^j) - d(a_1^j, a_1^j) = d(a_t^j, a_1^j)$, and discarding the remaining nonnegative terms gives

$$G_t^j \;\geq\; p_t(a_1^j \mid a_1^j)\, d(a_t^j, a_1^j) \;>\; 0,$$

where strict positivity holds because d is a metric and $a_t^j \neq a_1^j$. By Eq. 23,

$$p_t(a_1^j \mid a_1^j) = \frac{1}{Z_t(a_1^j)} = \left(1 + \textstyle\sum_{z \neq a_1^j} e^{-\beta_t d(z, a_1^j)}\right)^{-1} \xrightarrow[t \to 1]{} 1,$$

since β_t grows without bound and $d(z, a_1^j) > 0$ for every $z \neq a_1^j$. Combining with the divergence of $\dot{\beta}_t$ from Eq. 24 gives that $\lambda_t^j = \dot{\beta}_t\, G_t^j$ grows without bound as t approaches 1. For a frozen rate, the jump probability is $1 - e^{-h\lambda_t^j}$, which tends to 1 for fixed $h > 0$ as λ_t^j grows. This is a frozen-rate calculation, not an exact endpoint limit for the time-varying CTMC on $[0, 1]$. Substituting the freshly sampled target $\hat{a}_1^j$ for a_1^j throughout, as the solver does at inference, gives the stated terminal re-derivation. The corresponding expected replacement rate is bounded below in Eq. 26. □

Scope of Proposition 1. The proposition describes transition exposure, which can produce either correction or incorrect revision. It does not establish a net accuracy loss, nor does it cover every uniform-flow path or solver.

Corollary 2 (Continued exposure to transition)**.** *The realized rate vanishes exactly when the predicted target equals the token currently held:*

$$\Pr\left[\hat{\lambda}_t^j = 0 \mid a_t, \mathcal{C}_t\right] = \mathbb{Q}_{1|t}^{\theta}(a_1^j = a_t^j \mid a_t, \mathcal{C}_t). \tag{27}$$

If the position currently holds the correct token, its probability of remaining exposed to a transition at grid point t_k is $\varepsilon_{t_k}^j$, giving $\sum_k \varepsilon_{t_k}^j$ expected exposures over the sampling grid.

Proof of Corollary 2. The computation in part (i) used only that the value conditioning the velocity equals the token held at the position. It did not use that this value was the true target. Applying it at the sampled target gives, for any draw $\hat{a}_1^j = a_t^j$, that $u_t^j(z^j, a_t \mid \hat{a}_1^j) = 0$ for every $z^j \neq a_t^j$ and hence $\hat{\lambda}_t^j = 0$. Conversely, if $\hat{a}_1^j \neq a_t^j$ then the destination $z^j = \hat{a}_1^j$ contributes $p_t(\hat{a}_1^j \mid \hat{a}_1^j)\dot{\beta}_t\, d(a_t^j, \hat{a}_1^j) > 0$, so $\hat{\lambda}_t^j > 0$. Therefore $\{\hat{\lambda}_t^j = 0\} = \{\hat{a}_1^j = a_t^j\}$ exactly, and taking probabilities under the draw $\hat{a}_1^j \sim \mathbb{Q}_{1|t}^\theta(\cdot \mid a_t, \mathcal{C}_t)$ gives Eq. 27. When the position holds the correct token, $a_t^j = a_1^j$, the complement has probability ε_t^j, and summing over the grid gives the stated expected exposure count $\sum_k \varepsilon_{t_k}^j$. Since the source is uniform at $t = 0$, the match probability begins at $1/|\mathcal{S}|$ and rises toward one only as the denoiser concentrates, so the count is close to the full number of sampling steps whenever residual uncertainty persists. □

Fixed points of the marginalized sampler. Because the predicted target is redrawn each step, the object governing whether a position settles is the rate marginalized over that draw. For an incumbent token a_t^j,

$$\mathbb{E}\Big[\hat{\lambda}_t^j\Big] = \sum_{b \in \mathcal{S}} \mathbb{Q}_{1|t}^\theta(a_1^j = b \mid a_t, \mathcal{C}_t) \sum_{z^j \neq a_t^j} p_t(z^j \mid b)\, \dot{\beta}_t \Big[d(a_t^j, b) - d(z^j, b)\Big]_+ .$$

Every summand is nonnegative, and for $t \in (0, 1)$ both $\dot{\beta}_t > 0$ and $p_t(z^j \mid b) > 0$. The expectation therefore vanishes if and only if $[d(a_t^j, b) - d(z^j, b)]_+ = 0$ for every $z^j \neq a_t^j$ and every b in the support of the denoiser, that is, whenever $d(z^j, b) \geq d(a_t^j, b)$ for all $z^j \neq a_t^j$. Choosing $z^j = b$ gives $0 = d(b, b) \geq d(a_t^j, b)$, which forces $b = a_t^j$ since d is a metric. Hence

$$\mathbb{E}\Big[\hat{\lambda}_t^j\Big] = 0 \quad \Longleftrightarrow \quad \mathbb{Q}_{1|t}^\theta(\cdot \mid a_t, \mathcal{C}_t) = \delta_{a_t^j}, \tag{28}$$

so the only fixed points of the marginalized sampler are states at which the denoiser is a point mass on the token already held. This is a statement about certainty, not correctness: a confidently incorrect denoiser produces a fixed point at a wrong token, while an uncertain denoiser produces none, and in neither case is the time at which the sampler settles under the sampler's control. The selective-absorption construction below creates a fixed point independently of the denoiser's confidence, at a time the policy selects.

Absorption sets the rate to zero. Under Eq. 2, $u_t^{\theta,\mathrm{abs},j} = (1 - f_t^j) u_t^{\theta,j}$, so a position absorbed at time τ_j has $f_s^j = 1$ and hence zero outflow for all $s \geq \tau_j$. Therefore $a_s^j = a_{\tau_j}^j$ and $\hat{\lambda}_s^j = 0$ on $[\tau_j, 1]$, and the cumulative exposure $\int_{\tau_j}^1 \mathbb{E}[\hat{\lambda}_s^j]\, \mathrm{d}s$ is truncated at τ_j rather than accumulated to the end of sampling. □

On late-time concentration. Eq. 24 shows that $\dot{\beta}_t$ grows without bound, and $p_t(\hat{a}_1^j \mid \hat{a}_1^j) = 1/Z_t(\hat{a}_1^j)$ tends to 1 as β_t grows, so the rate in part (ii) behaves as $\dot{\beta}_t\, d(a_1^j, \hat{a}_1^j)$ for large t: a late target prediction error acts through an increasingly concentrated transport. We deliberately do not claim that the resulting error probability diverges. The finite-step jump probability is bounded by one, and ε_t^j is expected to decrease as the denoiser sees a more resolved context, so the two factors act in opposite directions. The defensible statement is that the consequence of residual prediction error is governed by the product $\varepsilon_t^j \dot{\beta}_t$ rather than by ε_t^j alone, which makes that product the natural diagnostic to measure.

Measured replacement rate. Figure 5 measures this product on the Sudoku denoiser under the native non-absorbing flow (Nikoli, n=100, seeds 0 to 4). Per grid step we log the incumbent error $\varepsilon_t = 1 - \mathbb{Q}_{1|t}^\theta(a_t^j \mid a_t)$ averaged over generated positions, the per-step hazard c_k of the schedule, and their product, which is the rate at which a currently held token is replaced. The hazard alone rises to one at the endpoint on every grid, but the incumbent error does not vanish: at t=1 it is 0.24, 0.39, 0.51, 0.59, and 0.64 for $K \in \{16, 32, 64, 128, 256\}$, so the terminal replacement rate equals those values. The rate therefore grows as the grid is refined, from 0.24 at K=16 to 0.64 at K=256, which is the opposite of what a discretization artifact would produce: refining the integration brings the sampler closer to the continuous flow and makes the late churn worse, not better. This is the empirical

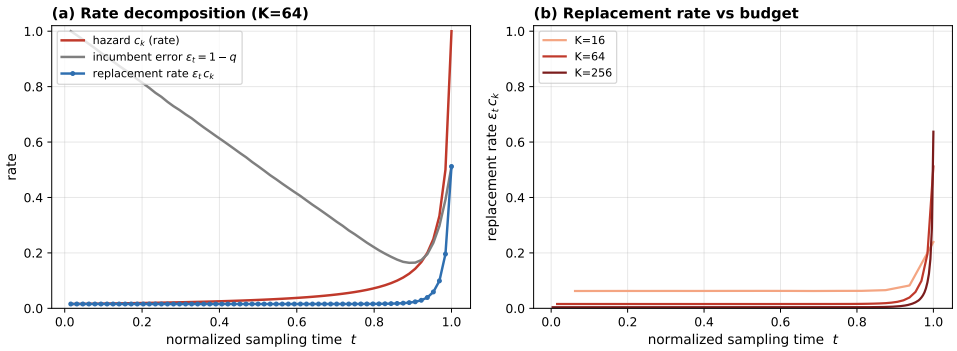


Figure 5: **The replacement rate is the product of residual error and hazard.** (a) At K=64 the per-step hazard c_k (red) rises to one as t approaches 1 while the incumbent error ε_t (grey) falls and then rises again in the last steps. Their product (blue), the correct-token replacement rate, is small for most of sampling and spikes at the endpoint. (b) The product across grids $K \in \{16, 64, 256\}$: the terminal spike grows with K, so refining the discretization increases the late churn.

content of Proposition 1: the schedule gradient sets the opportunity for replacement, and the residual denoiser error decides whether it is taken.

The absorption threshold. Fix a position j and let $\rho = \Pr[a_t^j = a_1^j]$ be the probability that it currently holds the correct token. Over the remaining steps, let r be the probability that an incorrect token is corrected and w the probability that a correct token is replaced, the latter generated by the rate of part (ii). Absorbing at the current state fixes the token, so its expected terminal error is $\mathbb{E}_{\text{abs}} = 1 - \rho$. Leaving the position active gives

$$\mathbb{E}_{\text{active}} = (1 - \rho)(1 - r) + \rho\, w,$$

the first term counting incorrect tokens that are never corrected and the second correct tokens that are replaced. Continued refinement is preferable exactly when $\mathbb{E}_{\text{active}} < \mathbb{E}_{\text{abs}}$, i.e.

$$(1 - \rho)(1 - r) + \rho w < 1 - \rho \iff \rho w < (1 - \rho) r \iff \rho < \frac{r}{r + w}. \tag{29}$$

The threshold is increasing in r and decreasing in w, as expected: a sampler that corrects reliably should refine longer, and one whose replacement rate is high should absorb sooner. Both r and w are properties of the denoiser and the schedule and are not available at decode time, but the comparison depends on the current position only through ρ. This is what makes a reliability estimate the operative quantity.

C Details for Section 3: LEDFlow

This appendix follows the three subsections of Section 3: Appendix C.1 details the absorbing-state augmentation of Section 3.1, Appendix C.2 documents the sampler of Section 3.2, and Appendix C.3 collects the proofs, the calibration diagnostic, and the lookahead analysis behind Section 3.3.

C.1 Selective absorption as an absorbing-state uniform flow

This appendix expands the construction in Section 3.1. The uniform discrete flow has state a_t and learned coordinate velocity $u_t^{\theta,j}$. We augment it with $f_t = (f_t^1, \ldots, f_t^M) \in \{0, 1\}^M$, where $f_t^j = 0$ marks an active position and $f_t^j = 1$ marks a terminal position. Initially, $f_0^j = 0$ for every generated position.

Absorbing velocity. For a fixed augmented state (a_t, f_t), define

$$u_t^{\theta,\text{abs},j}(z^j, a_t, f_t, \mathcal{C}_t) = (1 - f_t^j) u_t^{\theta,j}(z^j, a_t, \mathcal{C}_t).$$

When $f_t^j = 0$, this is exactly the native uniform-flow velocity. When $f_t^j = 1$, every outgoing rate is zero, so the coordinate remains at its absorbed value. Because $1 - f_t^j$ is nonnegative, this modification preserves nonnegative off-diagonal rates and the zero row sum of the native CTMC velocity. The augmented process is therefore a valid CTMC between absorption decisions.

Irreversible update. At grid point t_k, a policy chooses an absorption block B_k from the active set $\mathcal{U}_k = \{j : f_{t_k}^j = 0\}$. For each $j \in B_k$, the update assigns the most likely target value under $\mathbb{Q}_{1|t_k}^{\theta}$ and changes $f_{t_k}^j$ from zero to one. Indicators never return to zero, hence

$$\mathcal{U}_{k+1} \subseteq \mathcal{U}_k, \qquad f_{t_{k+1}}^j \geq f_{t_k}^j.$$

The active positions subsequently follow the original uniform-flow update, while absorbed positions are copied unchanged. The schedule in Section 3.1 fixes the number of active positions remaining after each grid point. The priority policy determines which positions form B_k.

Relation to the exact absorbing representation. Appendix C.1.1 represents uniform transport as a mixture of conditional processes indexed by auxiliary absorbing tokens. That construction is an exact representation of the underlying uniform path. Our indicator f_t^j instead records an inference decision: after a model prediction is selected, Eq. (2) deliberately removes its future native-flow transitions. The selective augmentation therefore borrows the absorbing-state mechanism but does not claim to preserve the path law of the unmodified uniform flow. This is the decision whose local prediction error is analyzed in Section 3.3.

Pretraining the task-specific puzzle denoiser. For the puzzle tasks, we pretrain a task-specific uniform-flow denoiser θ once, with the standard denoising cross-entropy objective and uniform-noise corruption. This checkpoint is fixed before any sampler is evaluated and is shared by every sampler in Table 3. No sampler, including LEDFLOW, is trained or tuned on it. Its input includes a per-position binary status indicator $f_t^j \in \{0, 1\}$, embedded and added to the token embedding as an additional input feature, so the network can tell fixed positions from tentative ones. At each training step we draw a time $t \sim \mathcal{U}(0, 1)$ and construct the corrupted input as follows. Conditioning positions and a fraction t of the generated positions (drawn independently per position, matching the linear schedule $\kappa_t = t$) are marked *absorbed* ($f_t^j = 1$) and hold their ground-truth value. The remaining *active* positions ($f_t^j = 0$) carry a uniform-flow state that interpolates between the target and noise: each keeps its true value with probability t and is otherwise replaced by a uniformly random token, so the active tokens are a partially informative estimate that sharpens as t approaches 1 rather than pure noise or a single mask symbol. The denoiser predicts the clean value at every active position, trained with the denoising cross-entropy $\mathcal{L}_{\mathrm{DLM}}$ restricted to the active positions. Absorbed positions are conditioned on but not scored. At inference the model receives the active/absorbed configuration, fixes selected predictions, and updates the remaining positions. Training absorptions contain ground-truth values, whereas inference absorptions can be wrong and are selected adaptively. The indicator makes absorption status observable but does not eliminate this distribution shift or establish equivalence to the metric-induced path. On multimodal understanding, text-to-image generation, and mathematics we use the released FUDOKI weights: absorption in Eq. 2 is a sampler-side operation that only zeroes the outgoing rate of the selected positions, so the multimodal results of Section 4 are obtained by applying the same entropy-guided ordered absorption at inference to the pretrained FUDOKI uniform-flow model, leaving its weights unchanged. In both settings LEDFLOW is a training-free sampler. The only difference is whether the underlying model is a released backbone or a task-specific puzzle denoiser pretrained once for all samplers.

Checkpoint control. As a control on the model rather than the sampler, we pretrain a second uniform-flow denoiser without the status indicator and evaluate it under the plain uniform-flow sampler with no absorption, on the same 100 Nikoli puzzles at K=64 as Fig. 2. It attains 36.3% cell accuracy and solves $0/100$ puzzles, whereas the main checkpoint under the same non-absorbing sampler solves 0.410 (Fig. 2). The status indicator is therefore informative even when no position is ever absorbed, and the non-absorbing baselines in Table 3 run on the stronger of the two checkpoints. All reported sampler comparisons, including every result in Fig. 2, use the main checkpoint and differ only in the sampler.

