REVISION PROVABLY REDUCES SEQUENTIAL COMPUTATION IN DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models (DLMs) generate multiple tokens with a single forward pass per decoding round, achieving greater parallelism than autoregressive models (ARMs). Revision promises further reductions in sequential computation by allowing a DLM to overwrite previously generated tokens and thereby reducing the number of decoding rounds. Yet it remains unclear whether this advantage persists as DLMs evolve. To assess the expressive power of DLMs with and without revision, we work in a model where each Transformer's decoding round uses AC^0 computation. We compare the minimum number of rounds each class needs to sample $(X, f_L(X))$, where the word X is uniform on $\{0,1\}^n$ and f_L is the membership function of a binary regular language L. We show that revision yields an asymptotic advantage. A DLM with revision exactly samples this distribution in at most two rounds without chain-of-thought (CoT) positions. However, a DLM without revision has tight round complexity $\Theta(\log n/\log\log n)$ even with a constant number of CoT positions for every $L \notin \mathsf{AC}^0$, i.e., every regular language whose membership cannot be computed by such circuits. Moreover, for every such L and every constant $\theta > 0$, $O(n^{1-\theta})$ CoT positions do not suffice to sample this distribution in constant rounds without revision. We further show that there is a separation when the goal is to approximately sample a length-n word uniformly from some regular language L. These results provide a theoretical framework for understanding how revision reduces sequential computation and expands the scope of parallel speedups achievable by DLMs.

1 INTRODUCTION

Diffusion language models (DLMs) have recently emerged as a promising alternative to autoregressive models (ARMs) because they allow much greater parallelism during generation (Nie et al., 2025; Khanna et al., 2025; Taïga et al., 2026). By decoding multiple masked positions simultaneously, DLMs reduce the sequential computation needed to produce a sequence.

This capability has driven rapid progress in DLM development. As DLMs evolve, it is unclear which design choices will continue to be effective. This uncertainty makes theoretical research on expressivity particularly relevant because it can examine what each design could achieve with every other aspect of the DLM optimized within that design, thereby clarifying its limits.

Revision is one such design choice, with recent studies reporting improvements in generation quality and decoding parallelism (Song et al., 2025; Bie et al., 2026; Wang et al., 2026). It allows a DLM to overwrite previously revealed tokens. This can enable more aggressive parallel decoding without making early predictions irrevocable, thereby reducing sequential computation. In this work, we compare the limits of DLMs with and without revision, examining whether revision can still reduce sequential computation even when models are optimized within their respective design classes.

To make this precise, we measure the minimum number of decoding rounds required to sample a target distribution at a given sequence length for each DLM class (with or without revision). A DLM starts with a sequence of mask tokens. In each round, a DLM without revision selects a subset of masked positions to decode in parallel. With revision, it can also select positions decoded in earlier rounds. A DLM may also use chain-of-thought (CoT) positions as extra workspace, storing tokens that can guide later rounds but are not part of the output. We use AC^0 computation per round

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.

Table 1: Round complexity for regular-language sampling with a constant CoT budget.

Distribution	Language class	DLM w/ revision	DLM w/o revision	Reference
$\mathrm{Pair}_{L,n}$	$L = \mathsf{PARITY}$	$= 2$	$\omega(1)^*$	Jiang et al. (2026)
	$L \in \mathsf{AC}^0$	≤ 2	≤ 2	Ours
	$L \notin \mathsf{AC}^0$	≤ 2	$\Theta(\log n / \log\log n)$	
$\mathrm{Word}_{L,n}$	$L = 0^*1^*$	≤ 2	$\Omega(\log n)$	Ours
	all regular L	≤ 2	—	

* This lower bound assumes no CoT positions.

(polynomial-size, constant-depth circuits with unbounded-fan-in AND and NOT gates) and $O(\log n)$ fair bits per token draw, and connect this circuit model to Transformer realizations in section 2.1.

To study asymptotic behavior, we consider a family of target distributions indexed by n and use round complexity to describe how the minimum number of rounds for exact sampling grows with n. Jiang et al. (2026) provide an initial result that separates the two model classes. They consider the input–output pair distribution for parity on n input bits, that is, the distribution of

$$(X_1, \ldots, X_n, \mathrm{parity}(X_1, \ldots, X_n)),$$

where $X_1, \ldots, X_n$ are independent fair bits. With no CoT positions, a DLM without revision cannot sample this distribution in any constant number of rounds, while a DLM with revision samples it in two rounds. This establishes a round-complexity gap between DLMs with and without revision.

This separation motivates a broader investigation into which functions yield such a gap and how large it can be. We consider the membership functions $f_L(x) \coloneqq \mathbf{1}[x \in L]$ of binary regular languages $L \subseteq \{0,1\}^*$, including parity. We study sampling from the input–output pair distribution (Viola, 2012; De & Watson, 2012; Viola, 2020)

$$\mathrm{Pair}_{L,n} \coloneqq \mathrm{Law}\big(X, f_L(X)\big), \qquad X \sim \mathrm{Unif}(\{0,1\}^n), \tag{1}$$

where $\mathrm{Unif}(\cdot)$ denotes the uniform distribution. We prove that the round-complexity gap extends to every binary regular language outside AC^0. Our first result (Theorem 3.1) shows that a DLM with revision exactly samples $\mathrm{Pair}_{L,n}$ in at most two rounds with no CoT positions, lifting the parity result of Jiang et al. (2026) to the entire class of binary regular languages.

For DLMs without revision, the round complexity depends on the circuit complexity of the language L (Theorem 4.1). If $L \in \mathsf{AC}^0$, i.e., if f_L is computable in AC^0, two rounds still suffice, so revision provides no advantage. If $L \notin \mathsf{AC}^0$, the tight round complexity is $\Theta(\log n / \log\log n)$ with a constant CoT budget, which gives rise to the separation between the two models.

We further show that this gap cannot be eliminated unless a DLM without revision uses a near-linear CoT budget to compensate for its inability to overwrite tokens (Theorem 4.5). Specifically, for every constant $\theta > 0$, $O(n^{1-\theta})$ CoT positions are insufficient for constant-round sampling by a DLM without revision, and the lower bound is essentially tight because n positions suffice.

In addition to the joint distribution $\mathrm{Pair}_{L,n}$, a regular language L induces the conditional distribution of X given $f_L(X) = 1$ (Goldwurm et al., 2001; Bernardi & Giménez, 2012; Oudinet et al., 2013),

$$\mathrm{Word}_{L,n} \coloneqq \mathrm{Unif}(L \cap \{0,1\}^n), \qquad \text{for } L \cap \{0,1\}^n \neq \varnothing.$$

We show that revision can yield a round-complexity gap for $\mathrm{Word}_{L,n}$ as well. Since its atoms need not have dyadic probabilities (of the form $a/2^b$ for integers $a, b \geq 0$), we study approximate sampling in total variation (TV) distance. For every fixed $\epsilon > 0$, there is a DLM with revision that approximates $\mathrm{Word}_{L,n}$ to within ϵ in two rounds using a constant number of CoT positions. In contrast, under the same CoT constraint, a DLM without revision requires $\Omega_\epsilon(\log n)$ rounds for $L = 0^*1^*$.

Together, these results characterize the round-complexity gap for $\mathrm{Pair}_{L,n}$ and show that a gap can also arise for $\mathrm{Word}_{L,n}$, as summarized in Table 1. Thus, even when models are optimized within their respective design classes, revision can yield an asymptotic improvement in round complexity rather than merely a constant-factor gain.

More broadly, our techniques establish significant results on constant-depth sampling for distributions induced by regular languages, including the input–output pair distribution $\mathrm{Pair}_{L,n}$ and the conditional distribution $\mathrm{Unif}(L \cap \{0,1\}^n)$, and for acceptance-conditioned and unconditioned trajectories of finite Markov chains. These results form a second main contribution of this work. See Appendix E for details.

2 PRELIMINARIES

This section formalizes exact sampling by DLMs, defines the round and CoT complexities used in our results, and reviews notation for regular languages.

2.1 DLM SAMPLING MODEL

DLM decoding process. Let the token vocabulary be $V = \{0,1\}$ and let $\mathtt{M}$ denote a mask token. A **DLM** is specified by (N, D, ρ, F, p), where N is the sequence length, D is the number of decoding rounds, $\rho \in \{\circlearrowleft, \times\}$ specifies whether revision is allowed, F is the unmasking policy, and p is the predictor. The decoding process (Algorithm 1) starts from a sequence of N mask tokens and runs for D rounds. In each round, given the current sequence z, the deterministic policy F selects a set of positions, and the predictor p independently draws a fresh token at each selected position i from its marginal distribution $p^{(i)}(\cdot \mid z)$ over V. After D rounds, every position must contain a token from V. With a CoT budget k, the N positions are partitioned into a CoT set $\mathcal{C}$ of size k and an output set $\mathcal{O}$ of size $N-k$. Both the policy F and predictor p read the full sequence, but only the output subsequence $z_{\mathcal{O}}$, in position order, is returned.

Algorithm 1 DLM decoding

Input: CoT budget k, with $[N] = \mathcal{C} \sqcup \mathcal{O}$ and $|\mathcal{C}| = k$
$z \leftarrow \mathtt{M}^N$
for $t = 1, \ldots, D$ **do**
  $S \leftarrow F(z)$
  for all $i \in S$ **in parallel do**
    sample $z_i \sim p^{(i)}(\cdot \mid z)$ over V
return $z_{\mathcal{O}}$ ▷ discard the CoT subsequence $z_{\mathcal{C}}$

In a **DLM without revision** ($\rho = \times$), the policy may select only masked positions, so $F(z) \subseteq \{i : z_i = \mathtt{M}\}$. Once a token is revealed, it is never updated. In a **DLM with revision** ($\rho = \circlearrowleft$), the policy may select any position, so $F(z) \subseteq [N]$. The formulation above aligns with common decoding processes for DLMs with and without revision (Nie et al., 2025; Song et al., 2025; Li et al., 2025).

DLM circuit model. An AC^0 **circuit family** consists of Boolean circuits with unbounded-fan-in AND gates and NOT gates. The family has a common polynomial size bound in the number of input bits and a common constant depth bound. Depth is the maximum number of gates on an input-to-output path, including NOT gates. A **randomized-AC^0 circuit family** is an AC^0 circuit family whose circuits additionally take polynomially many independent fair bits. Consider such a circuit R with ordinary input x, r fair bits, and finite output alphabet Ω. On input x, the circuit R **exactly samples** a distribution $\mathcal{D}$ on Ω if, for every $z \in \Omega$,

$$\Pr_{U \sim \mathrm{Unif}(\{0,1\}^r)}[R(x,U) = z] = \Pr_{Z \sim \mathcal{D}}[Z = z]. \tag{2}$$

We study **families of DLMs** $(S_n)_{n \geq 1}$, where $S_n = (N_n, D_n, \rho, F_n, p_n)$ and the revision status ρ is the same for all n. The policies $(F_n)_{n \geq 1}$ are computed by an AC^0 circuit family, while the predictor marginals $(p_n^{(i)})_{n \geq 1,\, i \in [N_n]}$ are sampled by a randomized-AC^0 circuit family. We additionally assume that each predictor marginal uses $O(\log n)$ independent fair bits. Both circuit families take the current sequence z of length N_n as their ordinary input, and the randomized family samples from each conditional distribution $p_n^{(i)}(\cdot \mid z)$ exactly. We omit the subscript n when the length is clear.

Remark. The following points explain why our circuit model is a reasonable choice and clarify how to interpret the separations obtained within it. First, an unrestricted round could hide arbitrarily

complex computation rather than represent a bounded model forward pass. Second, the per-round AC^0 restriction matches the logit expressivity of the constant-depth, constant-precision, polynomial-width Transformers considered below (Lemma D.1). Third, we aim to use the separation established in the AC^0 model to offer a theoretical indication that revision can extend the limits of parallel sampling by DLMs more broadly. Finally, for approximate uniform sampling from $L = 0^*1^*$, the no-revision lower bound holds even with computationally unrestricted policies and predictors, provided the CoT budget is constant (Theorem 5.2).

Transformer realization. To connect the circuit model to a neural implementation, we consider constant-depth, constant-precision, polynomial-width Transformers (Li et al., 2024). DLMs use full attention, and ARMs use causal attention. We say that a model family is **Transformer-realized** if its policies and predictors are implemented by this architecture, with token sampling and unmasking as in Algorithm 4 and Algorithm 5.

Theorem 2.1. *All round and CoT upper and lower bounds in Sections 3 to 5 remain valid for Transformer-realized families under their stated assumptions.*

The lower bounds apply because Transformer-realized families satisfy the circuit-model restrictions. The upper bounds follow because the logits implementing our constructions' policies and predictors are AC^0-computable. The proof is given in Appendix D.

DLM sampling task. Let $\mathcal{D} = (\mathcal{D}_n)_{n\in I}$ be a distribution family with index set $I \subseteq \mathbb{N}_{\geq 1}$, where $\mathcal{D}_n$ is a distribution on $\{0,1\}^{\ell_n}$. For an error tolerance $\epsilon \geq 0$, a DLM family $S = (S_n)_{n\geq 1}$ **ϵ-approximately samples $\mathcal{D}$ with CoT budget** $(k_n)_{n\geq 1}$ if, for every $n \in I$, the model S_n has $N_n = \ell_n + k_n$ positions partitioned into an output set $\mathcal{O}_n$ of size ℓ_n and a CoT set $\mathcal{C}_n$ of size k_n, and the marginal law Q_n of the output subsequence $z_{\mathcal{O}_n}$ is within total variation distance ϵ of $\mathcal{D}_n$, i.e.,

$$d_{\mathrm{TV}}(Q_n, \mathcal{D}_n) = \frac{1}{2} \sum_{x\in\{0,1\}^{\ell_n}} |Q_n(x) - \mathcal{D}_n(x)| \leq \epsilon.$$

In particular, when $\epsilon = 0$, we say that the DLM family S **samples $\mathcal{D}$ with CoT budget** $(k_n)_{n\geq 1}$, or **samples $\mathcal{D}$ without CoT** if $k_n = 0$ for all n.

For a language L, write the distribution families as $\mathrm{Pair}_L = (\mathrm{Pair}_{L,n})_{n\geq 1}$ and $\mathrm{Word}_L = (\mathrm{Word}_{L,n})_{n\in\{n\geq 1 : L\cap\{0,1\}^n \neq \varnothing\}}$.

2.2 Round and CoT complexity

For a DLM family $S = (S_n)_{n\geq 1}$, define its **(worst-length) round complexity** by $R_S(n) := \max_{1\leq m\leq n} D_m$. If S has CoT budget $(k_n)_{n\geq 1}$, define its **(worst-length) CoT complexity** by $K_S(n) := \max_{1\leq m\leq n} k_m$.

Remark. Rather than measuring complexity at length n by D_n alone, we use the prefix maximum $R_S(n)$ for three reasons. First, for a positive, nondecreasing function $T(n)$, $D_n \in O(T(n))$ is equivalent to $R_S(n) \in O(T(n))$. Second, a pointwise lower bound $D_n \in \Omega(T(n))$ need not hold because some lengths may admit $O(1)$-round samplers while others require $\Omega(T(n))$ rounds. Third, a weaker statement $D_n \notin o(T(n))$ does not capture this worst-length behavior. In contrast, $R_S(n) \in \Omega(T(n))$ is strictly stronger because it guarantees that lengths requiring $\Omega(T(n))$ rounds occur densely. The same three reasons apply to CoT complexity.

Finally, we define the **round complexity of a distribution family** $\mathcal{D}$. For an error tolerance $\epsilon \geq 0$ and revision status $\rho \in \{\circlearrowleft, \times\}$, let $\mathcal{S}^\rho_\epsilon(\mathcal{D})$ denote the set of DLM families with revision status ρ that ϵ-approximately sample $\mathcal{D}$ with a constant CoT budget, i.e., $\sup_n k_n < \infty$. For a positive, nondecreasing function T, we write

$$\begin{aligned} R^\rho_\epsilon(\mathcal{D}) \in O(T(n)) \quad &\Longleftrightarrow \quad \exists S \in \mathcal{S}^\rho_\epsilon(\mathcal{D}), \quad R_S(n) \in O(T(n)), \\ R^\rho_\epsilon(\mathcal{D}) \in \Omega(T(n)) \quad &\Longleftrightarrow \quad \forall S \in \mathcal{S}^\rho_\epsilon(\mathcal{D}), \quad R_S(n) \in \Omega(T(n)). \end{aligned}$$

We write $R^\rho_\epsilon(\mathcal{D}) \in \Theta(T(n))$ when both the upper and lower bounds above hold. In particular, $R^\rho(\mathcal{D}) := R^\rho_0(\mathcal{D})$ denotes exact-sampling round complexity.

2.3 Regular languages and DFA runs

A binary language $L \subseteq \{0,1\}^*$ is **regular** if it is recognized by a **deterministic finite automaton (DFA)** (Rabin & Scott, 1959). A DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ over a finite alphabet Σ has a finite state set Q, transition function $\delta : Q \times \Sigma \to Q$, start state $q_0 \in Q$, and accepting-state set $F \subseteq Q$. Extend the transition function δ to words $w \in \Sigma^*$ by $\delta^*(q, \varepsilon) \coloneqq q$ for the empty word ε and $\delta^*(q, wa) \coloneqq \delta(\delta^*(q, w), a)$ for $a \in \Sigma$, so that $\delta^*(q, w)$ is the state reached from q after reading w. A word $x \in \Sigma^n$ is accepted by the DFA if $\delta^*(q_0, x) \in F$. On input x, the DFA starts at $y_0 = q_0$ and visits $y_i \coloneqq \delta^*(q_0, x_1 \cdots x_i)$ for $i = 1, \ldots, n$. The **run** of $\mathcal{A}$ on x is $(x, y_0, y_1, \ldots, y_n)$.

3 Sampling Regular Languages in Two Rounds with Revision

We show that, for every binary regular language L, DLMs with revision can sample $\mathrm{Pair}_{L,n}$ in two rounds without CoT. This lifts the two-round result of Jiang et al. (2026) for DLMs with revision from parity to the entire class of binary regular languages. Our proof techniques also yield several further contributions to the study of constant-depth sampling of finite-state distributions (Appendix E).

Theorem 3.1. *For every binary regular language L, there is a DLM family $(S_n)_{n\geq 1}$ with $\rho = \circlearrowleft$ that samples $\mathrm{Pair}_{L,n}$ without CoT and uses $D_n = 2$ decoding rounds for every n. Consequently, $R^{\circlearrowleft}(\mathrm{Pair}_L) \in O(1)$.*

Before giving the formal proof, we use parity to illustrate how a DLM with revision can sample the input–output pair distribution of a regular language in two rounds.

Example. In the first round, sample independent fair bits $U_1, \ldots, U_n$ in the n word positions. In the second round, overwrite the word positions with X and fill the label position with Y, where

$$(X_1, \ldots, X_n, Y) = (U_1, U_1 \oplus U_2, \ldots, U_{n-1} \oplus U_n, U_n).$$

The seed-to-word map $U \mapsto X$ is bijective, so the word X is uniform, and $Y = U_n = \bigoplus_{i=1}^{n} X_i$.

3.1 Proof by Reduction

At a high level, the proof of Theorem 3.1 reduces two-round sampling to the following randomized-AC^0 run-sampling result for a DFA recognizing L.

Proposition 3.2. *For every DFA $\mathcal{A}$ over $\{0,1\}$, there is a randomized-AC^0 circuit family $(C_n)_{n\geq 1}$ such that each circuit C_n takes n fair bits and exactly samples the run of $\mathcal{A}$ on a uniform word $X \sim \mathrm{Unif}(\{0,1\}^n)$.*

To prove Proposition 3.2, a first attempt is to sample a uniform word X and then compute the recurrence $y_i = \delta(y_{i-1}, X_i)$. This approach would require the recurrence to be computable in AC^0, which would place every regular language in AC^0. In general, this is impossible.