Frozen backbones. On frozen FUDOKI the indicator f_t is maintained by the sampler and is not a denoiser input. The model sees only the current token configuration. Fixing tokens can therefore change the states visited by a frozen backbone, without a guarantee that those states follow its training distribution. The puzzle denoiser, by contrast, takes the status indicator as an input feature, so fixed positions are visible to it under every sampler.

C.1.1 Deriving the exact absorbing representation of uniform flow

This appendix derives the representation used in Section 3.1. We first follow the data-to-noise convention of Gourevitch et al. [12], then reverse time to recover the source-to-target convention of discrete flow matching. Tokens are identified with their one-hot vectors whenever they parameterize a categorical distribution.

Step 1: expand the conditional absorbing transition. Fix a coordinate j, a current token a^j, and an auxiliary absorbing token u^j. The conditional transition over $[t, t+h]$ is

$$\begin{aligned} q_{t+h|t}(z^j \mid a^j, u^j) &= \mathrm{Cat}\big(z^j; \alpha_{t+h|t} a^j + (1-\alpha_{t+h|t}) u^j\big) \\ &= \alpha_{t+h|t}\delta(z^j, a^j) + (1-\alpha_{t+h|t})\delta(z^j, u^j). \end{aligned} \tag{30}$$

Writing $\alpha_{t+h|t} = \alpha_{t+h}/\alpha_t$ and assuming differentiability gives

$$\alpha_{t+h|t} = 1 + \frac{\dot\alpha_t}{\alpha_t} h + o(h) = 1 - \nu_t h + o(h), \qquad \nu_t := -\frac{\dot\alpha_t}{\alpha_t} \geq 0, \tag{31}$$

because the corruption schedule α_t is nonincreasing. Substituting (31) into (30) yields

$$q_{t+h|t}(z^j \mid a^j, u^j) = \delta(z^j, a^j) + h\nu_t\big[\delta(z^j, u^j) - \delta(z^j, a^j)\big] + o(h). \tag{32}$$

Step 2: identify the conditional CTMC velocity. Comparing (32) with the infinitesimal CTMC update in (11) gives

$$u_t^{j,u}(z^j, a^j) = \nu_t\big[\delta(z^j, u^j) - \delta(z^j, a^j)\big]. \tag{33}$$

For $z^j \neq a^j$, this rate equals $\nu_t\delta(z^j, u^j) \geq 0$, and summing over z^j gives zero. It is therefore a valid CTMC velocity. If $a^j = u^j$, every rate vanishes. Otherwise, the only possible jump is from a^j to u^j, after which the coordinate remains at u^j. This verifies the absorbing property directly.

Step 3: marginalize the absorbing token. Let $U^j \sim \mathrm{Unif}(\mathcal{S})$. Since $\mathbb{E}[\delta(z^j, U^j)] = 1/|\mathcal{S}|$, averaging either (30) or (33) gives

$$\mathbb{E}_{U^j}\big[q_{t+h|t}(z^j \mid a^j, U^j)\big] = \alpha_{t+h|t}\delta(z^j, a^j) + \frac{1-\alpha_{t+h|t}}{|\mathcal{S}|}, \tag{34}$$

$$\mathbb{E}_{U^j}\left[u_t^{j,U}(z^j, a^j)\right] = \nu_t\left[\frac{1}{|\mathcal{S}|} - \delta(z^j, a^j)\right]. \tag{35}$$

Equation (34) is exactly the uniform-corruption transition, and (35) is its infinitesimal velocity. Thus the uniform transition is a mixture of conditional transitions, each with a single coordinate-specific absorbing token.

Step 4: reverse time to obtain the discrete-flow path. The diffusion convention above has $\alpha_0 = 1$ at data and α_t decreasing to 0 toward uniform noise. Let $r := 1 - t$ denote generation time and set $\kappa_r := \alpha_{1-r}$. Then $\kappa_0 = 0$, $\kappa_1 = 1$, and the endpoint-conditioned uniform flow path is

$$p^j_{r|1}(a^j \mid a_1^j) = \kappa_r\delta(a^j, a_1^j) + (1-\kappa_r)\frac{1}{|\mathcal{S}|}. \tag{36}$$

Introducing $U^j \sim \mathrm{Unif}(\mathcal{S})$ and using $\mathbb{E}[\delta(a^j, U^j)] = 1/|\mathcal{S}|$ gives the path identity

$$p^j_{r|1}(a^j \mid a_1^j) = \mathbb{E}_{U^j}\left[\kappa_r\delta(a^j, a_1^j) + (1-\kappa_r)\delta(a^j, U^j)\right]. \tag{37}$$

For a fixed realization $U^j = u^j \neq a_1^j$, the conditional path inside the expectation moves once from u^j to a_1^j. Its generation-time hazard is

$$h_\kappa(r) = \frac{\dot\kappa_r}{1-\kappa_r}. \tag{38}$$

Indeed, the source-token mass is $1 - \kappa_r$, so its outflow is $(1 - \kappa_r)h_\kappa(r) = \dot{\kappa}_r$, exactly the derivative of the mass assigned to a_1^j. If $u^j = a_1^j$, the conditional path is constant. Averaging these conditional paths proves the uniform discrete-flow identity used in Section 3.1.

Step 5: relate the exact lifting to selective absorption. The derivation above concerns the exact auxiliary-state representation of uniform transport. Appendix C.1 explains the distinct selective augmentation used by our sampler: Section 3.1 uses f_t^j to record a reliability-based terminal decision. Once $f_t^j = 1$, Eq. (2) sets every outgoing velocity at coordinate j to zero, making that absorbed value absorbing. This borrows the augmented absorbing-state structure while deliberately changing the native uniform-flow path according to the absorption policy.

C.2 The LEDFlow sampler: selective-absorption methodology

This subsection expands the implementation of Section 3 and separates the two operations performed at each grid point: an irreversible absorption decision and an otherwise unchanged uniform-flow update. No token-specific time map or modified CTMC velocity is introduced.

Active positions and absorption count. Only generated positions participate in the sampler. Their indicators are initialized as $f_{t_0}^j = 0$, and the active set before step k is $\mathcal{U}_k = \{j : f_{t_k}^j = 0\}$. The cosine schedule specifies the desired number

$$m_k = \left\lfloor M \cos\left(\frac{\pi}{2}\frac{k+1}{K}\right)\right\rfloor$$

of active positions after absorption. Consequently, $b_k = \max\{0, |\mathcal{U}_k| - m_k\}$ positions are made terminal at that step. Rounding can produce $b_k = 0$, in which case the sampler performs only the native flow update. At the final grid point, $m_{K-1} = 0$, so every remaining generated position is absorbed. This cosine absorption schedule is used for the FUDOKI and main Sudoku absorption comparisons. Both absorbing and non-absorbing Sudoku baselines use the same coordinatewise kinetic-optimal update. The schedule controls absorptions rather than native flow transitions. Explicit singleton-absorption ablations use $b_k = 1$ instead (Appendix D).

Entropy selection and value absorption. One denoiser evaluation provides the target marginal for each active position. We compute $H_\theta^j(s_k)$ and choose the b_k smallest values, with a deterministic index order used only to break exact ties. For every selected j, the sampler assigns the most likely target token and sets $f_{t_k}^j = 1$. Equation (2) then makes the position terminal: its outgoing velocity is zero at every later step and its value is copied unchanged. This is the only irreversible operation in LEDFlow.

Native flow on the remaining positions. Positions outside B_k^{LED} retain $f_{t_k}^j = 0$ and are advanced from t_k to t_{k+1} by the backbone's original uniform-flow solver. Thus, LEDFlow changes which predictions become final but does not change the learned posterior, the base probability path, or the numerical integration rule on active positions. Absorption and numerical integration are therefore separate design axes. For example, a time- or location-corrected update can be applied to the active positions before the same entropy-based absorption decision at the next grid point.

Matched-policy comparisons. The probability-margin, Info-Gain, and arbitrary-order baselines use the same generated-position mask, task-specific absorption schedule, most-likely-value update, and absorption rule as LEDFlow. They differ only in the score used to rank $\mathcal{U}_k$. This controlled construction attributes differences among these rows to the priority policy rather than to a different number of absorbed positions or a different uniform-flow update. In contrast, the native Euler and corrected-flow baselines set $f_t^j = 0$ throughout and therefore remain non-absorbing. They test numerical integration without the selective-absorption component.

Algorithm 1 gives the complete update order. The implementation first restricts selection to generated positions, refreshes the posterior and priority scores, absorbs the selected block, and finally applies the native velocity update only where the refreshed indicator remains zero. These steps directly implement Eqs. (2), (4), and (3).

C.3 THEORETICAL ANALYSIS

C.3.1 KL MOTIVATION AND LOCAL ABSORPTION-ERROR DECOMPOSITION

Why measure decoding error with KL divergence? The starting point follows the sequential-decoding formulation of Xu et al. [43]. Let $\pi(y)$ be a target distribution, $\sigma = (\sigma_1, \dots, \sigma_M)$ an unmasking order, and $v_\phi(\sigma \mid y)$ the probability assigned to that order along the teacher-forced trajectory of y. A denoiser assigns the path likelihood

$$p_\theta(y \mid \sigma) = \prod_{i=1}^{M} p_\theta^{\sigma_i}(y^{\sigma_i} \mid y^{\sigma_{<i}}), \tag{39}$$

and the sequential decoder induces $P^{\mathrm{seq}}_{\theta,\phi}(y) = \sum_\sigma v_\phi(\sigma \mid y) p_\theta(y \mid \sigma)$. Define the data-policy and model-policy joint laws

$$Q_\phi(y, \sigma) := \pi(y) v_\phi(\sigma \mid y), \qquad P_{\theta,\phi}(y, \sigma) := p_\theta(y \mid \sigma) v_\phi(\sigma \mid y). \tag{40}$$

Because both laws use the same order policy, its log-density cancels in their KL divergence:

$$\begin{aligned} D_{\mathrm{KL}}(Q_\phi \,\|\, P_{\theta,\phi}) &= \mathbb{E}_{y\sim\pi,\, \sigma\sim v_\phi(\cdot|y)} \left[\log \frac{\pi(y)}{p_\theta(y \mid \sigma)} \right] \\ &= -H(\pi) - \mathbb{E}_{y\sim\pi,\, \sigma\sim v_\phi(\cdot|y)} \left[\log p_\theta(y \mid \sigma) \right]. \end{aligned} \tag{41}$$

Marginalization from (y, σ) to y and the data-processing inequality then give

$$D_{\mathrm{KL}}\left(\pi \,\middle\|\, P^{\mathrm{seq}}_{\theta,\phi} \right) \le D_{\mathrm{KL}}(Q_\phi \,\|\, P_{\theta,\phi}). \tag{42}$$

Equations (41) to (42) explain why KL is useful for studying absorption decisions: it measures distributional mismatch at the decoder output, while an upper bound can be evaluated as accumulated prediction loss along an ordered trajectory. The cited work also considers an ordered block schedule $(B_1, \dots, B_K)$. If its parallel law samples positions in each block independently from the corresponding sequential-law marginals, then its parallelization gap is

$$D_{\mathrm{KL}}\left(P^{\mathrm{seq}}_{\theta,\phi} \,\middle\|\, P^{\mathrm{par}}_{\theta,\phi} \right) = \sum_{k=1}^{K} \mathbb{E}_{P^{\mathrm{seq}}_{\theta,\phi}} \left[\mathcal{TC}_{P^{\mathrm{seq}}_{\theta,\phi}}(Y^{B_k} \mid Y^{B_{<k}}) \right]. \tag{43}$$

This identity measures the dependence lost by parallelizing a sequential model law. Our local question uses a different reference: it compares the oracle joint posterior at an absorption event directly with the product of learned denoiser marginals. Proposition 2 therefore retains a total-correlation term for dependence loss and adds a separate term for marginal prediction error.

Step 1: abbreviate the three distributions. Fix the reachable state s_k and block B_k from Section 3.3. Write

$$P_B(b) := p^q_{1|s_k}(a_1^{B_k} = b), \quad P_j(b_j) := p^q_{1|s_k}(a_1^j = b_j), \quad Q_j(b_j) := \mathbb{Q}^\theta_{1|t_k}(a_1^j = b_j \mid s_k). \tag{44}$$

The oracle distribution P_B retains dependence among the positions in B_k, whereas the implemented absorption kernel is the product $\prod_{j\in B_k} Q_j$. We assume the latter is positive on the support of P_B. Otherwise the relevant KL divergence is infinite and the identity below holds in the extended-real sense.

Step 2: insert the product of oracle marginals. For every b in the support of P_B,

$$\log \frac{P_B(b)}{\prod_{j\in B_k} Q_j(b_j)} = \log \frac{P_B(b)}{\prod_{j\in B_k} P_j(b_j)} + \sum_{j\in B_k} \log \frac{P_j(b_j)}{Q_j(b_j)}. \tag{45}$$

Taking expectation under P_B separates the local absorption loss into two terms:

$$\begin{aligned} \mathcal{L}_{\mathrm{abs}}(B_k; s_k) = &\sum_b P_B(b) \log \frac{P_B(b)}{\prod_{j\in B_k} P_j(b_j)} \\ &+ \sum_{j\in B_k} \sum_b P_B(b) \log \frac{P_j(b_j)}{Q_j(b_j)}. \end{aligned} \tag{46}$$

Lemma 2 (Conditional total correlation of an absorption block)**.** *For any reachable state s_k and absorption block B_k,*

$$D_{\mathrm{KL}}\left(p^q_{1|s_k}(a_1^{B_k}) \,\middle\|\, \prod_{j\in B_k} p^q_{1|s_k}(a_1^j)\right) = \mathcal{TC}_q(a_1^{B_k} \mid s_k) \geq 0. \tag{47}$$

The quantity vanishes exactly when the positions in B_k are conditionally independent under the oracle posterior, including every singleton block.

Step 3: identify the joint term. The first term in (46) is

$$\begin{aligned} D_{\mathrm{KL}}\left(P_B \,\middle\|\, \prod_{j\in B_k} P_j\right) &= \sum_b P_B(b)\log P_B(b) - \sum_{j\in B_k}\sum_b P_B(b)\log P_j(b_j) \\ &= -H_q(a_1^{B_k} \mid s_k) + \sum_{j\in B_k} H_q(a_1^j \mid s_k) \\ &= \mathcal{TC}_q(a_1^{B_k} \mid s_k) = \mathcal{E}_{\mathrm{joint}}(B_k; s_k). \end{aligned} \tag{48}$$

The second equality marginalizes P_B over all coordinates except j. Since it is itself a KL divergence, the joint term is nonnegative and vanishes exactly when the coordinates in B_k are conditionally independent under the oracle posterior. In particular, it is zero for $|B_k| = 1$.

Step 4: identify the conditional term. For each $j \in B_k$, marginalizing the second term in (46) over b_{-j} gives

$$\begin{aligned} \sum_b P_B(b)\log\frac{P_j(b_j)}{Q_j(b_j)} &= \sum_{b_j} P_j(b_j)\log\frac{P_j(b_j)}{Q_j(b_j)} \\ &= D_{\mathrm{KL}}(P_j \| Q_j). \end{aligned} \tag{49}$$

Summing (49) over the block yields $\mathcal{E}_{\mathrm{cond}}(B_k; s_k)$. Combining (48) and (49) proves Eq. (5).

Step 5: interpret the limiting cases. If the block is a singleton, independent sampling introduces no factorization error and only the denoiser KL remains. If every denoiser marginal equals its oracle marginal, the conditional term vanishes, but a multi-position block can still incur the joint term because the product kernel does not reproduce conditional dependence. Both terms vanish when the block is conditionally independent and every marginal is exact. For a random or adaptive absorption policy, the same identity holds after conditioning on the realized pair (s_k, B_k). Taking expectation over that pair preserves the decomposition. Consequently, the policy affects the conditional term through the states at which predictions become irreversible, while block size determines whether an additional factorization cost can arise. When the oracle conditional at s_k is a point mass, as on unique-solution puzzles, every conditional entropy in the total correlation is zero and the joint term vanishes for every block regardless of its size. A nondegenerate oracle, such as the multi-solution Sudoku corpus or graph coloring, is required to observe it.