Nevertheless, the run of a DFA with an aperiodic transition monoid is computable in AC^0 (Barrington et al., 1992). We define this property and then formalize this lemma. Its proof appears in Appendix A.

Definition 3.3 (Transition monoid and aperiodicity)**.** For a DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, each word $u \in \Sigma^*$ induces the transformation $T_u(q) \coloneqq \delta^*(q, u)$. These transformations form the finite transition monoid

$$M_{\mathcal{A}} \coloneqq \{T_u : u \in \Sigma^*\}, \qquad T_{uv} = T_v \circ T_u. \tag{3}$$

$M_{\mathcal{A}}$ is aperiodic if each element a satisfies $a^{j+1} = a^j$ for some $j \geq 1$ (Schützenberger, 1965).

Lemma 3.4. *Let $\mathcal{B}$ be a DFA with state set H, alphabet Σ, transition function τ, and an aperiodic transition monoid. There is an AC^0 circuit family that, given a start state $s \in H$ and a word $u \in \Sigma^n$, outputs the state $\tau^*(s, u)$ reached from s after reading u.*

We exploit this result by reducing run sampling for $\mathcal{A}$ to run sampling for a hidden DFA $\mathcal{B}$ whose transition monoid is aperiodic, so Lemma 3.4 lets us compute the entire run of $\mathcal{B}$ in AC^0 by evaluating all prefixes of its input block sequence in parallel. Specifically, $\mathcal{B}$ has a finite state set H, takes b-bit input blocks for a constant b, and has a projection $\phi : H \to Q$. To represent $\mathcal{A}$, the projected state reached from any hidden state h after a uniform block must have the same distribution as the state reached by $\mathcal{A}$ from $\phi(h)$ after a uniform b-bit block. Lemma 3.5 formalizes the construction of $\mathcal{B}$.

Lemma 3.5. *For a DFA $\mathcal{A} = (Q, \{0,1\}, \delta, q_0, F)$, there are a constant block length b, a hidden DFA $\mathcal{B}$ with state set H and transition function $\tau : H \times \{0,1\}^b \to H$, and a surjective projection $\phi : H \to Q$ such that*

1. *for every $h \in H$ there is a bijection $\pi_h : \{0,1\}^b \to \{0,1\}^b$ such that, for all $w \in \{0,1\}^b$,*

$$\phi\big(\tau(h, w)\big) = \delta^*(\phi(h), \pi_h(w)), \tag{4}$$

2. *the transition monoid of $\mathcal{B}$ is aperiodic.*

The full proof of Lemma 3.5 appears in Appendix A. To give an example, we use parity to show how a hidden DFA satisfies this lemma.

Example. The parity DFA has $Q = \{0,1\}$, initial and accepting state 0, and $\delta(s, a) = s \oplus a$. Take $b = 1$, $H = \{0,1\}$, $\phi(h) = h$, $\tau(h, u) = u$, and $\pi_h(u) = h \oplus u$. Then π_h is a bijection and $\phi(\tau(h, u)) = u = \delta(\phi(h), \pi_h(u))$. For each fixed h and uniform bit U, the bijection makes $\pi_h(U)$ uniform, so the projected hidden transition has the same distribution as the parity DFA's transition under a uniform input bit. Every nonempty hidden input resets the state to its last bit, so $\mathcal{B}$ is aperiodic. Its prefix states are $h_i = U_i$, recovering the two-round parity sampler above.

Remark. Restricting Lemma 3.5 to the modular-counting languages $L_q = \{x \in \{0,1\}^* : |x|_1 \equiv 0 \pmod q\}$, where $q \geq 2$ and $|x|_1$ counts the ones in x, provides an intermediate case for understanding the general construction and proof. The parity construction motivates the extension to L_q. This extension groups input bits into blocks, a technique that motivates the extension to arbitrary regular languages. For readers interested in the details, Appendix A.1 gives the explicit block transitions for L_q and shows how they yield two-round sampling.

Given $\mathcal{B}$, the sampler processes the fair bits in blocks and computes their boundary states in parallel. We sketch the proof of Proposition 3.2 below. The full proof appears in Appendix A.

Proof sketch of Proposition 3.2. Write $n = bm + r$ with $0 \leq r < b$, and fix a hidden state $\iota(q) \in \phi^{-1}(q)$ for every $q \in Q$. Partition the n fair bits into m hidden blocks $u_1, \ldots, u_m \in \{0,1\}^b$ and a tail $v \in \{0,1\}^r$. Starting from the hidden state $h_0 = \iota(q_0)$, it runs $\mathcal{B}$ on the blocks and obtains the prefix states $h_j = \tau^*(\iota(q_0), u_1 \cdots u_j)$. Since $\mathcal{B}$ is aperiodic, all of them are computable in parallel by AC^0 circuits (Lemma 3.4). The blocks are then decoded in parallel. The bijection $\pi_{h_{j-1}}$ of Equation (4) projects the hidden block u_j to a block w_j, and running δ on w_j from the boundary state $y_{(j-1)b} = \phi(h_{j-1})$ computes the states $y_{(j-1)b+1}, \ldots, y_{jb-1}$.

Conditioned on the preceding blocks, the hidden block u_j remains uniform, and the bijection $\pi_{h_{j-1}}$ preserves its distribution. Hence the concatenated word $w_1 \cdots w_m v$ is uniform on $\{0,1\}^n$. The maps τ, π_h, and δ are represented by constant-size tables hardwired into the circuit, and each block is processed in constant depth, so the sampler is a randomized-AC^0 circuit family. □

Finally, a DLM with revision implements C_n in two rounds to prove Theorem 3.1. It first samples C_n's fair-bit input, then overwrites the seed positions with the output word and fills the label position with its acceptance bit.

Proof of Theorem 3.1. Let $\mathcal{A} = (Q, \{0,1\}, \delta, q_0, F)$ recognize L, and let $(C_n)_{n \geq 1}$ be the circuit family given by Proposition 3.2. In the first round, the policy selects the first n positions and their predictors reveal independent fair bits, which form the random input to C_n. In the second round, the policy selects all $n + 1$ positions. The predictors overwrite the first n positions with the word x in the run produced by C_n, and the predictor at the last position outputs $\mathbf{1}[y_n \in F]$. These outputs are deterministic AC^0 functions of the first-round bits. The terminal sequence has law $\mathrm{Pair}_{L,n}$. □

Remasking. Besides revision, remasking is another decoding design that allows a DLM to return a revealed position to the masked state and decode it again in a later round (Wang et al., 2025; Huang et al., 2026). In a **DLM with remasking**, the policy F may likewise select revealed positions to remask, but the local predictor $p^{(i)}$ at each such position outputs only `M` rather than a token in $\{0,1\}$. The position can receive a new bit in a later round.

By adding only a constant number of CoT positions, a DLM with remasking can also exactly sample $\mathrm{Pair}_{L,n}$ in four rounds (Theorem A.3). This result shows that constant-round sampling extends across these DLM decoding designs that revisit revealed positions.

4 Limits of DLMs without Revision

The two-round sampler with revision sets a benchmark for regular-language sampling. In this section, we establish a round-complexity separation between DLMs with and without revision and show that a small CoT budget cannot eliminate it. We also discuss a separation between ARMs and DLMs.

4.1 Tight Superconstant Round Complexity without Revision

This subsection shows that, with a constant CoT budget, DLMs without revision may require superconstant rounds to exactly sample $\mathrm{Pair}_{L,n}$.

Theorem 4.1. *For every binary regular language L, the round complexity satisfies $R^{\times}(\mathrm{Pair}_L) \in O(1)$ if $L \in \mathsf{AC}^0$. If $L \notin \mathsf{AC}^0$, then*

$$R^{\times}(\mathrm{Pair}_L) \in \Theta\left(\frac{\log n}{\log\log n}\right).$$

Proof. If $L \in \mathsf{AC}^0$, a DLM samples the uniform word X in the first round, and in the second round its label predictor outputs $f_L(X)$, computed by an AC^0 circuit. For $L \notin \mathsf{AC}^0$, the matching bounds follow from Theorems 4.3 and 4.4. □

For $L \notin \mathsf{AC}^0$, the lower bound is connected to the circuit complexity of f_L. Without revision, a DLM cannot reuse revealed positions as workspace, and sampling $\mathrm{Pair}_{L,n}$ becomes as hard as computing f_L, as described below. Intuitively, such a model essentially samples a word X first and then computes $f_L(X)$.

Lemma 4.2. *Consider a DLM family S without revision that exactly samples $\mathrm{Pair}_{L,n}$ with CoT budget $(k_n)_{n\geq 1}$. For every n, there are Boolean circuits $(T_{n,b})_{b\in\{0,1\}^{k_n}}$ of size polynomial in $n+k_n$ and depth $O(D_n)$ such that*

$$c = f_L(x) \quad \Longleftrightarrow \quad T_{n,b}(x,c) = 1 \text{ for some } b \in \{0,1\}^{k_n}.$$

Proof. Fix n, a CoT string $b \in \{0,1\}^{k_n}$, and a candidate output $z = (x,c) \in \{0,1\}^{n+1}$. We construct $T_{n,b}$ to test whether S_n can produce z with CoT transcript b.

The policy is deterministic and revealed bits never change, so (b,z) determines one candidate trace. Set $w \leftarrow \mathtt{M}^{n+1+k_n}$. In each round, compute the set of selected positions $I = F_n(w)$ and check, for every $i \in I$, that the predictor at i can output the prescribed bit. Then fill these positions with their prescribed bits. Reject if any check fails or if any positions remain masked after the final round.

Each local check can be implemented by an explicit circuit. Let the circuit $C_{n,i}(w,r)$ implement the predictor at position i, padded to use $s_n = O(\log n)$ fair bits. For $a \in \{0,1\}$, the check is

$$\mathbf{1}[p_n^{(i)}(a \mid w) > 0] = \bigvee_{r\in\{0,1\}^{s_n}} \mathbf{1}[C_{n,i}(w,r) = a].$$

Each local check has polynomial size and constant depth because $2^{s_n} = \mathrm{poly}(n)$, and the checks within a round run in parallel. Since a no-revision trace has at most $n+1+k_n$ nonempty rounds, unrolling it yields a circuit of size polynomial in $n+k_n$ and depth $O(D_n)$. □

Theorem 4.3. *For a binary regular language L outside AC^0, sampling Pair_L without revision with a constant CoT budget has round complexity $R^{\times}(\mathrm{Pair}_L) \in \Omega(\log n/\log\log n)$.*

Proof. For a constant CoT budget, the circuits in Lemma 4.2 can be combined into a polynomial-size Boolean circuit family of depth $O(D_n)$ recognizing L. Applying the regular-language depth lower bound (Lemma B.3) yields the claimed round lower bound. □

The $\Omega(\log n/\log\log n)$ lower bound is matched by Theorem 4.4 (with the full proof in Appendix B.2). Together, these results lift the parity result of Jiang et al. (2026) for DLMs without revision to the entire class of binary regular languages and determine the tight round complexity for every language in the class.

Theorem 4.4. *For every binary regular language L, there is a DLM family without revision that samples $\mathrm{Pair}_{L,n}$ without CoT in $O(\log n/\log\log n)$ decoding rounds for every n. Consequently, the round complexity satisfies $R^{\times}(\mathrm{Pair}_L) \in O(\log n/\log\log n)$.*

4.2 Near-Linear CoT is Required to Replace Revision

The results in Section 3 and Section 4.1 establish a round-complexity gap between DLMs with and without revision. This subsection shows that adding a small number of CoT positions to DLMs without revision does not eliminate the gap. A near-linear CoT budget is necessary, and this bound is tight up to polylogarithmic factors.

Theorem 4.5. *For a binary regular language L outside AC^0, n binary CoT positions suffice for a two-round DLM family without revision that samples $\mathrm{Pair}_{L,n}$, whereas $O(n^{1-\theta})$ CoT positions do not suffice for constant-round sampling for any constant $\theta > 0$.*

The lower and upper bounds follow from Theorems 4.6 and 4.7, respectively. We first give the lower bound. By Lemma 4.2, fixing a CoT string yields a shallow circuit that accepts only correctly labeled pairs. The circuits for all CoT strings cover every such pair. By Lemma B.5, for $L \notin \mathsf{AC}^0$, this requires $2^{\Omega(n/(\log n)^C)}$ polynomial-size constant-depth circuits for some constant $C > 0$, yielding the CoT lower bound. The full proof is given in Appendix B.3.

Theorem 4.6. *For a binary regular language $L \notin \mathsf{AC}^0$, consider a DLM family S without revision that exactly samples $\mathrm{Pair}_{L,n}$ and has $R_S(n) \in O(1)$. There is a fixed integer $i_S \geq 0$ such that*

$$K_S(n) = \Omega_S\left(\frac{n}{(\log n)^{i_S}}\right).$$

In particular, S does not have CoT complexity $O(n^{1-\theta})$ for any constant $\theta > 0$.

The lower bound is matched up to polylogarithmic factors by Theorem 4.7. Its proof adapts the construction for DLMs with revision in Lemma 3.5, as formalized in Lemma B.7. The full proof appears in Appendix B.3.

Theorem 4.7. *For every binary regular language L and every fixed integer $i \geq 0$, there is a two-round DLM family without revision that samples $\mathrm{Pair}_{L,n}$ using $O_L(n/(\log n)^i)$ CoT positions. For $i = 0$, the construction can use exactly n CoT positions.*

4.3 Comparison with Autoregressive Models

An ARM generates one token per round from left to right, so sampling $\mathrm{Pair}_{L,n}$ with k_n CoT positions takes $n + 1 + k_n$ rounds. In particular, it requires at least $n + 1$ rounds regardless of its CoT budget. This sequential decoding constraint separates ARMs from DLMs, which can decode many tokens in a single round and therefore sample the same distributions in sublinear or even constant rounds.

Table 2: Round-complexity comparison across models and CoT budgets for $\mathrm{Pair}_{L,n}$ when $L \notin \mathsf{AC}^0$.

Model	CoT budget	Round complexity
ARM	$O(n)$	$\Theta(n)$
DLM w/o revision	0	$\Theta(\log n/\log\log n)$
DLM w/o revision	$O(n)$	≤ 2
DLM w/ revision	0	≤ 2

For every binary regular language L, n CoT positions suffice for an ARM to sample $\mathrm{Pair}_{L,n}$ in $2n+1$ rounds (lemma B.8). Table 2 compares the round bounds for ARMs and DLMs.

5 A Round Separation for Approximate Conditional Sampling

We have characterized the gap between DLMs with and without revision for sampling $\mathrm{Pair}_{L,n}$, including tight round bounds (Sections 3 and 4.1) and the CoT budgets under which the gap persists

(Section 4.2). Conditioning the uniform word X on $f_L(X) = 1$ gives $\mathrm{Word}_{L,n}$, another standard sampling target studied in the theory of regular languages (Goldwurm et al., 2001; Bernardi & Giménez, 2012; Oudinet et al., 2013). We show that revision can also yield a round-complexity gap for approximate sampling of this distribution.

For any fixed TV error $0 < \epsilon < 1$, two rounds and a constant CoT budget suffice with revision (Theorem 5.1). In contrast, for the regular language 0^*1^*, reaching the same error without revision requires $\Omega(\log n)$ rounds with any constant CoT budget (Theorem 5.2). The full proofs are given in Appendices C.2 and C.3.

Theorem 5.1. *For every binary regular language L and error tolerance $0 < \epsilon \leq 1/2$, there is a DLM family with revision that ϵ-approximately samples $\mathrm{Word}_{L,n}$ in two rounds using $O_L(\log(1/\epsilon))$ CoT positions. The policy and predictor circuits have size $\mathrm{poly}_L(n, 1/\epsilon)$ and depth $O_L(1)$.*

Theorem 5.2. *For the language $L = 0^*1^*$, every constant CoT bound K, and every error tolerance $0 < \epsilon < 1$, any DLM family $S = (S_n)_{n\geq 1}$ without revision that ϵ-approximately samples $\mathrm{Word}_{L,n}$ with $k_n \leq K$ CoT positions must use $D_n = \Omega_{K,\epsilon}(\log n)$ decoding rounds. This lower bound holds even when the policies and predictors are computationally unrestricted.*

6 Related Work

Remasking and revision in DLMs. Without remasking or revision, decoded tokens remain fixed, allowing early errors to affect later predictions (Wang et al., 2025; Bie et al., 2026). Wang et al. address this by remasking decoded tokens for resampling, while Bie et al. edit them directly. Jiang et al. (2026) show that remasking or revision lets DLMs reuse token positions as working memory and reduces round complexity of sampling the distribution $\mathrm{Pair}_{\mathsf{PARITY},n}$ compared with no-revision decoding.

Transformer expressiveness. Circuit complexity provides a framework for analyzing Transformer expressiveness. For constant-depth, constant-precision Transformers of polynomial width, Li et al. (2024) show that their recognition power is exactly AC^0. They also show that CoT enables the simulation of Boolean circuits with a number of steps proportional to circuit size. For masked DLMs, Svete & Sabharwal (2026) relate their reasoning power to CoT and looped Transformers and establish an $O(\log n)$-round upper bound for regular-language recognition.

Recognition and sampling of regular languages. A regular language belongs to AC^0 if and only if its stable syntactic semigroup is aperiodic (Barrington et al., 1992; Straubing, 1994). Sampling can be easier than recognition, as illustrated by parity, which lies outside AC^0 but whose input–output pair distribution admits an exact randomized-AC^0 sampler (Viola, 2012). For uniform generation of accepted words, Goldwurm et al. (2001) study logspace-uniform probabilistic Boolean circuits with fair bits, allowing failure while ensuring exact uniformity on successful outputs.

7 Conclusion

Within our circuit model and with a constant CoT budget, revision lowers the round complexity of exact input–output pair sampling for every regular language outside AC^0. For approximate uniform accepted-word sampling, a gap holds for the language $L = 0^*1^*$ even with computationally unrestricted policies and predictors and a constant CoT budget.

Our proofs trace these gaps to how DLMs with and without revision use workspace. Revision lets a DLM first store random bits in its output positions and then overwrite them with the sample computed from those bits. Without revision, a constant CoT budget does not remove the round cost of recognizing a language outside AC^0. For input–output pair sampling, more CoT positions can store intermediate results, but closing the round gap requires a near-linear budget.

Together, these results provide a theoretical framework for understanding how revision reduces sequential computation and expands the scope of parallel speedups achievable by DLMs.

Reproducibility Statement

Section 2 specifies the sampling model, computational assumptions, and target distributions. The main results and their additional assumptions are stated in Sections 3 to 5. Their proofs and supporting constructions appear in these sections and in Appendices A to E.

AI use statement

We used generative AI tools to assist with drafting and editing parts of the manuscript, formulating mathematical claims, and developing proof arguments. The authors checked the claims and proofs, revised the AI-assisted text, and take responsibility for the final content of this work.

References

David A. Mix Barrington, Kevin Compton, Howard Straubing, and Denis Thérien. Regular languages in NC^1. *Journal of Computer and System Sciences*, 44(3):478–499, 1992. doi: 10.1016/0022-0000(92)90014-A.

Nicolas Basset, Jean Mairesse, and Michèle Soria. Uniform sampling for networks of automata. In *28th International Conference on Concurrency Theory (CONCUR 2017)*, volume 85 of *Leibniz International Proceedings in Informatics*, pp. 36:1–36:16, 2017. doi: 10.4230/LIPIcs.CONCUR.2017.36.