C.3.2 Proof of Corollary 1

Proposition 4 (Conditional error and absorption risk)**.** *For a reachable state s_k and an absorption block B_k with $|B_k| = b$, let $\hat{z}^j = \arg\max_z \mathbb{Q}^\theta_{1|t_k}(a_1^j = z \mid s_k)$, $c_j = p^q_{1|s_k}(a_1^j = \hat{z}^j)$, and $c_j^\star = \max_z p^q_{1|s_k}(a_1^j = z)$. Then*

$$\sum_{j\in B_k}(1-c_j) \leq \sum_{j\in B_k}(1-c_j^\star) + \sqrt{2b\,\mathcal{E}_{\mathrm{cond}}(B_k; s_k)}. \tag{50}$$

Proof of Proposition 4. Fix a reachable state s_k and a position $j \in B_k$, and abbreviate the oracle and model marginals at that position by $p := p^q_{1|s_k}(a_1^j = \cdot)$ and $\mathbb{Q} := \mathbb{Q}^\theta_{1|t_k}(a_1^j = \cdot \mid s_k)$. Let

$\hat{z}^j = \arg\max_z \mathbb{Q}(z)$ be the token written by Eq. 4 and $z^{j\star} = \arg\max_z p(z)$ the oracle mode, so that $c_j = p(\hat{z}^j)$ and $c_j^\star = p(z^{j\star})$. Adding and subtracting $\mathbb{Q}$ at both tokens,

$$c_j^\star - c_j = p(z^{j\star}) - p(\hat{z}^j) = \big[p(z^{j\star}) - \mathbb{Q}(z^{j\star})\big] + \underbrace{\big[\mathbb{Q}(z^{j\star}) - \mathbb{Q}(\hat{z}^j)\big]}_{\le 0} + \big[\mathbb{Q}(\hat{z}^j) - p(\hat{z}^j)\big],$$

where the middle bracket is nonpositive because $\hat{z}^j$ maximizes $\mathbb{Q}$. Since $\mathrm{TV}(p, \mathbb{Q}) = \max_{A \subseteq \mathcal{S}} |p(A) - \mathbb{Q}(A)|$ dominates the discrepancy on any singleton, each outer bracket is at most $\mathrm{TV}(p, \mathbb{Q})$, giving the plug-in bound $c_j^\star - c_j \le 2\,\mathrm{TV}(p, \mathbb{Q})$. Pinsker's inequality $\mathrm{TV}(p, \mathbb{Q}) \le \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p \,\|\, \mathbb{Q})}$ then yields

$$(1 - c_j) - \big(1 - c_j^\star\big) \;\le\; \sqrt{2\, D_{\mathrm{KL}}(p \,\|\, \mathbb{Q})},$$

whose divergence is exactly the j-th summand of $\mathcal{E}_{\mathrm{cond}}(B_k; s_k)$ in Eq. 5. Denote this divergence by D_j. Summing over $j \in B_k$ and applying Cauchy-Schwarz to the b square roots,

$$\sum_{j \in B_k} \sqrt{2 D_j} \;\le\; \sqrt{2b \sum_{j \in B_k} D_j} \;=\; \sqrt{2b\, \mathcal{E}_{\mathrm{cond}}(B_k; s_k)},$$

which gives Eq. 50. □

Two remarks delimit the result. First, the bound separates the block's oracle ambiguity $\sum_{j \in B_k}(1 - c_j^\star)$, which is a property of q at s_k for the chosen block, from the model-dependent excess, which is bounded by a function of $\mathcal{E}_{\mathrm{cond}}$. The ambiguity term does depend on which block is selected, since a different block sums different positions, so a position that is genuinely ambiguous under the oracle is not rendered reliable by a small conditional KL. The absorption policy of Section 3 targets only the second term. Second, the $\sqrt{\cdot}$ and the Cauchy-Schwarz step are both loose, and the statement is a control of an upper bound rather than a ranking of realized risk: decreasing $\mathcal{E}_{\mathrm{cond}}$ tightens the bound on the excess absorption risk, but a smaller bound does not by itself guarantee a smaller realized excess. This is what motivates using the conditional term as the objective in Section 3, and it is the reason the policy is evaluated empirically rather than claimed to minimize realized risk. Combined with Definition 1, the two bounds are $(1 - c_j) - (1 - c_j^\star) \le \sqrt{2 D_j}$ and $D_j \le \phi(H_\theta^j)$, connecting the absorption error to the entropy criterion used at inference.

C.3.3 Entropy-guided absorption: proof of Theorem 1

This appendix proves the local absorption result in Theorem 1, derives Lemma 1, and records the scope of both claims. Throughout, the sampler state s_k is reachable under the oracle process and $\mathcal{U}_k$ is the set of positions that remain active immediately before absorption event k.

Scope of the main-text results. A wrong absorption can leave the oracle-reachable set, in which case the oracle conditional at the resulting state may be undefined. The statements below therefore control error only while the trajectory remains oracle-consistent. Theorem 1 is a conditional exchange argument for a surrogate bound, not a derivation of entropy-error regularity, and it does not establish optimal terminal accuracy or control the trajectory after an oracle-inconsistent absorption. Low entropy alone does not imply correctness: Definition 1 is an envelope assumption whose empirical coverage on the Sudoku denoisers, including the relaxed offset ϕ_0 and the confidently-wrong tail, is reported in Appendix C.3.6. The observed coverage is descriptive calibration evidence and does not establish the pointwise assumption or a held-out guarantee. Corollary 1 shows that entropy selection minimizes the excess-risk bound for a fixed absorption count, but it need not minimize total risk, because the oracle term also varies across blocks. For $b > 1$, even exact marginals need not produce a jointly valid block of modes. Lemma 1 is worst-case additive, so the realized lookahead error can grow more slowly than η_m. It compares score sensitivity rather than terminal losses under the two policies, and does not prove that local entropy is better whenever the bound is large.

Relaxed entropy-error regularity. The relaxed inequality in Eq. 51 adds a position-independent constant ϕ_0 to accommodate the confidently-wrong tail at entropy near zero. It holds on 99.4% of sampled states for the 6.4M denoiser in the diagnostic of Appendix C.3.6. As ϕ_0 is the same for every active position, it shifts $\Phi(B_k; s_k)$ by the constant $b\phi_0$ and leaves the minimizer of $\sum_{j \in B_k} \phi\Big(H_\theta^j(s_k)\Big)$

unchanged. The bound remains conditional on the relaxed inequality holding pointwise. An offset chosen as a mean provides empirical coverage, not a universal upper bound.

$$D_{\mathrm{KL}}\Big(p^q_{1|s_k}(a^j_1)\,\Big\|\,\mathbb{Q}^\theta_{1|t_k}(a^j_1 \mid s_k)\Big) \le \phi_0 + \phi\Big(H^j_\theta(s_k)\Big), \qquad \phi_0 \ge 0, \tag{51}$$

Alternative uncertainty scores. For maximum predicted probability and probability margin, the corresponding uncertainty scores are $1 - p_{(1)}$ and $1 - (p_{(1)} - p_{(2)})$, respectively, where $p_{(1)} \ge p_{(2)}$ are the two largest posterior probabilities. Minimizing these scores selects the highest-confidence positions, and entropy in Eq. 3 can be replaced by the corresponding score to obtain a different ordering policy under the same absorption count. Definition 1 is stated for predictive entropy. Entropy regularity does not imply the analogous conditions for maximum probability or probability margin, and the scores can rank positions differently, so the resulting policies are compared empirically in Appendix E.5.

Oracle objective and tractable envelope. Fix an absorption count $b \ge 1$ and recall $\mathfrak{B}_k(b) = \{B \subseteq \mathcal{U}_k : |B| = b\}$. An oracle that knew the realized target sequence could fix every absorbed position correctly, eliminating errors propagated from earlier irreversible decisions. For the inference setting, where absorbed values are still predicted by the denoiser, we use the true conditional posterior only as an analytical reference. By Proposition 2, this oracle reference would select

$$B^{\mathrm{oracle}}_k \in \arg\min_{B \in \mathfrak{B}_k(b)} \underbrace{\sum_{j \in B} D_{\mathrm{KL}}\Big(p^q_{1|s_k}(a^j_1)\,\Big\|\,\mathbb{Q}^\theta_{1|t_k}(a^j_1 \mid s_k)\Big)}_{\mathcal{E}_{\mathrm{cond}}(B;s_k)}.$$

This decision cannot be evaluated at inference because $p^q_{1|s_k}$ is unavailable. Under Definition 1, however, every summand admits the denoiser-computable envelope

$$D_{\mathrm{KL}}\Big(p^q_{1|s_k}(a^j_1)\,\Big\|\,\mathbb{Q}^\theta_{1|t_k}(a^j_1 \mid s_k)\Big) \le \phi\Big(H^j_\theta(s_k)\Big).$$

Summing over a candidate block gives

$$\mathcal{E}_{\mathrm{cond}}(B; s_k) \le \Phi(B; s_k) := \sum_{j \in B} \phi\Big(H^j_\theta(s_k)\Big).$$

The entropy policy need not coincide with B^{oracle}_k. The claim is that it exactly minimizes this tractable upper bound.

Proof of Theorem 1.

Proof. Fix an absorption count b and let B_{low} contain b positions having the smallest predictive entropies. Consider any other block B of size b. If $B \ne B_{\mathrm{low}}$, there are $i \in B_{\mathrm{low}} \setminus B$ and $j \in B \setminus B_{\mathrm{low}}$ with $H^i_\theta(s_k) \le H^j_\theta(s_k)$. Since ϕ is nondecreasing, exchanging j for i cannot increase Φ because

$$\Phi((B \setminus \{j\}) \cup \{i\}; s_k) - \Phi(B; s_k) = \phi\big(H^i_\theta(s_k)\big) - \phi\Big(H^j_\theta(s_k)\Big) \le 0.$$

Repeating the exchange produces B_{low}, proving that it minimizes Φ over $\mathfrak{B}_k(b)$.

When $b = 1$, Eq. (8) follows immediately. Moreover, conditional total correlation vanishes for a singleton:

$$\mathcal{E}_{\mathrm{joint}}(\{j\}; s_k) = H_q(a^j_1 \mid s_k) - H_q(a^j_1 \mid s_k) = 0.$$

Hence Proposition 2 reduces to $\mathcal{L}_{\mathrm{abs}}(\{j\}; s_k) = \mathcal{E}_{\mathrm{cond}}(\{j\}; s_k)$, and the lowest entropy position minimizes the upper bound on the complete local absorption error.

For part (ii), convergence to the oracle posterior on reachable states means that, for every admissible j and $z^j \in \mathcal{S}$,

$$\mathbb{Q}^\theta_{1|t_k}(a^j_1 = z^j \mid s_k) \longrightarrow p^q_{1|s_k}(a^j_1 = z^j).$$

Because $\mathcal{S}$ is finite, each term with positive oracle mass converges continuously and each zero-mass term contributes zero. Therefore,

$$D_{\mathrm{KL}}\Big(p^q_{1|s_k}(a^j_1)\,\Big\|\,\mathbb{Q}^\theta_{1|t_k}(a^j_1 \mid s_k)\Big) \longrightarrow 0.$$

Every admissible block contains finitely many positions, so

$$\mathcal{E}_{\mathrm{cond}}(B; s_k) = \sum_{j \in B} D_{\mathrm{KL}}\Big(p^q_{1|s_k}(a^j_1) \,\Big\|\, \mathbb{Q}^\theta_{1|t_k}(a^j_1 \mid s_k)\Big) \longrightarrow 0.$$

This holds for every admissible block and thus for the block selected by any absorption policy, which proves part (ii). □

Why the result is local. The exchange argument holds at a fixed state and fixed absorption count. Absorbing a prediction changes both the active set and the conditioning state used by the denoiser at later events. Theorem 1 therefore controls the next block contribution to $\mathcal{E}_{\mathrm{cond}}$. It does not claim that greedy entropy selection minimizes the final loss over every complete absorption trajectory. When $b > 1$, it does not optimize the joint term $\mathcal{E}_{\mathrm{joint}}(B; s_k)$, which can vary with the selected block.

Reachable states. The regularity condition is stated on oracle-reachable states, where every absorbed partial configuration has positive probability under q and its oracle conditional is defined. A learned sampler can leave this set after an incorrect irreversible prediction. At such a state, $p^q_{1|s_k}$ may be undefined if no valid completion remains, so the KL condition cannot be evaluated. The theorem should consequently be read as controlling the error incurred while the sampler remains on an oracle-consistent trajectory. Its experiments test whether this local rule also reduces departures from that set. They do not turn the pointwise premise into a global guarantee.

Sampling versus most-likely absorption. The KL decomposition in Proposition 2 uses the categorical denoiser marginal as the prediction kernel. Algorithm 1 instead absorbs its most likely value. The two kernels are generally different, including on unique-solution tasks unless the learned marginal is itself degenerate. We therefore use the divergence analysis to justify the position-selection rule and evaluate the deployed most-likely-value rule empirically rather than claiming equality of the two output laws.

Remark 1 (Entropy at inference and training). The inference policy absorbs low-entropy positions because they have the smallest next-block conditional-error bound. This differs from weighting a training loss, which changes how the denoiser is learned. Theorem 1 concerns only the inference decision and does not assume that predictive entropy equals downstream information gain.

C.3.4 Derivation of the global lookahead bound

Recall that $m = |\mathcal{U}(s)|$ and $R_r(s) = \sum_{j \in \mathcal{U}(s)} h^r_j(s)$ for $r \in \{q, \theta\}$. Write

$$G^r_j(s) = R_r(s) - \mathbb{E}_{z \sim P^j_r(\cdot|s)} R_r(s^{j,z}), \qquad D^r_{jj'}(s) := G^r_j(s) - G^r_{j'}(s),$$

so that $R_r(s)$, which does not depend on the candidate, cancels exactly in $D^r_{jj'}$. The selection made by Info-Gain depends on the scores only through the differences $D^\theta_{jj'}$, so these are the quantities the bound must control.

For a single successor aggregate the assumed pointwise entropy error gives $|R_\theta(s^{j,z}) - R_q(s^{j,z})| \le (m-1)\delta$, since $s^{j,z}$ leaves $m-1$ positions active. Moreover $0 \le R_q(s^{j,z}) \le (m-1)\log|\mathcal{S}|$, so the total-variation assumption implies

$$\Big|\mathbb{E}_{z \sim P^j_\theta} R_q(s^{j,z}) - \mathbb{E}_{z \sim P^j_q} R_q(s^{j,z})\Big| \le (m-1)\,\epsilon \log|\mathcal{S}|,$$

where we used $|\mathbb{E}_P f - \mathbb{E}_{P'} f| \le \mathrm{TV}(P, P')\,(\sup f - \inf f)$ with TV in the $\frac{1}{2}\ell_1$ convention of Section 3.3. Combining the two displays, the error of the expected successor aggregate for a single candidate is at most $\eta_m := (m-1)(\delta + \epsilon \log|\mathcal{S}|)$.

Successors outside the oracle support. The expectation over $z \sim P^j_\theta(\cdot \mid s)$ can place mass on a value z with $P^j_q(z \mid s) = 0$. The successor $s^{j,z}$ then has no oracle conditional, so $h^q_{j'}(s^{j,z})$ is undefined. We adopt the convention $h^q_{j'}(s^{j,z}) := \log|\mathcal{S}|$ for every active j' at such a successor, treating an oracle-inconsistent absorption as leaving the remaining positions maximally uncertain, and we require the pointwise entropy condition $|h^\theta - h^q| \le \delta$ only at successors of positive model

probability with this extension in force. Any other bounded extension leaves the total-variation step unchanged because it uses only the range $[0, (m-1)\log|\mathcal{S}|]$ of R_q. If one prefers not to impose the entropy condition at off-support successors at all, their total model mass is at most $\mathrm{TV}(P_\theta^j, P_q^j) \le \epsilon$, which adds at most $\epsilon(m-1)\log|\mathcal{S}|$ to the first display and changes η_m to $(m-1)(\delta + 2\epsilon\log|\mathcal{S}|)$ without altering the form or reading of the lemma.

Since the current aggregate cancels,

$$\left|D^\theta_{jj'}(s) - D^q_{jj'}(s)\right| \le \left|\mathbb{E}_{P_\theta^j} R_\theta(s^{j,z}) - \mathbb{E}_{P_q^j} R_q(s^{j,z})\right| + \left|\mathbb{E}_{P_\theta^{j'}} R_\theta(s^{j',z}) - \mathbb{E}_{P_q^{j'}} R_q(s^{j',z})\right| \le 2\eta_m.$$

For the model and oracle maximizers, insert and subtract the model scores and note that the differences again eliminate $R_r(s)$:

$$G^q_{j^\star} - G^q_{\hat{j}} = D^q_{j^\star\hat{j}} \le \underbrace{D^\theta_{j^\star\hat{j}}}_{\le 0} + \left|D^\theta_{j^\star\hat{j}} - D^q_{j^\star\hat{j}}\right| \le 2\eta_m.$$

The analogous two-error argument for the local entropy minimizers gives

$$h^q_{\hat{j}_{\mathrm{ent}}}(s) - \min_j h^q_j(s) \le 2\delta,$$

which completes the proof of Lemma 1. □

Interpretation of the lookahead result. Three qualifications delimit what the lemma does and does not say. First, it bounds score estimation, not the realized terminal loss: Info-Gain remains effective whenever $2\eta_m$ is small relative to the oracle margin $D^q_{j^\star j'}$ between its best and second-best candidates, and the decision-relevant diagnostic is therefore the ratio of the two rather than η_m alone. Second, because the bound is stated on score *differences*, it is invariant to the normalization of the aggregate: rescaling every candidate score by a common positive factor rescales η_m and the margin identically and changes neither the selected candidate nor the comparison with the local rule. The window dependence that remains comes from the $m-1$ positions genuinely carried inside each successor aggregate, not from the choice of units. Third, the additivity over those positions is worst case. If the per-position entropy errors are not adversarially aligned, a concentration argument would give $O(\sqrt{m-1})$ in place of $O(m-1)$, so the lemma should be read as an upper envelope on how the lookahead window can amplify error rather than as the typical rate. The result isolates the source of accumulation: global lookahead combines errors over the unresolved positions, whereas local entropy selection estimates one position at the current state.

C.3.5 Additional comparison with Info-Gain

At a reachable state s, couple the entropy and Info-Gain policies so that they differ only in the next absorbed position and share the continuation. Let L_{local} and L_{IG} be their final losses. Let E_s be the event that Info-Gain selects an oracle-consistent position and set $\kappa_\theta(s) = \mathbb{P}(E_s \mid s)$. Define

$$B_s = \mathbb{E}[L_{\mathrm{local}} - L_{\mathrm{IG}} \mid E_s, s], \qquad C_s = \mathbb{E}[L_{\mathrm{IG}} - L_{\mathrm{local}} \mid E_s^c, s].$$

When correct lookahead helps and incorrect lookahead hurts, $B_s, C_s \ge 0$ measure the corresponding benefit and cost.

Proposition 5 (When the local entropy policy beats Info-Gain)**.** *At any state satisfying the definitions above,*

$$\mathbb{E}[L_{\mathrm{IG}} - L_{\mathrm{local}} \mid s] = (1 - \kappa_\theta(s))C_s - \kappa_\theta(s)B_s.$$

If $B_s + C_s > 0$, the local entropy policy has no larger expected final loss if and only if

$$\kappa_\theta(s) \le \frac{C_s}{B_s + C_s}. \tag{52}$$

Proof. Conditioning on E_s and E_s^c gives the displayed identity. Requiring its right-hand side to be nonnegative and rearranging gives Eq. (52). □

This identity does not assert that local entropy always outperforms lookahead. It makes the comparison depend on how often the lookahead choice is oracle-consistent and on the relative costs of its correct and incorrect choices. These quantities can be estimated by paired continuations on tasks with reference solutions, as illustrated by the paired continuation estimates below.

Table 5: Measured comparison on Sudoku-Extreme (600 paired reachable states per denoiser). κ_θ is the probability that Info-Gain selects an oracle-consistent position and $\kappa^\star = C_s/(B_s + C_s)$. The local entropy policy has no larger expected loss when $\kappa_\theta \le \kappa^\star$.

Denoiser	κ_θ	$\kappa^\star$	B_s	C_s	better policy
6.4M	0.527	0.698	0.0043	0.0100	entropy
21.3M	0.695	0.397	0.0038	0.0025	Info-Gain

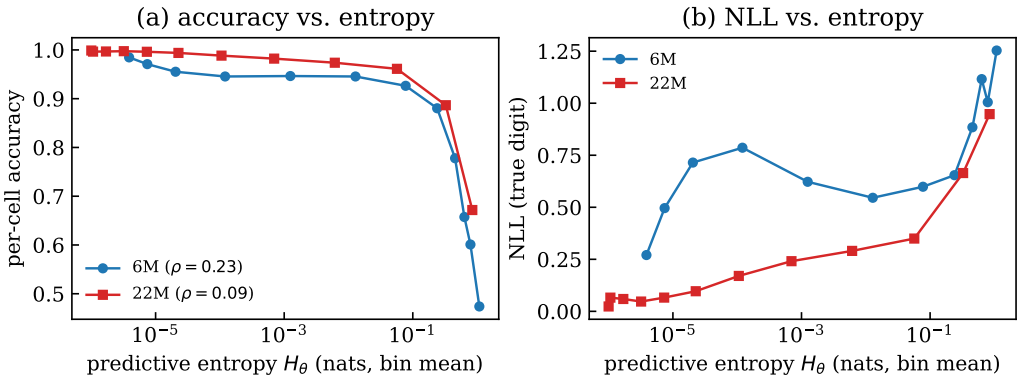


Figure 6: **Entropy-accuracy correlation, both denoisers.** Same binned trajectory data as Table 7 (all 12 equal-count bins, 6.4M and 21.3M), plotted against the model's predictive entropy H_θ on a log axis. (a) Per-cell accuracy (1−err) decreases with entropy at both capacities, with Spearman $\rho(H, \text{err}) = 0.23$ (6.4M) and 0.09 (21.3M) (the low ρ reflects that most mass sits in the flat, near-zero-entropy, near-ceiling regime common to reachable Sudoku states, and the drop is concentrated in the top bins, consistent with the nonlinear slope L_ϕ reported below). (b) NLL of the true digit rises with entropy at both capacities. The 21.3M denoiser's NLL curve sits uniformly below the 6.4M curve, the calibration improvement with capacity that motivates comparing the two models throughout this appendix.

Measured comparison. On Sudoku-Extreme, paired continuations estimate the three quantities in Proposition 5 and recover the observed winner at both denoiser sizes (Table 5). This is a diagnostic of the stated identity, not evidence that model size itself defines the threshold. It is also a per-state comparison with a shared continuation: end-to-end on the same 21.3M denoiser the entropy policy still solves more puzzles than Info-Gain (0.964 against 0.947 at n=1000, p=0.027, Table 20), so the per-state advantage of lookahead at states where it is oracle-consistent has not yet translated into a trajectory-level win at this capacity.

C.3.6 Calibration and miscalibration

The following analysis provides an empirical diagnostic of the entropy-error relationship in Definition 1 and measures the surrogate slack used by Theorem 1. Per-cell error rises with the denoiser's predictive entropy, from 0.015 in the lowest-entropy bin to 0.526 in the highest (Table 7), and the empirical slope decreases from the 6.4M to the 21.3M denoiser. The two reported constants move in opposite directions with capacity, the slope falling from 0.45 to 0.37 while the surrogate gap ε rises from 0.21 to 0.30, and the two measure different things. The slope describes how sharply error grows with the model's own uncertainty, and a better-calibrated denoiser improves it. The surrogate gap describes how well a *static* score tracks the *path-conditional* predictive entropy, and a stronger denoiser exploits constraint propagation more aggressively, so its entropies move more along the path and a fixed ranking tracks them less well. A capable denoiser therefore makes the entropy-error relationship tighter and the fixed-score policy's job harder at the same time. The fixed score nevertheless recovers only part of the adaptive policy's gain on the weaker model.

Table 6: Offset ϕ_0 summaries for the relaxed envelope (nats). The mean column gives the offsets used to compute the reported coverage. The other columns describe the distribution's upper tail.

Denoiser	Mean	95th percentile	99th percentile	Maximum
6.4M	0.57	0.002	16.70	20.85
21.3M	0.12	0.001	0.003	22.33

Table 7: Entropy-error diagnostic (Definition 1) on Nikoli (n=100): per-cell error and NLL of the true digit, binned by the model's predictive entropy H_θ over the whole sampling trajectory (6.4M denoiser, representative equal-count bins). Per-cell error rises monotonically with entropy. The bottom row reports the empirical error-entropy slope and surrogate error ε for both denoisers.

H_θ (bin mean)	0.00	0.24	0.46	0.64	0.79	1.08
per-cell error	0.015	0.120	0.222	0.343	0.399	0.526
NLL (true digit)	0.27	0.65	0.88	1.12	1.01	1.25

6.4M: empirical slope $= 0.45$, ε=0.21 | 21.3M: empirical slope $= 0.37$, ε=0.30
95th-pct envelope slope: $L_{\mathrm{env},0.95}$≈1.70/1.34 (6.4M/21.3M). Reported confidently-wrong fraction 0.99%/0.35%. Strict-ϕ compliance 95.5%/97.6%, relaxed-ϕ compliance 99.4%/99.7%.

Pointwise envelope. Definition 1 requires a *pointwise* bound $\mathrm{KL} \le \phi(H_\theta)$, not merely the bin-mean trend above. Fitting a nondecreasing 95th-percentile upper envelope to the full per-cell cloud (158,191 reachable partial configurations) gives an empirical envelope slope $L_{\mathrm{env},0.95} \approx 1.70$ (6.4M) and 1.34 (21.3M), about 3.6 to 3.8 times the bin-mean OLS slope. The robust calibration summary reports a small *confidently-wrong* fraction (0.99% at 6.4M and 0.35% at 21.3M), with near-zero-entropy NLL reaching approximately 20.9 and 22.3 nats, respectively. The fitted inequality with $\phi(0) = 0$ at the mean slope covers 95.5% (6.4M) and 97.6% (21.3M) of sampled reachable states. The relaxed envelope uses the mean offset $\phi_0 = 0.57$ nats for 6.4M and 0.12 nats for 21.3M (Table 6), under which $\mathrm{KL} \le \phi_0 + \phi(H_\theta)$ holds on 99.4% and 99.7% of sampled states. For any fixed position-independent offset, the additive contribution $b\phi_0$ preserves the entropy argmin at a fixed absorption count, but a mean offset does not establish a pointwise upper bound. Substituting the updated envelope slopes into the diagnostic expression $2L\varepsilon M$ gives 39.8 and 45.2 nats, respectively, compared with 10.5 and 12.4 using the bin-mean slopes. These fitted slopes and empirical coverage rates do not establish a uniform bound on every reachable state.

Oracle versus model entropy. To separate "entropy is the ordering signal" from "*the model's* entropy is the signal," we compare absorbing by the exact oracle entropy H_q against absorbing by the model's H_θ. Here H_q is the entropy of the completion-count distribution over viable digits, computed on a multi-solution corpus where $H_q > 0$ is non-degenerate. Over 120 multi-solution puzzles (median 23 completions), H_θ correlates strongly with H_q ($r = 0.88$ at 6.4M, 0.90 at 21.3M), and ordering by the model's offline H_θ matches or exceeds ordering by the exact H_q (valid-completion rate 0.958 vs. 0.900 at 6.4M, and 0.983 vs. 0.983 at 21.3M), while adaptively recomputing H_θ reaches 1.000. In this experiment, model entropy matches or exceeds the tested oracle-entropy ordering. The improvement from adaptive recomputation suggests that updating the score as the active state changes matters beyond the initial ranking.

D Experimental settings and hyperparameters

This appendix collects the experimental protocols and hyperparameters for §4, following the order of the main experiments.

Sampler protocol. At each grid point, **LEDFlow** recomputes predictive entropy from the current sampler state and absorbs the lowest-entropy active positions under the absorption count. Active positions then follow the backbone's unchanged update, while absorbed positions remain fixed under Eq. (2). Top K-margin, Info-Gain, and arbitrary-order policies use the same absorption schedule, value update, and absorption rule and differ only in the score used to select the block. The absorption schedule is task specific: on the FUDOKI understanding, text-to-image, and mathematics

Table 8: Two attempts to make a fixed external priority beat adaptive entropy-guided absorption by inducing miscalibration. *internal* is adaptive entropy and *external* is the best of the fixed external score and logical forcedness. Top: density shift (train dense, test sparse). Bottom: wrong constraint (train Latin, test Sudoku) over a difficulty sweep. The external order wins only in the floored hard band (a non-significant $\sim$2-puzzle margin). Wherever the denoiser is capable, confidence wins.

probe	regime (denoiser)	internal	external	gap
(a) density shift, test givens 26–31, n=185				
	calibrated (ECE 0.008)	0.973	0.746	+0.227
	overconfident (ECE 0.074)	0.827	0.546	+0.281
(b) wrong constraint: Latin-trained (box-blind), test Sudoku				
	hard (givens 32–40, floored)	0.109	**0.124**	−0.015*
	medium (givens 40–48)	0.637	0.630	+0.007
	easy (givens 44–52)	0.827	0.810	+0.017

*floored regime ($\sim$0.11 solve). $\sim$2-puzzle margin at n=137, not significant.

Table 9: Protocol map for Sudoku results. All samplers at a given capacity share the same checkpoint. Each ID identifies a separate reported experiment, not an interchangeable estimate. K is the number of sampling steps. It need not equal the number of singleton absorptions. Solve accuracies and policy labels are those of Table 3.

ID	Checkpoint / location	Decoder or absorption protocol	Cohort / sampling steps
M	Shared task-specific 6.4M; Table 3, all rows; Fig. 2	Kinetic-optimal updates; absorbing rows add a cosine absorption schedule	Matched across all rows: same puzzles, Generated/Nikoli/Extreme $n = 500/100/700$, five seeds, $K = 64$; Fig. 2 traces the Nikoli subset
A	Task-specific 6.4M; Table 24	Ordering-score ablation; cosine absorption schedule	Temperature sweep, seeds $0, 1$, $K = 64$
L	Shared model at each capacity; Table 23	Singleton absorptions	Each set $n = 100$, five seeds
G	6.4M and 21.3M; Table 20	Separate singleton-ordering study	Generated $n = 1000$, 22–34 givens

harnesses the cosine schedule of Appendix C.2 sets b_k at each of the K grid points. The main Sudoku comparisons also use a cosine absorption schedule over $K = 64$ steps, with coordinatewise kinetic-optimal updates in both absorbing and non-absorbing arms (Appendix E.4). Explicit singleton-absorption diagnostics evaluate once per generated position (Table 26). Their NFE and vanishing joint term do not describe the runs with cosine absorption schedules. Appendix C.2 gives the complete update order and matched-policy controls.

Absorption-policy baselines. Top K-margin selects the b_k active positions with largest $p_{(1)} - p_{(2)}$, where $p_{(1)} \geq p_{(2)}$ are the two largest posterior probabilities, and uses the same argmax absorption rule as LEDFLOW. Info-Gain uses a lookahead window of eight, the best-performing setting in Figure 4(c), with its additional evaluation cost reported in Appendix E.7.

Understanding. We evaluate with VLMEvalKit [8] on the full standard splits, following the established protocol [40, 36]. The base model is frozen FUDOKI throughout and only the decode-time sampler changes. The non-absorbing samplers, guidance, Top K-margin, and LEDFLOW all use one denoiser evaluation per grid point. Info-Gain additionally spends lookahead evaluations (Appendix E.7). Split sizes, metrics, and the 95% intervals appear in Table 10. The reported Euler MMBench score uses dev-EN, whereas our evaluation uses the full standard split.