Olivier Bernardi and Omer Giménez. A linear algorithm for the random sampling from regular languages. *Algorithmica*, 62:130–145, 2012. doi: 10.1007/s00453-010-9446-5.

Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, et al. LLaDA2.1: Speeding Up Text Diffusion via Token Editing. *arXiv preprint arXiv:2602.08676*, 2026.

Anindya De and Thomas Watson. Extractors and lower bounds for locally samplable sources. *ACM Transactions on Computation Theory (TOCT)*, 4(1):1–21, 2012.

Massimiliano Goldwurm, Beatrice Palano, and Massimo Santini. On the circuit complexity of random generation problems for regular and context-free languages. In *STACS 2001*, volume 2010 of *Lecture Notes in Computer Science*, pp. 305–316, 2001. doi: 10.1007/3-540-44693-1_27.

Jordan Horacsek, Chin Ho Lee, Igor Shinkar, Emanuele Viola, and Renfei Zhou. Local samplers for product distributions. In *53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026)*, volume 374 of *Leibniz International Proceedings in Informatics*, pp. 109:1–109:23, 2026. doi: 10.4230/LIPIcs.ICALP.2026.109.

Zemin Huang, Yuhang Wang, Zhiyang Chen, and Guo-Jun Qi. Don't settle too early: Self-reflective remasking for diffusion language models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5d8bc24ef15b7a808b6dc403c01583e2-Abstract-Conference.html.

Thore Husfeldt. *Dynamic Computation*. Brics dissertation series ds-97-3, University of Aarhus, 1997. URL https://brics.dk/DS/97/3/BRICS-DS-97-3.pdf.

Haozhe Jiang, Nika Haghtalab, and Lijie Chen. Diffusion language models are provably optimal parallel samplers. In *International Conference on Learning Representations*, 2026.

Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion, 2025. URL https://arxiv.org/abs/2506.17298.

David A. Levin and Yuval Peres. *Markov Chains and Mixing Times*. American Mathematical Society, Providence, RI, 2 edition, 2017. doi: 10.1090/mbk/107. With contributions by Elizabeth L. Wilmer.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *International Conference on Learning Representations*, 2024.

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 *Advances in Neural Information Processing Systems*, volume 38, pp. 50608–50646, 2025.

Johan Oudinet, Alain Denise, and Marie-Claude Gaudel. A new dichotomic algorithm for the uniform random generation of words in regular languages. *Theoretical Computer Science*, 502:165–176, 2013. doi: 10.1016/j.tcs.2012.07.025.

Michael O Rabin and Dana Scott. Finite automata and their decision problems. *IBM journal of research and development*, 3(2):114–125, 1959.

Benjamin Rossman. Criticality and decision tree size of AC^0. Manuscript, 2017. URL `https://www.math.utoronto.ca/rossman/criticality-circuits.pdf`.

Marcel Paul Schützenberger. On finite monoids having only trivial subgroups. *Inf. Control.*, 8(2): 190–194, 1965.

Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

Howard Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Progress in Theoretical Computer Science. Birkhäuser, Boston, MA, 1994. doi: 10.1007/978-1-4612-0289-9.

Anej Svete and Ashish Sabharwal. On the reasoning abilities of masked diffusion language models. In *International Conference on Learning Representations*, 2026.

Adrien Ali Taïga, James Assiene, Daniele Calandriello, Rahma Chaabouni, João Gante, Tamara von Glehn, Nate Keating, Chris Knutsen, Martin Kukla, Tianlin Liu, Ivan Lobov, Ofir Nabati, João Gabriel Oliveira, Nicolas Perez-Nieves, Nastasia Prutianova, Bobak Shahriari, Jean Tarbouriech, Pavel Tyletski, Çağlar Ünlü, Cindy Wu, Glenn Cameron, Jerome Connor, Sertan Girgin, Maarten Grootendorst, Alon Levkovitch, Eliya Nachmani, Omar Sanseviero, Piotr Stanczyk, Quentin Berthet, Andrew Campbell, Clément Crepy, Valentin De Bortoli, Arnaud Doucet, Romuald Elie, Alexandre Galashov, Klaus Greff, Alexis Jacq, David Ruhe, Yu-Han Wu, Sebastian Flennerhag, Brendan O'Donoghue, George Scrivener, and Shantanu Thakoor. DiffusionGemma Technical Report, 2026. URL `https://arxiv.org/abs/2608.00146`.

Emanuele Viola. The complexity of distributions. *SIAM Journal on Computing*, 41(1):191–218, 2012. doi: 10.1137/100814998.

Emanuele Viola. Sampling lower bounds: Boolean average-case and permutations. *SIAM Journal on Computing*, 49(1):119–137, 2020. doi: 10.1137/18M1198405.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL `https://papers.nips.cc/paper_files/paper/2025/hash/d877ea0dd78bdbe54830670618c1de09-Abstract-Conference.html`.

Linxuan Wang, Ziyi Wang, Yikun Bai, Wei Deng, Guang Lin, and Qifan Song. Generalized discrete diffusion with self-correction. *arXiv preprint arXiv:2603.02230*, 2026.

A Sampling with Revision and Remasking

The two-round revision bound in theorem 3.1 rests on the DFA run sampler of proposition 3.2. Appendix A.1 first illustrates the block representation for modular counting, where transitions can be reassigned on the original state set. Appendix A.2 then proves lemma 3.5 for arbitrary DFAs by introducing hidden states, and appendix A.3 uses this representation to prove proposition 3.2. Finally, appendix A.4 proves the four-round remasking bound in theorem A.3.

A.1 Two-round revision sampling for modular counting

Theorem A.1 gives an explicit instance of lemma 3.5 for modular counting. It shows how state-dependent block bijections turn the residue DFA's cyclic transitions into order-preserving block maps while retaining the original states and block-transition counts. The $q = 5$ calculation makes these maps concrete before the proof implements the two-round sampler of theorem 3.1.

Theorem A.1 (Two-round revision sampling for MOD_q)**.** *Let $q \geq 2$ be an integer and let $L_q \coloneqq \{x \in \{0,1\}^* : |x|_1 \equiv 0 \pmod q\}$, where $|x|_1$ is the number of ones in x. The residue DFA $\delta(s,a) = s+a \pmod q$ on $Q = \mathbb{Z}_q$ admits an aperiodic block representation satisfying lemma 3.5 with $b = 2q$, $H = Q$, and $\phi = \mathrm{id}$. Consequently, for every $n \geq 1$, $\mathrm{Pair}_{L_q,n}$ has a randomized-AC^0 sampler using n fair bits, with depth $O_q(1)$ and size $n^{O_q(1)}$. A DLM with revision samples this distribution exactly in two rounds without CoT.*

Proof. The DFA has states $Q = \mathbb{Z}_q$, initial state 0, accepting set $\{0\}$, and transition $\delta(s,a) = s + a \pmod q$.

Block counts. Set $b = 2q$ and let $N^b(s,t)$ denote the number of length-b blocks taking state s to state t. With residue subscripts interpreted modulo q, define

$$c_d \coloneqq |\{w \in \{0,1\}^{2q} : |w|_1 \equiv d \pmod q\}|, \qquad N^b(s,t) = c_{t-s}. \tag{5}$$

The binomial counts are

$$c_0 = \binom{2q}{q} + 2, \qquad c_d = \binom{2q}{d} + \binom{2q}{q-d} \quad (1 \leq d < q). \tag{6}$$

They satisfy

$$c_d = c_{-d}, \qquad c_0 \geq c_1 \geq \cdots \geq c_{\lfloor q/2 \rfloor}. \tag{7}$$

To verify the inequalities, first note that $c_0 - c_1 = \frac{1}{q+1}\binom{2q}{q} + 2 - 2q \geq 0$. Indeed, the Catalan numbers satisfy $\mathrm{Cat}_1 = 1$ and $\mathrm{Cat}_q \,/\, \mathrm{Cat}_{q-1} = 2(2q-1)/(q+1) \geq 2$ for $q \geq 2$, so $\mathrm{Cat}_q \geq 2^{q-1} \geq 2q-2$. For $1 \leq r < \lfloor q/2 \rfloor$, put $\Delta_k = \binom{2q}{k} - \binom{2q}{k-1}$. Then $c_r - c_{r+1} = \Delta_{q-r} - \Delta_{r+1}$, and

$$\frac{\Delta_{q-r}}{\Delta_{r+1}} = \frac{(2r+1)(2q-r)}{(q+r+1)(2q-2r-1)} \prod_{j=1}^{q-2r-1} \frac{q+r+j}{r+1+j} \geq \frac{(2r+1)(2q-r)}{(r+2)(2q-2r-1)} > 1.$$

The lower bound keeps the first factor in the product and bounds every remaining factor by 1. This proves equation (7).

An order compatible with the counts. Order the states by the zigzag chain

$$\xi_0 \prec \xi_1 \prec \cdots \prec \xi_{q-1}, \qquad (\xi_0, \xi_1, \ldots) = (0, 1, -1, 2, -2, \ldots), \tag{8}$$

listing each residue once. For even q, the final state is $q/2$. Let $P_k \coloneqq \{\xi_0, \ldots, \xi_k\}$ and define

$$G_s(\xi_k) \coloneqq \sum_{t \in P_k} c_{t-s}. \tag{9}$$

We claim that for each adjacent pair $s \prec s'$ and every k,

$$G_s(\xi_k) \geq G_{s'}(\xi_k). \tag{10}$$

The prefixes are intervals of residues, $P_{2j} = \{-j, \ldots, j\}$ and $P_{2j+1} = \{-j, \ldots, j+1\}$. For an even prefix, symmetry and cancellation give

$$G_r(\xi_{2j}) = G_{-r}(\xi_{2j}), \qquad G_{-r}(\xi_{2j}) - G_{r+1}(\xi_{2j}) = c_{j-r} - c_{j+r+1} \geq 0.$$

The inequality follows from equation (7), since for every adjacent step $-r \prec r+1$ the valid indices satisfy $2\max\{j, r\} \leq q-1$, and hence $|j-r| \leq \min\{j+r+1, q-j-r-1\}$. For an odd prefix, reflection around $1/2$ and cancellation instead give

$$G_{-r}(\xi_{2j+1}) = G_{r+1}(\xi_{2j+1}), \qquad G_r(\xi_{2j+1}) - G_{-r}(\xi_{2j+1}) = c_{j+1-r} - c_{j+r+1} \geq 0.$$

Here an adjacent step $r \prec -r$ satisfies $2\max\{j+1, r\} \leq q$, so $|j+1-r| \leq \min\{j+r+1, q-j-r-1\}$. These two cases cover all adjacent pairs and prove equation (10).