Generation. GenEval [11] is evaluated on the full suite of 553 prompts with 4 images per prompt, scored with the standard Mask2Former and CLIP pipeline, with the standard K=64 sampling steps (denoiser evaluations). FUDOKI's native default is 128 and scores saturate by $K\sim$32, consistent

with prior reports [36]. We use one fixed LEDFLOW configuration for the full suite and all reported category scores.

Reasoning. All Sudoku results at a given model capacity use the same checkpoint. In Table 3, both non-absorbing and absorbing rows use the shared task-specific 6.4M Sudoku denoiser (Appendix C.1) at $K = 64$ with coordinatewise kinetic-optimal updates. Non-absorbing rows keep every generated position active throughout. Absorbing rows apply the same update at active positions and use a cosine absorption schedule m_k to determine how many positions to absorb. These absorbing policies differ only in their ordering score. All rows are evaluated on the same puzzles and seeds (protocol M in Table 9). The capacity diagnostics use a 6.4M denoiser of width 256 with 8 layers and 8 heads, trained for 60k steps. Their capacity-sensitivity comparison against a 21.3M denoiser, width 384 with 12 layers and 12 heads trained for 120k steps on a mixed easy-to-hard corpus, is reported separately in §E.5.3. This is a robustness check rather than a controlled scaling study, because capacity, number of training steps, and training distribution change together. The three Sudoku sets are held-out synthetic puzzles, Nikoli [28], and Sudoku-Extreme [38]. The additional constraint structures are 9×9 Latin squares, 3-coloring of random graphs, and molecular infilling on MOSES SMILES [26]. The Sudoku denoisers are trained on the Kaggle million-puzzle set after deduplication by puzzle string and solution grid against the evaluation sets.

Reasoning evaluation details. All Sudoku rows of Table 3, absorbing and non-absorbing, follow the matched protocol of Section 4: the same checkpoint, the same 500 generated, 100 Nikoli, and 700 Sudoku-Extreme puzzles, five seeds, and $K{=}64$. Figure 2 traces this cohort's Nikoli subset and reports its solve accuracies. The capacity and ordering-ladder diagnostics retain their own $n{=}100$, five-seed protocol. Zero seed variance under deterministic decoding does not imply zero puzzle-sampling uncertainty. Table 20 reports puzzle-level confidence intervals and paired tests on a separate $n{=}1000$ cohort. Its generated subset is not the generated set in Table 3, so their absolute accuracies should not be used interchangeably.

Generality. For the additional constraint structures, we train a fresh 4.8M denoiser for 9×9 Latin squares and a 1.2M graph-attention discrete flow denoiser for random 3-coloring. Molecular infilling uses a character-level model on MOSES SMILES and $n{=}2000$ examples. The pretrained-model experiments change only the native sampler while leaving model weights fixed. On GSM8K, LLaDA-8B-Instruct [22] uses $n{=}200$ and Dream-7B [47] uses $n{=}1000$, both with 128 sampling steps, generation length 256, and argmax token choice. LLaDA additionally uses block length 32. MMaDA-8B [45] uses $n{=}500$, 256 steps, and chain-of-thought prompting. BBH logical deduction pools $n{=}750$ three-, five-, and seven-object puzzles. For open multimodal mathematics, LLaDA-V uses seed-averaged default-order baselines on MathVerse ($n{=}900 \times 4$) and MathVista ($n{=}1000 \times 2$), whereas MMaDA uses a single default-order run on $n{=}900$ and $n{=}1000$, respectively.

E ADDITIONAL EXPERIMENTAL ANALYSIS

This appendix provides the additional analyses and full result tables for §4, in the same order as the main experiments.

E.1 UNDERSTANDING

Uncertainty of the main-text gains. The gain that clears its reported approximate 95% interval is +1.6 on POPE (±1.3). The remaining margins, including +2.3 on MM-Vet and +1.2 on MMMU, are smaller than their intervals and we do not read them as individually significant. Table 10 reports the intervals and their approximation limits.

E.2 GENERATION

E.2.1 TEXT-TO-IMAGE GENERATION

We use this setting as a control on soft structure, where the absorption policy should matter less than on tightly constrained outputs. We evaluate the full GenEval suite (553 prompts, four images per prompt, Mask2Former and CLIP scoring) on frozen FUDOKI with the standard $K{=}64$ sampling

Table 10: Understanding suite on the full standard splits via FUDOKI's VLMEvalKit. The table includes the six benchmarks of Table 1. The last column is an approximate 95% interval on the *native-sampler-vs*-LEDFLOW *difference* ($1.96\sqrt{2p(1-p)/N}$, evaluated at the native-sampler rate). Entropy-guided absorption improves every benchmark in the main-text suite, and the POPE margin clears the reported interval. The remaining margins are smaller than their intervals. This places short-answer understanding at the low-sensitivity end of the observed task spectrum, consistent with a smaller conditional term than on the constrained tasks in §4.1.

benchmark	N	native sampler	LEDFLOW (ours)	metric	spread	95% CI (diff)
POPE	5127	87.4	89.0	acc	+1.6	±1.3
MME	2374	1494.5	1504.6	P+R	+10.1	—
MMBench	4377	73.7	74.6	Overall	+0.9	±1.8
GQA	12578	57.0	58.0	acc	+1.0	±1.2
MM-Vet	218	38.2	40.5	GPT-judge	+2.3	±9.1
MMMU	1050	36.2	37.4	acc	+1.2	±4.1

The MME row is a summed perception-plus-reasoning score rather than an accuracy, so the binomial interval does not apply. These intervals do not account for paired predictions. The binomial approximation is also not a calibrated interval for graded GPT-judge scores such as MM-Vet. Paired example-level resampling is needed for inferential comparisons.

Table 11: Understanding suite in context: reported unified understanding-and-generation models alongside our frozen-FUDOKI decode-time samplers (condensed in Table 1 of the main text). Numbers are compiled from prior work [35]. Understanding is a measurement-validity control, not a SOTA claim. Higher is better, "—" not reported.

model	POPE	MME[†]	MMBench[‡]	GQA	MMMU	MM-Vet[§]
Reported unified understanding-and-generation models (context)						
LWM	75.2	—	—	44.8	—	9.6
Chameleon	—	—	—	—	22.4	8.3
Show-o-256	73.8	948.4	—	48.7	25.1	—
Show-o-512	80.0	1097.2	—	58.0	26.7	—
D-DiT	84.0	1124.7	—	59.2	—	—
VILA-U	85.8	1401.8	—	60.8	—	33.5
ILLUME	88.5	1445.3	65.1	—	38.2	37.0
TokenFlow-XL	86.8	1545.9	68.9	62.7	38.7	40.7
Janus	87.0	1338.0	69.4	59.1	30.5	34.3
Janus-Pro-1B	86.2	1444.0	**75.5**	59.3	36.3	39.8
FUDOKI (reported) [40]	86.1	1485.4	73.9	57.6	34.3	38.0
FUDOKI + guidance [35]	86.8	1492.7	74.2	58.2	35.4	38.6
Frozen FUDOKI, decode-time samplers (this setting)						
native sampler (FUDOKI)	87.38	1494.5	73.7	57.0	36.2	38.2
LEDFLOW (ours)	89.04	1504.6	74.6	58.0	37.4	40.5

[†]our MME is MME-P (perception only), the metric the reported models use. [‡]our MMBench is the full standard split, harder than the dev-EN split behind the published FUDOKI 73.9. [§]MM-Vet uses a GPT judge, and absolute values shift with the judge version, so we re-grade with the standard 0 to 1 partial-credit protocol (VLMEvalKit), which gives scores in the reported range.

steps. LEDFLOW reaches 0.781 Overall against the native sampler's 0.753. Its improvements span the category scores, including color attribute (0.723 vs. 0.663) and position (0.715 vs. 0.680).

Paired bootstrap intervals on GenEval. We resample the 553 prompts with replacement, stratified within each tag and keeping a prompt's four images together, with 10,000 replicates. The Overall gain of LEDFLOW over the native sampler is +0.0282 with a 95% interval of $[+0.009, +0.048]$ and $p = 0.003$, so the interval excludes zero. The per-tag differences are color attribute +0.060 $[+0.005, +0.117]$, position +0.035 $[-0.018, +0.090]$, colors +0.029 $[-0.003, +0.061]$, sin-

Table 12: GenEval in context: generation-only and unified understanding-and-generation models alongside the frozen-FUDOKI decode-time samplers of Table 2 (main text). The external rows modify or replace the model and are orthogonal to the decode-time comparison. External rows as compiled by their cited source. Higher is better, "—" not reported.

method	Single	Two	Count	Colors	Pos.	Attr.	Overall↑
Generation-only models (context)							
SDXL [34]	0.98	0.74	0.39	0.85	0.15	0.23	0.55
DALL-E 3 [34]	0.96	0.87	0.47	0.83	0.43	0.45	0.67
SD3-Medium [35]	0.99	0.94	0.72	0.89	0.33	0.60	0.74
FLUX.1-dev [34]	0.98	0.93	0.75	0.93	0.68	0.65	0.82
Unified understanding-and-generation models (context)							
Chameleon [35]	—	—	—	—	—	—	0.39
UniDisc [35]	0.92	0.47	0.15	0.67	0.13	0.19	0.42
LWM [35]	0.93	0.41	0.46	0.79	0.09	0.15	0.47
SEED-X [35]	0.97	0.58	0.26	0.80	0.19	0.14	0.49
Emu3-Gen [35]	0.98	0.71	0.34	0.81	0.17	0.21	0.54
Show-o [35]	0.95	0.52	0.49	0.82	0.11	0.28	0.53
+ Mask-GRPO [34]	0.99	0.90	0.69	0.85	0.35	0.59	0.73
TokenFlow-XL [35]	0.95	0.60	0.41	0.81	0.16	0.24	0.55
ILLUME [35]	0.99	0.86	0.45	0.71	0.39	0.28	0.61
Janus [35]	0.97	0.68	0.30	0.84	0.46	0.42	0.61
D-DiT [35]	0.97	0.80	0.54	0.76	0.32	0.50	0.65
Janus-Pro-1B [35]	0.98	0.82	0.51	0.89	0.65	0.56	0.73
Janus-Pro-7B [34]	0.99	0.89	0.59	0.90	0.79	0.66	0.80
MMaDA [34]	0.96	0.60	0.45	0.81	0.14	0.25	0.56
Frozen FUDOKI with decode-time samplers (Table 2)							
Euler@64 (base FUDOKI) [36]	0.9625	0.8384	0.4875	0.8833	0.7200	0.6300	0.7536
LEDFLOW (ours)	0.9563	0.8712	0.5219	0.9016	0.7150	0.7225	**0.7814**

gle object +0.019 $[-0.016, +0.056]$, counting +0.019 $[-0.037, +0.078]$, and two objects +0.008 $[-0.028, +0.043]$. Only color attribute is individually significant, so the Overall gain reflects a consistent direction across tags rather than one decisive category. This gain is small relative to those on constrained reasoning tasks, consistent with text-to-image prompts imposing weaker structure. The comparison with exact guidance (0.7800) cannot be tested this way, because that value is a published number [35] rather than a per-prompt run of ours, so we report it as a numerical comparison only.

GenEval in the masked discrete flow literature. Placing LEDFLOW against the recent masked discrete flow literature on the same FUDOKI base (Table 13) separates two orthogonal axes. Among *frozen-model samplers* (our setting) the whole family sits within a band of about 0.05: LEDFLOW (0.781) is the strongest entry, ahead of the native sampler (0.753) and of the corrected-sampler family (Euler 0.754, time-/location-corrected 0.76 to 0.77 [36]), though the margin over the corrected samplers is comparable to the benchmark's own resolution. The higher scores in the literature come from the orthogonal *training* axis (guidance/RLHF 0.78 [35], rate-aware policy optimization 0.93 [34]), which modify the model rather than the sampler and are complementary to our decode-time contribution. Table 12 places the frozen-sampler comparison against generation-only and unified models.

E.2.2 Comparison with related generation methods

E.3 Trajectory analysis under matched settings

Figure 2 compares native uniform flow with LEDFLOW on 100 Nikoli puzzles at $K=64$: puzzle solve accuracy rises from 0.410 to 0.845, while correct-to-wrong reversions fall from 0.094 to 0.026 of generated cells. These are the motivation measurements used in the main text. On text-to-image generation the two directions must be read on the counterfactual prompts. On in-distribution GenEval prompts the native flow recovers more than it destroys late in sampling (0.156 wrong-to-correct

Table 13: GenEval Overall on FUDOKI in the masked discrete flow literature, separating decode-time *samplers* (frozen model, our setting) from *training*-based methods. Among frozen-model samplers the spread is ≈ 0.05, and an external FUDOKI Euler@64 result reproduces our native sampler to 0.001. LEDFLOW is the strongest decode-time entry in this comparison, although its margin over the corrected-sampler family is within the benchmark's resolution. Training-based methods modify the model and address a separate axis.

method	GenEval Overall
Frozen-model samplers (this setting)	
native sampler (Euler@64, ours)	0.753
Euler@64 [36]	0.754
time-/location-corrected [36]	0.765
LEDFLOW (ours)	**0.781**
Training-based (context)	
FUDOKI base (+CFG) [35]	0.77
+ guidance / RLHF [35]	0.78
+ dFlowGRPO [34]	0.93

Table 14: **Text-to-image trajectory measurements.** FUDOKI text-to-image (GenEval prompts, $K{=}64$). Token statistics are per image token, with 95% bootstrap intervals over images for the difference, and correctness flips are the fraction of samples whose GenEval verdict changes between $t{=}0.75$ and the endpoint, on in-distribution GenEval prompts and on the 16 counterfactual out-of-distribution prompts of Table 15. The Sudoku counterpart of these measurements is Figure 2(c). ⋆GenEval prompts are in-distribution for the pretrained backbone, so late revisions there can recover memorized configurations. The counterfactual prompts isolate the same measurement without that confound, and under them correct-to-wrong exceeds wrong-to-correct.

Measurement	Euler	LEDFLOW	Diff. [95% CI]
Token changes per position	4.14	2.33	$+1.81$ $[1.72, 1.88]$
excluding the final step	3.44	2.30	$+1.13$ $[1.04, 1.21]$
Positions changed more than once	0.97	0.73	$+0.24$ $[0.22, 0.25]$
Changes occurring after $t{=}0.75$	0.45	0.11	$+0.34$ $[0.34, 0.35]$
Mean time of last change	0.96	0.54	$+0.42$ $[0.41, 0.43]$
GenEval accuracy	0.753	0.781	-0.028 $[-0.037, -0.020]$
In-dist. correct to wrong⋆	0.052	0.010	$+0.042$
In-dist. wrong to correct⋆	0.156	0.000	$+0.156$
OOD correct to wrong	0.125	0.025	$+0.100$
OOD wrong to correct	0.037	0.013	$+0.024$

against 0.052 correct-to-wrong), which is what memorized configurations predict when the prompt distribution is the one the backbone was trained on. On the counterfactual prompts of Table 15, where that route is unavailable, the ordering reverses and correct-to-wrong dominates (0.125 against 0.037). Table 14 and Figure 7 report the text-to-image trajectory analysis.

Ordering diagnostic. Table 16 compares absorption policies under the uniform discrete flow. All four arms use the same denoiser, puzzles, grid ($K{=}64$), and argmax value rule, and the three absorbing arms share one code path. The first two use argmax values and differ in selection rule, while the third samples the absorbed value. Arbitrary-order absorption raises cell accuracy from 0.413 to 0.723 (paired puzzle-level 95% interval $[0.285, 0.336]$) and removes every correct-to-wrong flip. Ordering raises it further to 0.935 (interval $[0.166, 0.256]$) and raises solve accuracy from 0.410 under the arbitrary order to 0.845. The mechanism is visible in the bad-absorption column, the fraction of absorption events that write an incorrect value, which falls from 28.0% under the arbitrary order to 6.6% under the entropy order. Absorbing a sampled value instead of the argmax changes

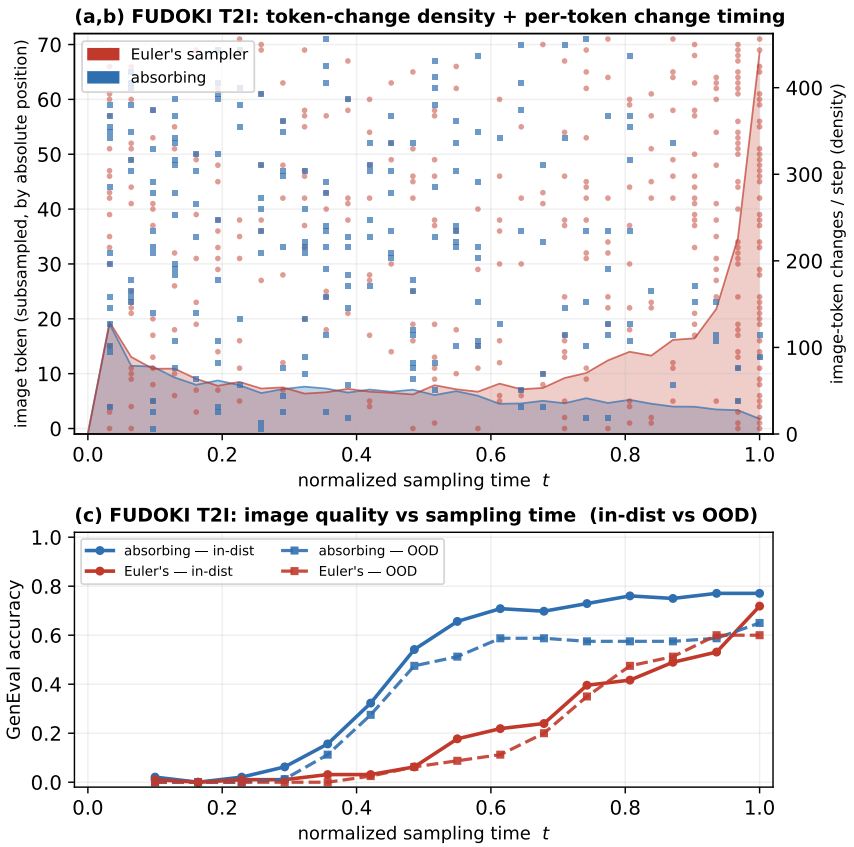


Figure 7: **FUDOKI text-to-image trajectories, native uniform flow versus LEDFLOW.** (a,b) Image-token change density over sampling time: the native flow accumulates most of its changes after t=0.75, while LEDFLOW front-loads absorption and its density decays. (c) GenEval accuracy of the image decoded from the intermediate state, on in-distribution GenEval prompts (solid) and on the 16 counterfactual prompts of Table 15 (dashed). LEDFLOW reaches its final quality by $t\approx$0.6 and holds it. The native flow reaches comparable quality only at the endpoint, after the late burst of changes in (a,b).

Table 15: **Out-of-distribution prompts for the text-to-image trajectory analysis.** Sixteen counterfactual prompts, four per GenEval tag, written so that the requested attribute, count, pairing, or spatial relation is unlikely under the training distribution. They are scored with the standard GenEval detector pipeline and are the dashed curves in Figure 7 and the OOD rows of Table 14.

Tag	Prompts
Colors	a photo of a green cow; a photo of a purple giraffe; a photo of a blue banana; a photo of a pink elephant
Counting	a photo of eight apples; a photo of six cats; a photo of seven birds; a photo of five clocks
Two objects	a photo of a toothbrush and a giraffe; a photo of a pizza and a fire hydrant; a photo of a laptop and a cow; a photo of an elephant and a spoon
Position	a photo of a car above a bird; a photo of a boat above a bed; a photo of a bicycle above an elephant; a photo of a cup above a train

little (0.923 against 0.935 cell accuracy, 0.821 against 0.845 solve), so the gain comes from freezing plus ordering rather than from the value rule.

Table 16: **Absorption-policy diagnostic on Nikoli** (n=100, 6.4M denoiser, K=64, seeds 0 to 4). Bad-absorption is the fraction of absorption events whose absorbed value is wrong. The arbitrary-order and entropy-order solve accuracies are the values of Table 3. The no-absorption arm never fixes a position, so no puzzle completes. Paired puzzle-level bootstrap intervals for the cell-accuracy differences: no absorption to arbitrary +0.310 $[0.285, 0.336]$. Arbitrary to entropy +0.212 $[0.166, 0.256]$.

Arm	Cell acc.	Solve	Correct→wrong	Changes/cell	Late-change frac.	Bad-absorption
Uniform flow (no absorption)	0.413	0.000	0.281	1.87	0.60	—
+ arbitrary absorption order	0.723	0.410	0.000	1.00	0.37	0.280
+ entropy order (LEDFLOW)	**0.935**	**0.845**	0.000	1.00	0.37	**0.066**
+ entropy order, sampled absorption	0.923	0.821	0.000	1.00	0.37	0.078

Table 17: **Sudoku solve accuracy by sampler.** These values and policy labels reproduce Table 3. All rows share the same checkpoint, puzzles, seeds, and K=64 (protocol M). The identical Euler and Time-corrected rows are a discretization effect, explained in Appendix E.5.4.

Sampler	Generated	Nikoli	Sudoku-Extreme
Euler	0.610	0.410	0.078
FreeCorrection	0.630	0.400	0.126
Time-corrected	0.610	0.410	0.078
Location-corrected	0.588	0.360	0.096
Top K-margin	0.820	0.796	0.271
Info-Gain	0.804	0.780	0.231
LEDFLOW	0.865	0.845	0.269

E.4 ABSORPTION-ORDER COMPARISON AND SELECTED-POSITION ANALYSIS

Table 17 reports the current Sudoku solve accuracies and policy labels from Table 3. All methods at this capacity use the shared task-specific 6.4M Sudoku checkpoint. The non-absorbing methods use kinetic-optimal updates. Absorbing methods additionally use a cosine absorption schedule. All rows are evaluated on the same puzzles and seeds (protocol M in Appendix D). The Euler and Time-corrected rows coincide on every puzzle column. Appendix E.5.4 explains why the two integrators agree to first order on these tasks and quantifies the residual. The ordering diagnostic in Table 16 and the earlier trajectory diagnostics below are separate experiments.

What the current comparison shows. On generated and Nikoli puzzles, entropy absorption attains higher solve accuracy than the other displayed policies. On Sudoku-Extreme, margin is numerically higher (0.271 versus 0.269). These are descriptive comparisons on a matched cohort of 100 puzzles per set. The paired significance tests for the ordering policies are reported at n=1000 in Table 20.

Statistical evidence. On seed-0 per-puzzle arrays, exact McNemar tests favor entropy over the native sampler and over arbitrary absorption on all three sets ($p \leq 3 \times 10^{-8}$). This diagnostic does not separate entropy from the other confidence scores. The paired comparison against probability margin and maximum probability is the n=1000 test in Table 20, where the three are statistically tied and entropy separates from Info-Gain (-0.066, p=3×10^{-9}).

Decoder and schedule. Both non-absorbing and absorbing baselines use coordinatewise kinetic-optimal updates. Here a singleton flow transition changes one coordinate. It does not specify the number of positions absorbed at a grid point. The non-absorbing baseline keeps all generated positions active. Absorbing policies apply the same kinetic-optimal update to positions that remain active, while a cosine absorption schedule over $K = 64$ grid steps determines the size of each irreversible absorption block. A step can absorb zero, one, or several positions, so this protocol is not the $b_k = 1$ absorption ablation and need not have zero joint dependence error. For policies without additional lookahead, the grid protocol uses one denoiser evaluation per step, giving $\mathrm{NFE} = K$. Figure 2 reports separate trajectory and endpoint runs under kinetic-optimal updates.

Selected-position reliability. As a reliability diagnostic rather than a validation of a specific theoretical envelope, Table 18 bins the entropy gate's absorbed positions by entropy at absorption and

Table 18: **Absorption error at absorbed positions** (LEDFLOW gate, seed 0, flow-active decode): empirical error rate binned by entropy at absorption, with the number of absorbed positions per bin. Low entropy predicts low error on generated/Nikoli. The relation fails under hard-puzzle over-confidence (Sudoku-Extreme). "—" marks empty bins.

	Generated		Nikoli		Sudoku-Extreme	
Entropy at absorption	err	n	err	n	err	n
$[0.00, 0.05)$	0.059	26496	0.086	5540	0.448	36240
$[0.05, 0.10)$	0.700	20	0.167	6	0.550	866
$[0.10, 0.20)$	0.750	16	0.333	6	0.570	732
$[0.20, 0.40)$	0.750	16	0.778	9	0.537	555
$[0.40, 0.80)$	0.600	35	0.800	10	0.622	585
$[0.80, 1.60)$	0.778	9	0.000	1	0.714	70
$[1.60, 10.0)$	1.000	1	—	0	1.000	6
Overall	0.061	26593	0.088	5572	0.457	39054

reports the empirical absorption-error rate per bin. We do not fit or test a KL envelope here. Under the flow-active decode almost all absorptions are made at very low entropy (99.6% / 99.4% of absorptions on generated/Nikoli), where the error rate is 5.9% / 8.6%. Error rises with entropy over the sparse higher-entropy bins. On Sudoku-Extreme the denoiser is over-confident: even the lowest-entropy absorptions are wrong 44.8% of the time. The first erroneous absorption is correspondingly rare and late on easy puzzles (generated: 14% of puzzles have any wrong absorption, median first-wrong time t=0.55, and Nikoli: 16%, t=0.31) but common and early on hard puzzles (Sudoku-Extreme: 75%, t=0.23). Low absorption error where the policy operates is consistent with, but does not by itself establish, the conditional guarantee's envelope assumption. The hard-puzzle regime shows confidently wrong absorption that any such assumption must exclude.

E.5 REASONING

Two qualifications apply to the main-text reasoning table. Info-Gain leads on molecular infilling, so local entropy is not uniformly preferable, and the current results do not isolate why that ordering reverses. Average ranks are descriptive: small mathematics differences, including near-floor GSM8K accuracy, do not establish meaningful wins.

E.5.1 SUDOKU AS A TRANSPARENT ORACLE

Sudoku provides a transparent oracle because quantities that are usually inaccessible can be computed directly, in particular the terms in the absorption-policy analysis (Theorem 1). The joint term is the exception: on a unique-solution puzzle the oracle conditional is a point mass, so the conditional total correlation of Lemma 2 is identically zero for every block, and unique-solution Sudoku cannot be used to diagnose the joint term. For that term we use the multi-solution variant described below and the graph-coloring task in Appendix E.6.1. Its vocabulary and sequence length are bounded by $|\mathcal{S}| = 9$ digits and $M \leq 81$ cells, much smaller than the subword vocabulary and open-ended length of free-text generation [31]. More importantly, a standard Sudoku puzzle has a unique solution [32], so $q(a_1 \mid \mathcal{C}) = \mathbf{1}[a_1 = a_1^\star]$ and $H(q) = 0$. The oracle denoiser is therefore invariant to the absorption priority by Section 3.3. Any priority effect exhibited by a learned denoiser arises from approximation error and is captured by the conditional term $\mathcal{E}_{\text{cond}}$ in Proposition 2. For every reachable partial fill a_S with at least one valid completion, the true conditional distribution is computable from completion counts:

$$q(a_1^j = v \mid a_S) = \frac{N(a_S, j, v)}{\sum_{v'} N(a_S, j, v')}, \qquad H_q(a_1^j \mid a_S) = -\sum_{v=1}^{9} q(a_1^j = v \mid a_S) \log q(a_1^j = v \mid a_S),$$

where $N(a_S, j, v)$ counts valid completions of a_S with cell j fixed to v. Constraint propagation and backtracking compute these counts in milliseconds for standard 9×9 puzzles. When no valid completion exists, the conditional is undefined, so oracle diagnostics use only consistent prefixes. On the unique-solution RRN [24], Sudoku-Extreme [38], and million-puzzle Kaggle corpora, the counts

Table 19: Structured-task results for uniformization and fixed-score absorption. Higher is better. "—" denotes the SMILES setting without a static relevance score.

Sampler	Generated	Nikoli	Extreme	Latin	Graph	Molec.
Uniformization	0.562	0.240	0.074	0.694	1.000	0.831
Fixed-score absorption	0.702	0.420	0.158	0.946	1.000	—

satisfy $N \in \{0, 1\}$ and H_q vanishes along every consistent path. The multi-solution variant [20] provides nondegenerate oracle entropy when needed.

Structured-task-specific baselines. Table 19 retains two controls omitted from the cross-task main table. Uniformization is evaluated here as an exact CTMC control on the puzzle tasks. We did not adapt it to the mathematics harness. The fixed-score absorption policy similarly ranks positions by a static task score before selective absorption, so it does not directly apply when absorption is not used and positions can be rewritten. We also did not define a corresponding static relevance score for character-level SMILES infilling.

E.5.2 Reasoning controls

Two controls separate the absorption policy from the other choices a sampler makes. Table 20 repeats the Sudoku comparison at n=1000 with puzzle-level bootstrap intervals and paired McNemar tests, so that the ordering gap is read against sampling uncertainty rather than seed variance, and stratifies it by difficulty. Its lower block adds the competing ordering policies on the same puzzles: the probability-margin and maximum-probability policies are not significantly different from entropy in these paired tests (p=0.23 and 0.80). This does not establish statistical equivalence, and the scores need not induce the same ordering of positions. Info-Gain trails by 0.066 (p=3×10^{-9}) at roughly six times the wall-clock on this task. On the stronger 21.3M denoiser the gap narrows to 0.017 but remains significant (p=0.027), consistent with the crossover analysis of Appendix C.3.5: lookahead gains on the entropy policy as the denoiser improves, but has not overtaken it end-to-end at this capacity. The separation that matters is therefore between confidence-ordered single-position absorption and either lookahead ordering or no ordering at all, which is the comparison Lemma 1 and Figure 4(c) speak to. Table 21 then crosses the absorption priority with the value-selection rule: holding value selection fixed and varying only the priority moves accuracy far more than holding the priority fixed and switching between argmax and sampled values. This attributes the gain to position selection rather than to the rule used to assign an absorbed value. Table 22 extends the same manipulation to arbitrary and high-entropy-first policies.

E.5.3 Sensitivity to Sudoku denoiser capacity

Table 22 checks whether the ordering result persists under the stronger Sudoku denoiser described in Appendix D. Because capacity, number of training steps, and training distribution change together, this is a robustness sensitivity rather than a controlled parameter-scaling study. These ordering diagnostics use the shared checkpoint at each capacity under the uniform discrete flow. Within this diagnostic, entropy-guided absorption is the strongest sampler on both Nikoli and Sudoku-Extreme under both training regimes.