Reassigning a common block rank. Give each block $u \in \{0,1\}^b$ its lexicographic rank $\rho(u) \in \{0, \ldots, 2^b - 1\}$, and set $G_s(\xi_{-1}) = 0$. Define the new transition by

$$\tau(s, u) = \xi_k \quad \Longleftrightarrow \quad G_s(\xi_{k-1}) \leq \rho(u) < G_s(\xi_k). \tag{11}$$

The same rank is used at every starting state. If $s \prec s'$ and $\tau(s', u) = \xi_k$, then $\rho(u) < G_{s'}(\xi_k) \leq G_s(\xi_k)$, so $\tau(s, u) \preceq \tau(s', u)$. Every block map therefore preserves the order, as does every composition. An order-preserving map g on a chain of q states has no cycle of length greater than one. Every orbit therefore reaches a fixed point within $q-1$ steps, so $g^{q-1} = g^q$. Thus the transition monoid is aperiodic.

For $q = 5$, we use blocks of length 10 and the order $0 \prec 1 \prec 4 \prec 2 \prec 3$. The residue counts are

$$(c_0, c_1, c_2, c_3, c_4) = (254, 220, 165, 165, 220).$$

Table 3 displays all interval endpoints in equation (11). For example, a block of rank 450 induces

$$(\tau(0, u), \tau(1, u), \tau(4, u), \tau(2, u), \tau(3, u)) = (1, 1, 4, 4, 4),$$

which is nondecreasing in the stated order.

Table 3: Cumulative block counts for MOD_5. Rows and columns follow $0 \prec 1 \prec 4 \prec 2 \prec 3$. Each column is nonincreasing down the rows, so a common rank defines an order-preserving transition.

Starting state s	$G_s(0)$	$G_s(1)$	$G_s(4)$	$G_s(2)$	$G_s(3)$
0	254	474	694	859	1024
1	220	474	639	859	1024
4	220	385	639	804	1024
2	165	385	550	804	1024
3	165	330	550	770	1024

Recovering the original transitions. For each starting state s and destination t, the two sets

$$\{u \in \{0,1\}^b : \tau(s, u) = t\}, \qquad \{w \in \{0,1\}^b : \delta^*(s, w) = t\}$$

both have size c_{t-s}. Match their elements in lexicographic order. Taking the union of these matchings over t gives a bijection $\pi_s : \{0,1\}^b \to \{0,1\}^b$ satisfying

$$\tau(s, u) = \delta^*(s, \pi_s(u)) = s + |\pi_s(u)|_1 \pmod q. \tag{12}$$

This is precisely equation (4) with $H = Q$ and $\phi = \mathrm{id}$. The bijection depends on s, so the construction reassigns block labels separately at each state. It preserves the transition counts from every state while changing the maps induced by individual labels.

For the block $u = 0111000010$ of rank 450 in the $q = 5$ example, this matching gives $\pi_0(u) = 1110100101$ and $\pi_4(u) = 0100101110$. The first decoded block has six ones and moves state 0 to 1. The second has five ones and leaves state 4 at 4, as prescribed by the same common-rank map above.

From the representation to two rounds. Write $n = bm + r$ with $0 \le r < b$ and partition the n fair bits into $u_1, \dots, u_m \in \{0,1\}^b$ and a tail $v \in \{0,1\}^r$. Compute the boundary states and decode the blocks by

$$h_0 = 0, \qquad h_j = \tau^*(0, u_1 \cdots u_j), \qquad w_j = \pi_{h_{j-1}}(u_j), \qquad x = w_1 \cdots w_m v.$$

By lemma 3.4, all h_j can be computed in parallel in AC^0, after which each w_j is a lookup. Equation (12) ensures that h_j is the residue of the decoded prefix $w_1 \cdots w_j$. The tail lookup gives the acceptance bit $\beta = \mathbf{1}[h_m + |v|_1 \equiv 0 \pmod q]$. Given $x = w_1 \cdots w_m v$, recover the seed successively from $h_0 = 0$ using $u_j = \pi_{h_{j-1}}^{-1}(w_j)$ and $h_j = \tau(h_{j-1}, u_j)$. The tail is unchanged, so the seed-to-word map is a bijection of $\{0,1\}^n$. Hence x is uniform and (x, β) has law $\mathrm{Pair}_{L_q,n}$.

Following the conversion in the proof of theorem 3.1, the first round reveals the n fair seed bits in the first n positions. The second round overwrites those positions with x and reveals β in the last position. The construction uses block length $2q$, q hidden states, n fair bits, two rounds, and no CoT positions. The local tables for τ and π_s have $q2^{2q}$ entries, each storing a destination state and a $2q$-bit output block. Together with the aperiodic prefix circuits, they give depth $O_q(1)$ and size $n^{O_q(1)}$. These circuit bounds treat q as independent of n. □

A.2 Aperiodic block representation

Theorem A.1 uses the symmetry of the residue counts to order the original states. For arbitrary DFAs, the proof of lemma 3.5 introduces hidden states to control the transition maps within each component.

Lemma 3.4 supplies parallel state evaluation for an aperiodic DFA. Lemma A.2 then gives the criterion we use to verify aperiodicity in the proof of lemma 3.5.

Lemma 3.4 (Restated)**.** *Let $\mathcal{B}$ be a DFA with state set H, alphabet Σ, transition function τ, and an aperiodic transition monoid. There is an AC^0 circuit family that, given a start state $s \in H$ and a word $u \in \Sigma^n$, outputs the state $\tau^*(s, u)$ reached from s after reading u.*

Proof of Lemma 3.4. For fixed states $s, t \in H$, the language of words taking s to t has an aperiodic syntactic monoid, since it is a quotient of the transition monoid of $\mathcal{B}$. By Barrington et al. (1992), this language belongs to AC^0. Evaluate these constantly many languages in parallel and select the result for the input start state s. Applying this construction to every prefix increases the circuit size by at most a factor of n and preserves constant depth. □

Lemma A.2. *A **nontrivial cycle** of a map is a directed cycle of length at least two. Let $g : H \to H$ be a self-map of a finite set. Then $g^k = g^{k+1}$ for some $k \ge 1$ if and only if g has no nontrivial cycle.*

Proof. If g contains a nontrivial cycle, then $g^m \ne g^{m+1}$ for every $m \ge 1$, since a point on the cycle has distinct images under the two maps. Conversely, if every cycle of g is a fixed point, then every orbit reaches a fixed point within $|H|$ steps, so $g^{|H|} = g^{|H|+1}$. Consequently a finite transformation monoid is aperiodic if and only if none of its elements has a nontrivial cycle. □

We now prove lemma 3.5. The construction treats terminal and transient components separately while preserving the block-transition counts under projection.

Lemma 3.5 (Restated)**.** *For a DFA $\mathcal{A} = (Q, \{0,1\}, \delta, q_0, F)$, there are a constant block length b, a hidden DFA $\mathcal{B}$ with state set H and transition function $\tau : H \times \{0,1\}^b \to H$, and a surjective projection $\phi : H \to Q$ such that*

1. *for every $h \in H$ there is a bijection $\pi_h : \{0,1\}^b \to \{0,1\}^b$ such that, for all $w \in \{0,1\}^b$,*

$$\phi\big(\tau(h, w)\big) = \delta^*(\phi(h), \pi_h(w)), \tag{4}$$

2. *the transition monoid of $\mathcal{B}$ is aperiodic.*

Proof. Define the transition-count matrix $\mathbf{N}(q, r) := |\{a \in \{0,1\} : \delta(q, a) = r\}|$, with powers taken over the integers, so that $\mathbf{N}^b(q, r) = |\{w \in \{0,1\}^b : \delta^*(q, w) = r\}|$.

The idempotent support. Let $\mathbf{P}_{\mathrm{bool}}(q,r) \coloneqq \mathbf{1}[\mathbf{N}(q,r) > 0]$, multiplied over the Boolean semiring, so that $\mathbf{P}^j_{\mathrm{bool}}(q,r) = 1$ exactly when some word of length j leads from q to r. Since Boolean matrices form a finite monoid under Boolean multiplication, there is $a \geq 1$ with $\mathbf{P}_{\mathrm{aper}} \coloneqq \mathbf{P}^a_{\mathrm{bool}}$ idempotent. We view it as a directed graph on Q, whose strongly connected components form an acyclic condensation graph. A component is **terminal** if no edge leaves it, and **transient** otherwise. A walk that leaves a transient component never returns to it, because the condensation graph is acyclic.

Fix the block length $b \coloneqq ac$, where c is a positive integer chosen large enough at the end of the proof.

Terminal components. Let $R = \{t_1, \ldots, t_k\}$ be a terminal component of the graph. In $\mathcal{B}$ this component corresponds to $2k$ hidden states, $H_R \coloneqq \{t_1^-, \ldots, t_k^-, t_1^+, \ldots, t_k^+\}$, ordered as $t_1^- < \cdots < t_k^- < t_1^+ < \cdots < t_k^+$. Write $h_1 < \cdots < h_{2k}$ for this order and let $\phi(t_i^\pm) = t_i$.