The absorption-policy ladder. The middle block of Table 22 directly tests the priority policy: the denoiser, argmax value rule, absorption count, and number of sampling steps are held fixed. Selecting positions by increasing predictive entropy, in an arbitrary order, and by decreasing predictive entropy produces a monotone ladder at both capacities: $0.900/0.340/0.110$ on Nikoli at 6.4M and $0.970/0.820/0.550$ at 21.3M. Three features of the ladder are worth noting. The reversal is the worst configuration in the table, below uniformization and every corrected sampler, which separates the absorption-priority effect from the discretization effect. Process-level samplers occupy a narrow range, whereas changing only the order spans 0.79 at 6.4M. Arbitrary order is close to Euler (0.340 vs. 0.410 on Nikoli), further indicating that absorption priority drives the difference. The ladder compresses as the denoiser improves, from a span of 0.79 at 6.4M to 0.42 at 21.3M, which is qualitatively consistent with the oracle limit in Theorem 1(ii). Because the models also differ in

Table 20: Sudoku at scale (n=1000 generated puzzles, 6.4M, #givens $\in [22, 34]$): solve accuracy with 95% puzzle-level bootstrap CIs and paired McNemar tests. Upper block: paired against Euler. The adaptive and fixed-score absorption policies are both significant, and logical forcedness underperforms the model's entropy policy (an imperfect denoiser tracks calibration, not forcedness). Lower block: the competing single-position ordering policies on the same 1000 puzzles, paired against the entropy policy. The three confidence-based scores (entropy, probability margin, maximum probability) are statistically tied, whereas Info-Gain (window 8) is significantly below them and unordered Euler far below. Wall-clock per puzzle on one H100 is 6.9 ms (entropy), 6.2 (margin), 2.4 (max-probability), 54.5 (Info-Gain), and 8.6 (Euler, re-run at 0.427, within the interval of the upper-block value). On the 21.3M denoiser the entropy policy remains ahead of Info-Gain (0.964 against 0.947, p=0.027) at 9.5 against 149.3 ms. Right block: accuracy stratified by difficulty, where the ordering gap grows as puzzles harden. We note that this is not Theorem 1(ii) directly, which concerns the limit in which the denoiser approaches the oracle. The two agree only under the additional premise that harder puzzles leave the denoiser further from the oracle, which is plausible here but is not established by the theorem.

absorption policy	solve-acc	95% CI	vs Euler (McNemar)	by #givens (determined) 22–25	26–29	30–34
entropy guided (adaptive)	**0.922**	[.905, .938]	$+.486$, $p<10^{-12}$	0.837	0.949	0.967
logical-forcedness (MRV)	0.621	[.591, .651]	$+.185$, $p<10^{-12}$	—	—	—
fixed external score	0.527	[.496, .558]	$+.091$, $p=9\times10^{-7}$	0.288	0.493	0.775
arbitrary-order absorption	0.446	[.415, .477]	—	—	—	—
Euler@64	0.436	[.406, .467]	—	0.190	0.358	0.740
			vs entropy			
probability margin	0.915	[.897, .932]	$-.007$, p=0.23	0.824	0.935	0.973
maximum probability	0.920	[.903, .936]	$-.002$, p=0.80	0.834	0.943	0.970
Info-Gain (window 8)	0.856	[.834, .877]	$-.066$, $p=3\times10^{-9}$	0.732	0.868	0.952
21.3M denoiser, same puzzles						
entropy guided (adaptive)	**0.964**	[.952, .975]	—	0.929	0.970	0.988
Info-Gain (window 8)	0.947	[.933, .961]	$-.017$, p=0.027	0.892	0.957	0.985

Table 21: Decision-rule control (Nikoli, 6.4M, n=100, 5 seeds): absorption *priority* crossed with value *selection*, all single-position with per-step reevaluation. Priority dominates at both absorption rules. The argmax-vs-sample rule is a minor effect. The unordered + argmax cell is the arbitrary-order policy follows the uniform first-hitting choice of [51]. The entropy-guided and Euler entries are the Table 3 values for this cohort. Puzzle-level CIs and paired McNemar tests are in Table 20.

absorption priority	argmax value	sampled value
entropy guided	$\mathbf{0.845}$	0.874 ± 0.008
arbitrary order (uniform choice, [51])	0.380 ± 0.031	0.280 ± 0.026
Euler@64 (non-absorbing flow, argmax): 0.410		

training steps and data mixture, this comparison does not attribute the compression to parameter count alone. Table 23 places the fixed-score and arbitrary-order absorption baselines on the same continuum and extends the comparison to the generated set.

E.5.4 Why Euler and Time-corrected nearly coincide on the puzzle tasks

Table 3 reports the same accuracy for Euler and the time-corrected sampler on all six puzzle columns, while the three mathematics columns differ. Both rows run the same uniform discrete flow as Section 2, with no selective absorption. The near-coincidence follows from the discretization rather than from a reporting artifact.

The two integrators differ only in how they discretize the schedule hazard within a grid interval. On the grid $t_{k-1} < t_k$ with $\Delta t = t_k - t_{k-1}$, the per-step transition probability of a position whose

Table 22: **Sensitivity to Sudoku denoiser capacity** (solve accuracy). Parameter counts are exact up to one decimal place. The 6.4M entries of the LEDFLOW and Euler rows are the values of Table 3. The models also differ in training steps and data mixture, so the comparison demonstrates robustness across denoisers rather than an isolated size effect. The middle block isolates absorption priority by holding the denoiser and argmax value rule fixed. Accuracy decreases monotonically from entropy-guided to arbitrary and then high-entropy-first absorption.

	Nikoli↑		Sudoku-Extreme↑	
Sampler	6.4M	21.3M	6.4M	21.3M
Uniformization	0.240	0.630	0.074	0.194
Euler	0.410	0.820	0.078	0.224
Fixed external score	*0.420*	*0.870*	*0.158*	*0.312*
FreeCorrection	0.400	0.780	0.126	0.270
Time-corrected	0.410	0.820	0.078	0.227
Location-corrected	0.360	0.790	0.096	0.246
Absorption-policy ablation (same denoiser and value rule, priority varied alone)				
arbitrary absorption	0.340	0.820	0.104	0.256
high-entropy-first absorption	0.110	0.550	0.051	0.117
LEDFLOW (ours)	**0.845**	**0.970**	**0.269**	**0.514**

Table 23: **The absorption-order ladder.** Solve accuracy for adaptive low-entropy, offline-score, arbitrary, and high-entropy orders. Absorbing policies share the denoiser and argmax rule within each capacity; the non-absorbing Euler row is a separate reference. These results test position ordering under the uniform discrete flow. The larger spread for the weaker denoiser is an empirical observation, not a prediction of Theorem 1 without further assumptions.

Absorption order	Nikoli↑		Sudoku-Extreme↑		Generated↑	
	6.4M	21.3M	6.4M	21.3M	6.4M	21.3M
Low-entropy-first (ours)	**0.845**	**0.970**	**0.269**	**0.514**	**0.865**	**0.996**
Offline σ_{rel} (deployable)	0.420	0.870	0.143	0.301	0.840	0.960
Euler @64 (flow)	0.410	0.820	0.078	0.224	0.610	0.924
Arbitrary single-position	0.340	0.820	0.104	0.256	0.756	0.948
High-entropy-first (reversal)	0.110	0.550	0.051	0.117	0.580	0.860

Within each capacity, the absorbing policies share the denoiser and per-cell argmax rule, and unabsorbed positions continue under the uniform-flow update. The checkpoint is shared with the corresponding capacity in the main experiments, and the 6.4M entries for the low-entropy and Euler rows are the values of Table 3. The Euler row is a reference baseline, not an absorption-order variant.

current token differs from its predicted target is

$$p_k^{\mathrm{euler}} = 1 - \exp\Big(-\frac{\Delta t}{1-t_{k-1}}\Big), \qquad p_k^{\mathrm{tc}} = 1 - \frac{1-t_k}{1-t_{k-1}} = \frac{\Delta t}{1-t_{k-1}},$$

under the linear time parameterization used on the puzzle tasks. Both expressions discretize the same hazard $\Delta t/(1-t_{k-1})$, one exponentially and one linearly. They agree to first order in Δt and differ at $O(\Delta t^2)$.

On the puzzle tasks the vocabulary contains nine digits and every puzzle has a unique solution, so once the denoiser's predicted target at a position stabilizes, both integrators move that position to the same digit and the $O(\Delta t^2)$ discrepancy changes only *when* within the grid the move occurs, not *which* digit is written. Under common random numbers the two samplers therefore produce the same final grid except when the discrepancy places a transition on opposite sides of a step boundary, which changes the context seen by the denoiser at the next step and can alter the solved grid on a small fraction of puzzles.

The measurements match this account. Across our sampler sweeps the two agree exactly on roughly half of the (benchmark, K) pairs and differ by 0.001 to 0.006 on the rest, with no systematic sign: on

Nikoli at $K{=}32$ and $K{=}64$ they are identical (0.730 and 0.820 at 21.3M), while on Sudoku-Extreme at 21.3M they separate slightly at every K (0.224 vs. 0.227 at $K{=}64$, $n{=}700$). The residual is therefore a discretization effect, an order of magnitude below the +0.29 that the absorption policy contributes on the same benchmark, and it is smaller than the reporting resolution of the Sudoku splits in Table 3.

In the mathematics columns (MathVista, MathVerse, GSM8K) the Euler and corrected rows run FUDOKI's native uniform flow over the full subword vocabulary with selective absorption disabled. The LEDFLOW, Top K-margin, and Info-Gain rows in the same columns apply the cosine absorption schedule of Appendix C.2 on that native text flow and differ only in the ranking score. With a large vocabulary and open-ended answers, the predicted target changes more often between steps, the two updates cease to be near-equivalent, and the gaps in Table 3 (0.254 vs. 0.241 on MathVista, 0.026 vs. 0.022 on GSM8K) are the genuine difference between the exponential and the linearized step. The contrast between the two regimes is itself evidence for the paper's claim: on constrained puzzles, discretization corrections move accuracy by at most a few tenths of a point, while the absorption policy moves it materially.

E.5.5 CAN AN EXTERNAL PRIORITY BEAT CONFIDENCE? TWO INDUCED-MISCALIBRATION PROBES

On Sudoku our adaptive entropy policy is determined by the model's internal confidence, so we ask whether a fixed external priority can ever beat it. This comparison is outside the local optimality claim of Theorem 1, but tests whether an external score can help when confidence is miscalibrated. We induce that regime in two ways (Table 8). **(a) Density shift** (train dense givens 44 to 56, test sparse 26 to 31): the model is verifiably overconfident (ECE rises from 0.008 to 0.074) but its confidence priority beats the external priority by a *wider* margin than on the calibrated model. **(b) Wrong constraint** (train box-blind Latin squares, test Sudoku): a sign flip appears *only* in the hardest band where the box-blind model is floored (a non-significant $\sim$2-puzzle margin). Wherever the model is capable, confidence wins. In these probes, absorption-policy quality follows the confidence *ranking* more closely than calibration or absolute correctness. Models capable of solving the task rank cells well enough for their internal priority to win, whereas models with poor rankings rarely solve the puzzle. Neither manipulation produces a capable but systematically misranked denoiser. The same pattern appears when logical forcedness trails confidence (Table 20). We therefore make no accuracy claim for an external priority over internal confidence.

E.5.6 TEMPERATURE SCALING OF THE ORDERING ENTROPY

A related question is whether the entropy used for ordering should first be calibrated. We rescale the logits by a temperature T before computing the entropy, $H_\theta^j\big(\mathrm{softmax}(\ell^j/T)\big)$, so that T changes only the *order* in which positions are absorbed. The absorbed value remains the argmax of the unscaled posterior. We report the change in solve accuracy relative to the default $T{=}1$ within a single sweep, so the comparison is internal to this ablation and the absolute values of Table 3 remain the reported ones.

Solve accuracy is essentially flat in T (Table 24). On generated and Nikoli puzzles the whole sweep moves the solve rate by at most 0.02 in either direction. On Sudoku-Extreme, sharpening helps marginally, +0.007 at $T{=}0.5$, but the effect is about five puzzles out of 700 and is non-monotone, since $T{=}2$ returns to +0.002. Temperature mostly preserves the relative ranking of positions, so a flat response is the expected outcome. We read it as evidence that the ordering is robust to this knob rather than as evidence that the denoiser is calibrated, which the probes above address directly.

E.6 ENTROPY-GUIDED ABSORPTION AND GENERALITY

The transfer comparison in Figure 4(b) replaces each model's default generation order with the LEDFLOW order through that model's own sampler, leaving its weights unchanged. Several transfer differences are not significant, and MMaDA MathVista changes by -0.001. This subsection complements Figure 4(a,b) with results across additional constraint structures and pretrained model families. The comparisons retain the same absorption count while varying the priority policy, isolating whether the low-entropy order and its transfer persist beyond the primary Sudoku setting.

Table 24: **Temperature scaling of the ordering entropy** (6.4M denoiser, K=64). T rescales the logits used for the ordering score only. The absorbed value is the argmax of the unscaled posterior. Entries are changes in solve accuracy relative to the default T=1 of Table 3, measured within one sweep, so no entry restates an absolute accuracy for a configuration reported elsewhere.

T	Generated	Nikoli	Sudoku-Extreme
0.5	+0.005	−0.010	**+0.007**
0.7	+0.005	−0.010	+0.005
1.0 (default)	0.000	0.000	0.000
1.5	**+0.010**	−0.020	−0.011
2.0	+0.005	−0.010	+0.002

Ordering-analysis cohort. Figure 4(a) reports puzzle solve accuracy on $n = 1600$ puzzles, pooled over the two denoiser capacities of §E.5.3 and over the 100 Nikoli and 700 Sudoku-Extreme puzzles, with seeds averaged per puzzle and sets weighted by their size. Because the pool mixes two capacities and two difficulty levels, its accuracies are lower than the per-set Nikoli values of Table 3 and should not be compared with them directly. Intervals are puzzle-level bootstrap percentiles. This pooled comparison orders policies from adaptive low entropy through arbitrary and non-absorbing Euler to high entropy.

E.6.1 Additional constraint structures: Latin squares, graph coloring, molecular validity

To check that the absorption-policy effect is a property of *constraint structure* and not of Sudoku specifically, we repeat it on 9×9 Latin squares. The structure-dependence reproduces cleanly: entropy-guided absorption reaches 0.998 against 0.752 arbitrary-order absorption and 0.690 non-absorbing Euler flow (+0.31), and the external logical-forcedness order again underperforms confidence (0.948 vs. 0.998). The decisive order benefit is therefore not Sudoku-specific.

Off the grid: graph coloring and molecular validity. On random-graph 3-coloring, the paper's *other* separated effect dominates: on hard instances the single-position absorption samplers all reach $\sim$1.0 while the non-absorbing Euler flow floors at 0.45 to 0.63 (a +0.37 to +0.55 total-correlation gap, Lemma 2), while the priority among single-position absorption policies saturates (the near-oracle regime of Theorem 1(ii)). On masked infilling of drug-like molecules from MOSES SMILES [26], entropy-guided absorption is the most valid and the high-entropy-first policy the least, with the gap over Euler *growing* with the masked fraction (at mask 0.7: $0.896 > 0.843 > 0.828 > 0.767$, +0.068 over Euler, $z\approx4$). Across these domains, the dominant effect is consistent with the corresponding theoretical term: the conditional term on unique-solution grids with an imperfect denoiser, and the joint term on flexible coloring.

E.6.2 Transfer to pretrained diffusion models

We vary only the absorption order through each model's native sampler, using the protocols in Appendix D. The baseline in every row is the model's own default generation order, so each comparison replaces that default with the LEDFlow order while leaving weights, prompting, and the rest of the decoding pipeline unchanged. For LLaDA-8B that default is low-confidence remasking, which ranks positions by maximum probability, so the comparison there is between two confidence scores rather than against an uninformative order. Absolute accuracies are specific to our protocol (sample size, number of sampling steps, prompting, and extraction in Appendix D) and are not comparable to numbers published with each model's own evaluation harness. For MMaDA-8B on GSM8K in particular, both arms sit below the published figure because of these protocol differences, so only the within-row difference is meaningful. The entropy-guided order improves over the default order by 0.045 on LLaDA, 0.121 on Dream, and 0.228 on MMaDA for GSM8K. On BBH it improves LLaDA and Dream by 0.063 and 0.176. For MMaDA, the effect is strongest on GSM8K, positive on Track, ARC-Challenge, and CommonsenseQA, and nearly unchanged on BBH.

For open multimodal mathematics, the differences on MathVista and MathVerse are not significant for either model, providing the generality boundary summarized in Figure 4(b). Table 25 reports

Table 25: **Full transfer results across pretrained diffusion-model families.** We vary only the absorption policy through each model's native sampler. The text-reasoning experiments use matched wall-clock time. Higher is better. The label n.s. denotes a difference that is not statistically significant. Every cell shown in Figure 4(b) appears here. The cells left grey in that figure are not applicable rather than pending: LLaDA-8B and Dream-7B are text-only diffusion language models and are not evaluated on the image-conditioned benchmarks, while LLaDA-V is an understanding-only multimodal model without a generation head and is therefore evaluated on multimodal mathematics alone.

Model	Benchmark	Backbone family	Default order	LEDFLOW order	Δ
Text reasoning (native sampler, order-only ablation)					
LLaDA-8B-Instruct [22]	GSM8K	masked diffusion LM	0.725	**0.770**	$+0.045$
Dream-7B [47]	GSM8K	masked diff. LM (uniform prior)	0.371	**0.492**	$+0.121$ (z=5.5, $p\approx5\times10^{-8}$)
MMaDA-8B [45]	GSM8K	unified MM diffusion	0.246	**0.474**	$+0.228$ ($p\approx2\times10^{-18}$)
MMaDA-8B	BBH logical deduction	unified MM diffusion	0.370	**0.373**	$+0.003$ (χ^2=0.00, n.s.)
MMaDA-8B	Tracking shuffled objects	unified MM diffusion	0.180	**0.240**	$+0.060$ (χ^2=3.11, $p\approx0.08$)
MMaDA-8B	ARC-Challenge	unified MM diffusion	0.383	**0.417**	$+0.033$ (χ^2=0.71, n.s.)
MMaDA-8B	CommonsenseQA	unified MM diffusion	0.367	**0.400**	$+0.033$ (χ^2=4.61, $p\approx0.03$)
LLaDA-8B-Instruct	BBH logical deduction	masked diffusion LM	0.583	**0.645**	$+0.063$ ($p\approx10^{-3}$)
LLaDA-8B-Instruct	Tracking shuffled objects	masked diffusion LM	0.476	**0.592**	$+0.116$
LLaDA-8B-Instruct	ARC-Challenge	masked diffusion LM	0.803	**0.830**	$+0.027$
LLaDA-8B-Instruct	CommonsenseQA	masked diffusion LM	**0.767**	0.763	-0.004
Dream-7B	BBH logical deduction	masked diffusion LM	0.429	**0.605**	$+0.176$ ($p\approx6\times10^{-16}$)
Dream-7B	Tracking shuffled objects	masked diff. LM (uniform prior)	0.248	**0.296**	$+0.048$
Dream-7B	ARC-Challenge	masked diff. LM (uniform prior)	0.820	**0.890**	$+0.070$
Dream-7B	CommonsenseQA	masked diff. LM (uniform prior)	0.783	**0.833**	$+0.050$
Open multimodal mathematics (order-only ablation)					
LLaDA-V [48]	MathVista [19]	masked diffusion VLM	0.483	**0.499**	$+0.016$ (n.s.)
LLaDA-V	MathVerse [49]	masked diffusion VLM	0.218	**0.222**	$+0.004$ (n.s.)
MMaDA-8B	MathVista	unified MM diffusion	**0.329**	0.328	-0.001 (n.s.)
MMaDA-8B	MathVerse	unified MM diffusion	0.204	**0.206**	$+0.002$ (n.s.)

the complete model-family results and available significance information omitted from the compact main-text visualization.

E.7 GLOBAL LOOKAHEAD ERROR AND EFFICIENCY

Block size trades sequential cost against accuracy: singleton absorption requires one evaluation per generated position, whereas batching reduces the evaluation count and reintroduces the joint dependence term. In the reported puzzle ablations, larger blocks lower accuracy (Appendix E.7.2).

The main-text timing in Figure 4(d) is for the profiled configuration. A separate GenEval profile at $K = 16$ reports Info-Gain about 7.2 times slower than LEDFLOW. These timings do not imply constant overhead across sequence lengths and vocabularies. The block-size study is in Appendix E.7.2.

Info-Gain at its minimal window is not the entropy policy. Because Info-Gain scores a candidate by its own entropy minus the mean entropy of the remaining positions after a one-step lookahead, one might expect it to coincide with LEDFLOW at window 1. It does not. On the 6.4M denoiser, evaluated along a shared LEDFLOW trajectory, the two select the same position in only 21.8% of steps on Nikoli and 33.6% on Sudoku-Extreme, every puzzle diverges at least once, and the final grids agree on 80% of Nikoli puzzles and 24% of Sudoku-Extreme puzzles. Where they diverge, LEDFLOW is at least as accurate (0.900 against 0.830 on Nikoli, and 0.296 against 0.284 on Sudoku-Extreme). The downstream-entropy term therefore reorders selections even at the smallest window, and Figure 4(c) measures how that reordering degrades as the window grows.

The fixed-state window experiment underlying Figure 4(c) varies only the number of unresolved-position entropies aggregated by Info-Gain. The mean absolute score error increases from 0.17 with one term to 1.76 under full aggregation, while absorption accuracy peaks at window 8 and then falls to 0.31. Because window 8 is the maximum of that curve, we use it as the Info-Gain configuration in every accuracy and runtime comparison reported in the main text, so the baseline is never evaluated at a setting we have shown to be suboptimal. This directly complements the runtime comparison in panel (d): local entropy selection avoids both the accumulated counterfactual estimates and their additional denoiser evaluations.

Table 26: Sudoku efficiency (6.4M denoiser, Nikoli n=100, mean 55.7 blanks/puzzle, five seeds, one H100). The single-position LEDFLOW policy evaluates the denoiser once per absorbed position. Solve accuracies for LEDFLOW and Euler@64 are the Table 3 values. Table 20 reports puzzle-level bootstrap intervals and paired McNemar tests.

sampler	denoiser evals.	ms/puzzle	solve-acc
LEDFLOW (single-position absorption)	**55.7**	**10.5**	$\mathbf{0.845}$
Euler@16	16	3.6	0.238 ± 0.014
Euler@32	32	5.7	0.290 ± 0.038
Euler@64	64	9.6	0.410
Euler@128	128	17.5	0.334 ± 0.022
Euler@256	256	33.3	0.350 ± 0.030

Table 27: Profiled wall-clock (generated puzzles, n=200, mean $\sim$52 blanks, one H100, and ms/puzzle includes each policy's score computation and denoiser reevaluation). LEDFLOW's accuracy advantage over the non-absorbing baselines grows with denoiser quality (to 0.970 at 21.3M), while uniformization remains less accurate at comparable wall-clock time.

	6.4M		21.3M	
sampler	acc	ms	acc	ms
LEDFLOW (single-position absorption)	0.925	5.7	**0.970**	9.7
Euler@64 / @256	0.520/0.560	8.2/28.2	0.830/0.810	12.8/46.1
uniformization (exact, unord.)	0.425	7.3	0.750	11.3

E.7.1 WALL-CLOCK COMPARISON

For GenEval at K=16, Flow/Euler, LEDFLOW, probability-margin sampling, and Info-Gain require 2.96, 2.96, 2.98, and 21.21 seconds per image and rank, respectively. Info-Gain is therefore 7.2 times slower than LEDFLOW.

Where the per-step cost goes. To separate the cost of the ordering score from the rest of the step, we profile the components of a single LEDFLOW step over the Nikoli puzzles. The denoiser forward pass takes 94.8% of the step, the ordering score (entropy, sorting, and applying the absorption count) 3.6%, and the absorption update itself 1.6%. Two consequences follow. First, scoring is a small fraction of a step, so the choice among entropy, probability margin, and maximum probability cannot by itself produce a large wall-clock difference. Timing gaps of that size reflect implementation rather than the score. Second, the cost of Info-Gain comes from its extra denoiser evaluations, not from aggregating entropies. These shares were measured on CPU while the GPUs were occupied. We report the relative split rather than absolute times, and the split is insensitive to the device because the forward pass dominates on any hardware.

E.7.2 BLOCK SIZE AND THE JOINT TERM

Two ablations address the block size b that Theorem 1 leaves free and the joint term of Proposition 2 that it does not optimize. Both use the 6.4M denoiser, entropy ordering, and argmax absorption, and vary only the number of positions absorbed per step.

Block size. Table 28 compares single-position absorption with fixed blocks of 2, 4, and 8 positions. Fixed blocks cut denoiser evaluations from 58 to 8 but lower solve accuracy from 0.900 to 0.720 on Nikoli and from 0.928 to 0.830 on generated puzzles, and the fraction of absorption events that absorb a wrong value rises monotonically with b. Absorbing several positions at once from independent marginals ignores their dependence, which is the $b>1$ penalty the decomposition predicts.

Isolating the joint term on multi-solution puzzles. On unique-solution puzzles the oracle conditional is a point mass and the total correlation of any block is zero (Lemma 2), so the loss in Table 28 mixes the joint term with the denoiser's own error. To isolate the joint term we build puzzles with 22,

Table 28: **Block size** (Sudoku, 6.4M denoiser, entropy ordering). Each cell reports puzzle solve accuracy / cell accuracy / fraction of incorrect absorptions, with the number of denoiser evaluations per puzzle. Larger blocks trade accuracy for fewer evaluations.

Block size	Nikoli (n=100)		Generated (n=500)	
	solve / cell / bad	NFE	solve / cell / bad	NFE
Single (b=1)	0.900 / 0.936 / 0.065	58	0.928 / 0.969 / 0.033	59
Fixed b=2	0.880 / 0.922 / 0.079	29	0.918 / 0.966 / 0.036	30
Fixed b=4	0.850 / 0.918 / 0.083	15	0.892 / 0.959 / 0.043	15
Fixed b=8	0.720 / 0.875 / 0.127	8	0.830 / 0.941 / 0.061	8

Table 29: **The joint term isolated** (Sudoku, 6.4M denoiser, entropy ordering, n=150 puzzles per givens level). Fraction of valid final grids on multi-solution puzzles as a function of block size. The last column reports the unique-solution puzzles in the same sweep.

Givens	Multi-solution	Single (b=1)	Block b=4	Block b=8	Unique, any b
22	150/150	1.000	0.507	0.267	—
26	150/150	1.000	0.667	0.207	—
30	147/150	0.993	0.708	0.460	1.000

26, or 30 givens by blanking complete grids, which at these densities are almost all multi-solution, and score grid validity (every row, column, and box a permutation of 1 to 9) rather than agreement with one reference solution. Table 29 shows that single-position absorption stays valid on essentially every puzzle, because re-conditioning after each absorption collapses the remaining ambiguity onto one consistent branch, whereas absorbing a block of mutually dependent cells from independent marginals mixes digits from different solution branches and breaks the constraints: validity falls to 0.51 to 0.71 at b=4 and 0.21 to 0.46 at b=8, and worsens as puzzles become more under-determined. On the few unique-solution puzzles in the same sweep, block absorption is perfectly valid at every block size. The penalty therefore appears only when the jointly absorbed cells are dependent, which is the total-correlation term itself.

E.7.3 Comparison with entropy-bounded block unmasking

EB-Sampler [2] absorbs, at each step, the lowest-entropy position together with any further positions whose cumulative entropy stays under a threshold ϵ. At ϵ=0 it reduces to single-position LEDFlow. We reimplement it on the puzzle tasks with the same denoiser, argmax absorption, and ordering (Table 30). At ϵ=0.5 it cuts denoiser evaluations about thirteen-fold (55.7 to 4.2 on Nikoli) and loses about fourteen points of solve accuracy (0.900 to 0.760, and 0.928 to 0.792 on generated puzzles), and both the accuracy loss and the incorrect-absorption fraction grow monotonically with ϵ. EB-Sampler's adaptive batching is slightly more evaluation-efficient than fixed blocks at matched accuracy, but it sits on the same frontier as Table 28: fewer steps are bought by absorbing dependent positions together, and single-position LEDFlow occupies the accuracy-optimal end.

F Extended related work

Discrete flow simulation and numerical correction. Discrete flow models generate categorical data through continuous-time probability paths and associated CTMC velocities [4, 29]. Euler and tau-leaping approximate these dynamics on a finite grid, while corrected samplers reduce temporal or spatial discretization error and uniformization simulates the learned CTMC more directly [36, 37, 50]. Related work also generalizes the interpolating path between masked and uniform corruption [33], and optimizes the time discretization itself, allocating steps where the discrete diffusion process changes fastest [9]. These methods improve how the native transition process is integrated, and the schedule axis is orthogonal to ours: an improved step allocation sets the grid on which the absorption count m_k is spent, whereas LEDFlow decides which positions leave that grid. In contrast, LEDFlow augments inference with an irreversible absorption decision while leaving the learned posterior, base probability path, and native update on active positions unchanged. The two

Table 30: **EB-Sampler reimplementation on Sudoku** (6.4M denoiser). Cells report solve accuracy / cell accuracy / fraction of incorrect absorptions and the number of denoiser evaluations per puzzle. ϵ is the cumulative-entropy threshold in nats. ϵ=0 is LEDFLOW.

Sampler	ϵ	Nikoli (n=100)		Generated (n=500)	
		solve / cell / bad	NFE	solve / cell / bad	NFE
LEDFLOW (single)	0	0.900 / 0.936 / 0.065	55.7	0.928 / 0.969 / 0.033	53.2
EB-Sampler	0.5	0.760 / 0.898 / 0.104	4.2	0.792 / 0.932 / 0.070	3.7
EB-Sampler	1.0	0.670 / 0.871 / 0.130	3.6	0.664 / 0.903 / 0.100	3.2
EB-Sampler	2.0	0.520 / 0.829 / 0.173	3.0	0.528 / 0.879 / 0.124	2.7
EB-Sampler	4.0	0.270 / 0.791 / 0.211	2.6	0.334 / 0.846 / 0.158	2.3

design axes are complementary: a corrected update may be applied to active positions before the next absorption decision.

Confidence-guided absorption and revision. Confidence-based easy-first selection has a long history in iterative masked generation. MaskGIT fixes high-confidence image tokens over successive iterations, while later methods learn planners, use model confidence, or reconsider uncertain predictions [5, 16, 17, 46, 25, 15, 43, 23]. Remasking samplers reintroduce revision into masked diffusion at inference time [39], and confidence thresholds decode several positions per step [42]. Info-Gain instead uses counterfactual lookahead to estimate how one decision changes uncertainty over the remaining positions [44]. On the theoretical side, Cai & Li [3] show that confidence-based decoding in masked diffusion language models is provably efficient, in the sense that committing high-confidence positions first controls the number of denoiser calls needed to reach a target quality. Their analysis takes the absorbing (masked) corruption as given, so every position is committed exactly once and never revised. LEDFLOW places this policy question inside a uniform discrete flow: it absorbs the lowest-entropy active positions and suppresses their future velocity, while all other positions continue under the native flow. Our analysis separates the joint factorization and conditional prediction errors of this local decision, derives entropy from an upper bound on the latter, and quantifies how prediction error accumulates with the global lookahead window.

Entropy-bounded and gradient-steered samplers. Two recent samplers for masked diffusion apply KL analysis to unmasking, and the gap between what they control and what irreversible absorption costs is the gap this paper closes. EB-Sampler [2] bounds the dependence discarded within one parallel block and spends that bound on speed, enlarging the block to reach 2 to 3 times fewer function evaluations at matched quality. Its analysis never asks which position should be made terminal. The ordering criterion is taken from existing local proxies. Under a revisable sampler that omission is benign, since a poor early choice can be overwritten. Under absorption it is not, because the absorption count is fixed before any ordering is chosen and therefore cannot see the error that ordering determines. Section 3.3 supplies exactly that missing quantity, and the two mechanisms are composable: the entropy threshold bounds the block, the entropy ranking orders it. BoE steering [27] approximates one-step lookahead by differentiating successor-state entropy with respect to the injected embedding, which places it on the lookahead side of the contrast drawn in Section 3.3, and inherits the window sensitivity that Lemma 1 identifies. It is also not portable to our setting: its token importance score is a first-order expansion along $e_i(\alpha) = e_m + \alpha(\tilde{e}_i - e_m)$, anchored at the mask embedding e_m, and a uniform discrete flow has no absorbing mask symbol to supply that basepoint. Reporting a number for it would require inventing one and would measure our surrogate rather than the published method, so we do not. Its authors additionally scope the underlying relaxation away from "symbolic reasoning with brittle constraints," the regime in which our largest gains occur. Neither method has released code at the time of writing.

Guidance, training, and benchmark scope. Guidance methods target reward- or evidence-weighted distributions, and reinforcement-learning methods fine-tune the denoiser for downstream rewards [35, 34]. These approaches alter the target, learned model, or training objective. LEDFLOW is a training-free inference policy. Standard multimodal-understanding benchmarks often require short answers, leaving limited output structure for an absorption policy to exploit. We therefore evaluate across a spectrum: understanding measures behavior on weakly constrained outputs, whereas

text-to-image generation and structured reasoning test outputs with stronger dependencies (Tables 10, 2, and 3).