Define the matrix $\mathbf{N}_{\mathrm{hid}} : H_R \times H_R \to \mathbb{Z}_{\geq 0}$ by

$$\mathbf{N}_{\mathrm{hid}}(h_j, t_i^-) \coloneqq \left\lfloor \tfrac{2k+1-j}{2k+1}\, \mathbf{N}^b\big(\phi(h_j), t_i\big) \right\rfloor, \qquad \mathbf{N}_{\mathrm{hid}}(h_j, t_i^+) \coloneqq \left\lceil \tfrac{j}{2k+1}\, \mathbf{N}^b\big(\phi(h_j), t_i\big) \right\rceil \tag{13}$$

for $1 \leq j \leq 2k$ and $1 \leq i \leq k$.

The two entries over a state t_i sum to $\mathbf{N}^b(\phi(h_j), t_i)$, and block transitions from R remain in R, i.e., every word $w \in \{0,1\}^b$ maps a hidden state of H_R into H_R, so every row $h \in H_R$ sums to

$$\sum_{h' \in H_R} \mathbf{N}_{\mathrm{hid}}(h, h') = \sum_{t \in R} \mathbf{N}^b\big(\phi(h), t\big) = 2^b. \tag{14}$$

Write $G_h(x) \coloneqq \sum_{h' \leq x} \mathbf{N}_{\mathrm{hid}}(h, h')$ for the cumulative count of a row h, with $G_h(h_0) \coloneqq 0$. Order $\{0,1\}^b$ lexicographically, and let $\rho(w)$ be the rank of w. For $h \in H_R$ define $\tau(h, w) \coloneqq h_j$ when $G_h(h_{j-1}) \leq \rho(w) < G_h(h_j)$. By Equation (14) the intervals $[G_h(h_{j-1}), G_h(h_j))$ partition the ranks $\{0, 1, \ldots, 2^b - 1\}$, so τ is well defined on all of $\{0,1\}^b$.

For every $h \in H_R$ and $t_i \in R$, the number of blocks whose hidden destination projects to t_i is

$$|\{w : \phi(\tau(h, w)) = t_i\}| = \mathbf{N}_{\mathrm{hid}}(h, t_i^-) + \mathbf{N}_{\mathrm{hid}}(h, t_i^+) = \mathbf{N}^b(\phi(h), t_i).$$

This equals the number of blocks v with $\delta^*(\phi(h), v) = t_i$. Choose a bijection between these two sets for each t_i. Their unions partition $\{0,1\}^b$ on both sides, so the combined map is a bijection $\pi_h : \{0,1\}^b \to \{0,1\}^b$ satisfying Equation (4).

We now show that on the hidden states H_R no composition of the maps $\tau(\cdot, w)$ has a nontrivial cycle. To this end we show that $\mathbf{N}_{\mathrm{hid}}$ is order-compatible, that is, that for adjacent hidden states $h_j < h_{j+1}$ and every prefix x of the chain H_R, the cumulative count satisfies $G_{h_j}(x) \geq G_{h_{j+1}}(x)$. Order compatibility makes every $\tau(\cdot, w)$ order-preserving on H_R, and an order-preserving map has no nontrivial cycle.

For a prefix that ends at t_r^- with $1 \leq r \leq k$, Equation (13) gives

$$G_{h_j}(t_r^-) - G_{h_{j+1}}(t_r^-) = \sum_{i=1}^{r} \Big[\tfrac{2k+1-j}{2k+1} \mathbf{N}^b\big(\phi(h_j), t_i\big) - \tfrac{2k-j}{2k+1} \mathbf{N}^b\big(\phi(h_{j+1}), t_i\big) \Big] + O(k), \tag{15}$$

where the error term comes from the rounding of Equation (13). For a prefix ending at t_r^+ with $1 \leq r < k$, the row sums $G_h(t_k^+) = 2^b$ reduce the corresponding difference to $\sum_{i>r} \big[\mathbf{N}_{\mathrm{hid}}(h_{j+1}, t_i^+) - \mathbf{N}_{\mathrm{hid}}(h_j, t_i^+)\big]$, so that

$$G_{h_j}(t_r^+) - G_{h_{j+1}}(t_r^+) = \sum_{i=r+1}^{k} \Big[\tfrac{j+1}{2k+1} \mathbf{N}^b\big(\phi(h_{j+1}), t_i\big) - \tfrac{j}{2k+1} \mathbf{N}^b\big(\phi(h_j), t_i\big) \Big] + O(k). \tag{16}$$

Idempotency of $\mathbf{P}_{\mathrm{aper}}$ makes q and r mutually reachable, so $\mathbf{N}^a(q,r) \geq 1$ for all $q, r \in R$. The matrix $\mathbf{N}^a / 2^a$ restricted to R therefore has all positive entries and forms the transition matrix of an irreducible and aperiodic Markov chain on R. It converges to a stationary distribution π_R that is positive on every state of R (Levin & Peres, 2017, Theorem 4.9).

Since $\mathbf{N}^b/2^b = (\mathbf{N}^a/2^a)^c$ converges entrywise to π_R, the main terms of Equation (15) and Equation (16), divided by 2^b, satisfy

$$\begin{aligned} \frac{G_{h_j}(t_r^-) - G_{h_{j+1}}(t_r^-)}{2^b} &\longrightarrow \frac{1}{2k+1}\sum_{i=1}^{r}\pi_R(t_i) > 0, \\ \frac{G_{h_j}(t_r^+) - G_{h_{j+1}}(t_r^+)}{2^b} &\longrightarrow \frac{1}{2k+1}\sum_{i=r+1}^{k}\pi_R(t_i) > 0 \end{aligned} \tag{17}$$

as $c \to \infty$, with $1 \le r \le k$ in the first line and $1 \le r < k$ in the second. At the full prefix t_k^+, both cumulative counts equal 2^b. Since only finitely many prefixes are involved, some positive integer c_R satisfies $G_{h_j}(x) \ge G_{h_{j+1}}(x)$ for every adjacent pair and every prefix x, once $c \ge c_R$.

Order compatibility gives $G_{h_j} \ge G_{h_{j+1}}$ pointwise, so $G_{h_j}(h_\ell) \ge G_{h_{j+1}}(h_\ell)$ for every ℓ. If $\tau(h_{j+1}, w) = h_\ell$, then $\rho(w) < G_{h_{j+1}}(h_\ell) \le G_{h_j}(h_\ell)$, hence $\rho(w) < G_{h_j}(h_\ell)$ and therefore $\tau(h_j, w) \le h_\ell$. Thus $\tau(h_j, w) \le \tau(h_{j+1}, w)$ for every $w \in \{0,1\}^b$, so every composition of the maps $\tau(\cdot, w)$ preserves the order $h_1 < \cdots < h_{2k}$ on H_R. An order-preserving self-map g of a finite chain has no nontrivial cycle: if z is the least element of such a cycle and z' its predecessor in the cycle, then $z' > z$ and $z = g(z') \ge g(z)$, while $g(z) \ne z$ would give $g(z) > z$, a contradiction.

Transient components. Let C be a transient component of the graph. In $\mathcal{B}$ this component corresponds to $|C|$ hidden states $H_C := \{h_q : q \in C\}$, with $\phi(h_q) = q$. Transience gives $\sum_{q \in C} \mathbf{N}^b(q, C)/2^b \to 0$ as $c \to \infty$, so there is a positive integer c_C such that for every $c \ge c_C$ the internal mass satisfies $S_C := \sum_{q,t \in C} \mathbf{N}^b(q,t) < 2^b$.

For $q \in C$ put $W_q := \{w \in \{0,1\}^b : \delta^*(q,w) \in C\}$, so that $|W_q| = \sum_{t \in C} \mathbf{N}^b(q,t)$ and $\sum_{q \in C} |W_q| = S_C < 2^b$. Choose sets $S_q \subseteq \{0,1\}^b$ with $|S_q| = |W_q|$, one for each $q \in C$, that are pairwise disjoint across distinct q, and a bijection π_{h_q} of $\{0,1\}^b$ with $\pi_{h_q}(S_q) = W_q$. Define

$$\tau(h_q, w) := \begin{cases} t^+ & \text{if } \delta^*(q, \pi_{h_q}(w)) = t \text{ for a terminal } t, \\ h_t & \text{if } \delta^*(q, \pi_{h_q}(w)) = t \text{ for a transient } t. \end{cases}$$

Then $\phi(\tau(h_q, w)) = \delta^*(q, \pi_{h_q}(w))$ for every w, which is Equation (4), and

$$\tau(h_q, w) \in H_C \iff \delta^*\big(q, \pi_{h_q}(w)\big) \in C \iff w \in S_q. \tag{18}$$

We claim that if a composition $g = \tau(\cdot, w_m) \circ \cdots \circ \tau(\cdot, w_1)$ maps some hidden state $x_0 \in H_C$ into H_C, then the walk $x_0, \ldots, x_m$ with $x_i := \tau(x_{i-1}, w_i)$ is determined by $(w_1, \ldots, w_m)$. Since a walk in Q that leaves the component C never returns, the whole walk $\phi(x_0), \ldots, \phi(x_m)$ lies in C. For a block w, at most one pair (h_q, h_t) of states of H_C satisfies $\tau(h_q, w) = h_t$. Indeed, by Equation (18) the target of $\tau(h_q, w)$ lies in H_C only when $w \in S_q$, and the sets S_q are disjoint across distinct q, so at most one q qualifies, and $\tau(h_q, w)$ is then a single state. Hence each step (x_{i-1}, x_i) of the walk is the unique pair for the word w_i, and x_0, along with the whole walk, is determined by $(w_1, \ldots, w_m)$. So at most one hidden state of H_C has its image in H_C under g. In particular, g has no nontrivial cycle in H_C, since the states of such a cycle are distinct and each of them maps to another state of H_C.

Aperiodicity. There are finitely many components, so fix c large enough that $c \ge c_R$ and $c \ge c_C$ for all of them, and let $b := ac$. A nontrivial cycle of a composition of the maps $\tau(\cdot, w)$ must lie entirely inside one component of the acyclic condensation graph. On every component we have shown that no composition of these maps has a nontrivial cycle inside it. Hence no composition has a nontrivial cycle, and by Lemma A.2 the monoid generated by the maps $\tau(\cdot, w)$ is aperiodic. □

A.3 DFA RUN SAMPLING

We prove proposition 3.2 by combining the block representation of lemma 3.5 with the parallel evaluation in lemma 3.4. Algorithm 2 computes the hidden boundary states and decodes the blocks using the corresponding bijections. The resulting sampler is the circuit used in the second round of theorem 3.1.

Proposition 3.2 (Restated). *For every DFA $\mathcal{A}$ over $\{0,1\}$, there is a randomized-AC^0 circuit family $(C_n)_{n\geq 1}$ such that each circuit C_n takes n fair bits and exactly samples the run of $\mathcal{A}$ on a uniform word $X \sim \mathrm{Unif}(\{0,1\}^n)$.*

Proof of Proposition 3.2. Apply Lemma 3.5 with its block length b, write $n = bm+r$ with $0 \leq r < b$, and fix a state $\iota(q) \in \phi^{-1}(q)$ for every $q \in Q$. The sampler is Algorithm 2. The algorithm first draws n fair bits and partitions them into m hidden blocks $u_1, \ldots, u_m \in \{0,1\}^b$ and a tail $v \in \{0,1\}^r$. It then computes, in parallel for every $j \geq 1$, the hidden state $h_j := \tau^*(\iota(q_0), u_1 \cdots u_j)$ that the aperiodic hidden DFA $\mathcal{B}$ reaches after the first j blocks. Given these states, each hidden block is decoded independently. The bijection $\pi_{h_{j-1}}$ projects u_j to the block w_j, and running δ on w_j from the boundary state $y_{(j-1)b} = \phi(h_{j-1})$ fills the remaining positions. The tail is passed through unchanged, and δ advances the trajectory through it. The output is the word $w_1 \cdots w_m v$ together with the state trajectory $(y_0, y_1, \ldots, y_n)$.

Algorithm 2 Sampling a DFA run

Input: blocks $u_1, \ldots, u_m \in \{0,1\}^b$ and tail $v = (v_1, \ldots, v_r) \in \{0,1\}^r$
$h_0 \leftarrow \iota(q_0)$ $\quad$ $y_0 \leftarrow q_0$
for all $j = 1, \ldots, m$ **in parallel do**
    $h_j \leftarrow \tau(\cdot, u_j) \circ \cdots \circ \tau(\cdot, u_1)(h_0)$
for all $j = 1, \ldots, m$ **in parallel do**
    $y_{(j-1)b} \leftarrow \phi(h_{j-1})$ $\quad$ $w_j \leftarrow \pi_{h_{j-1}}(u_j)$ $\quad$ ▷ projected block
    for $i = 1, \ldots, b$ **do**
        $y_{(j-1)b+i} \leftarrow \delta\big(y_{(j-1)b+i-1},\, w_{j,i}\big)$
for $i = 1, \ldots, r$ **do**
    $y_{bm+i} \leftarrow \delta(y_{bm+i-1}, v_i)$
return $w_1 \cdots w_m v$ and $(y_0, y_1, \ldots, y_n)$

Correctness. Given the projected blocks, the source blocks are uniquely recovered from h_0 by $u_j = \pi_{h_{j-1}}^{-1}(w_j)$ and $h_j = \tau(h_{j-1}, u_j)$. The tail is unchanged. Thus the map from the n fair bits to the output word is a bijection of $\{0,1\}^n$, so the word is jointly uniform. Equation (4) makes the boundary states of consecutive blocks agree, and each block and the tail follow δ. The output is therefore the complete run on a uniform word. When $m = 0$, the construction consists only of the tail lookup from q_0.

Complexity. The sampler uses exactly n fair bits, one per input bit. The hidden states $h_j = \tau^*(h_0, u_1 \cdots u_j)$ are prefix states of the aperiodic hidden DFA $\mathcal{B}$, so by lemma 3.4 they are computable in AC^0. The block-boundary states are $\phi(h_j)$. The bijections π_h, the projection ϕ, the map ι, and the transition function δ are constant-size tables, so the blocks and tail are decoded in parallel. Thus the sampler is randomized-AC^0.

Finally, no sampler can use fewer than n fair bits: the run determines the input word X, whose entropy is n, while the output of a sampler with r fair bits has entropy at most r. □

Theorem 3.1 (Restated). *For every binary regular language L, there is a DLM family $(S_n)_{n\geq 1}$ with $\rho = \circlearrowleft$ that samples $\mathrm{Pair}_{L,n}$ without CoT and uses $D_n = 2$ decoding rounds for every n. Consequently, $R^{\circlearrowleft}(\mathrm{Pair}_L) \in O(1)$.*

Proof of Theorem 3.1. Let $\mathcal{A} = (Q, \{0,1\}, \delta, q_0, F)$ recognize L, and let $(C_n)_{n\geq 1}$ be the circuit family given by Proposition 3.2. We construct S_n from C_n. In the first round, the policy selects the first n positions and their predictors reveal independent fair bits, which form the random input to C_n. In the second round, the policy selects all $n+1$ positions. The predictors overwrite the first n positions with the word x in the run produced by C_n, and the predictor at the last position outputs $\mathbf{1}[y_n \in F]$. The predictors compute these outputs deterministically in AC^0, using the tokens revealed in the first round as the fair-bit inputs to C_n. The terminal sequence therefore has law $\mathrm{Pair}_{L,n}$. □

A.4 Four-round sampling with remasking

Lemma A.4 implements a circuit sampler through remasking, using mask positions to carry the intermediate information between rounds. Applying it to the DFA run sampler of proposition 3.2 gives the four-round bound in theorem A.3.

Theorem A.3. *For every binary regular language L, there is a DLM family with remasking that samples $\mathrm{Pair}_{L,n}$ exactly in four rounds using three binary CoT positions.*

Lemma A.4. *Use the remasking model of Theorem A.3. In each round, the policy selects revealed positions to remask and masked positions to decode. All choices and predictor outputs depend on the sequence at the start of the round, and updates occur simultaneously. The policy and predictors satisfy the circuit restrictions in Section 2.1. Let (G_n) be an AC^0 circuit family mapping r_n bits to m_n bits. The distribution $G_n(U)$, where $U \sim \mathrm{Unif}(\{0,1\}^{r_n})$, can be sampled by a DLM with remasking in four rounds using $\max\{0, r_n - m_n\} + 3$ CoT positions.*

Proof. Use $\max\{0, r_n - m_n\}$ CoT positions together with the output positions to hold the seed, and reserve three further CoT positions as phase markers. In round one, sample r_n independent fair bits in predetermined positions and fill any remaining non-marker positions with zeros. In round two, compute $Y = G_n(U)$ and remask precisely the output positions with $Y_i = 0$. The output mask pattern now determines Y. In round three, fill the masked output positions with zeros and remask all previously revealed output positions. In round four, fill the masked output positions with ones. The output is Y.

Reveal one phase marker as zero in each of the first three rounds and leave revealed markers unchanged. Their mask pattern distinguishes the four rounds using only the current sequence. All positions are revealed at termination. The policy evaluates G_n only in round two, and the predictors only sample fair bits or emit constants, so both satisfy the required circuit bounds. □

Proof of Theorem A.3. Apply Lemma A.4 to the DFA-run sampler followed by the acceptance-bit map in Theorem 3.1. It uses $r_n = n$ fair bits and outputs $m_n = n + 1$ bits, so three CoT positions suffice. □

Corollary A.5. *For every binary regular language L, a DLM with remasking samples $\mathrm{Pair}_{L,n}$ in four rounds using three CoT positions. For every $0 < \epsilon \le 1/2$, a DLM with remasking ϵ-approximately samples $\mathrm{Word}_{L,n}$ in four rounds using $O_L(\log(1/\epsilon))$ CoT positions, with circuit size $\mathrm{poly}_L(n, 1/\epsilon)$ and depth $O_L(1)$.*

Proof. Apply Lemma A.4 to the deterministic maps evaluated in the second round of the constructions for Theorems 3.1 and 5.1. Their seed lengths are n and $n+O_L(\log(1/\epsilon))$, respectively. The simulation preserves their output distributions. □

B Sampling without Revision

B.1 Circuit lower-bound tools

For $z \in \{0,1\}^n$ and $q \ge 2$, write $\mathrm{MOD}_q(z) := \mathbf{1}[\sum_i z_i \equiv 0 \pmod q]$.

Lemma B.1. *Fix $q \ge 2$. Let $(C_n)_{n\ge 1}$ be a polynomial-size family of circuits with unbounded-fan-in AND gates and NOT gates such that C_n computes MOD_q on n bits. Then, for every sufficiently large n,*

$$\mathrm{depth}(C_n) = \Omega\left(\frac{\log n}{\log\log n}\right).$$

Proof. For a symmetric Boolean function f on n bits, let $s(f) = (s_0, \dots, s_n)$ be its spectrum, where s_i is the value of f on inputs of Hamming weight i. Its boundary $b(f)$ is the smallest value such that $s_{\lfloor b(f)\rfloor} = \cdots = s_{\lfloor n-b(f)\rfloor}$. With this definition, the bound stated in Husfeldt (1997, Chapter 3, Sections 1 and 3.3) implies that any size-S, depth-d, unbounded-fan-in Boolean circuit computing f satisfies

$$d = \Omega\left(\frac{\log b(f)}{\log\log S}\right).$$

For $f = \mathrm{MOD}_q$, the spectrum is $s_i = \mathbf{1}[i \equiv 0 \pmod q]$. Every constant run in this spectrum has length at most $q-1$. Hence its central constant segment has at most $q-1$ entries, which gives $b(f) = \Theta(n)$ for fixed q. Finally, the polynomial-size assumption gives $S = n^{O(1)}$, and the claimed bound follows. □

Lemma B.2. *Let L be a binary regular language outside AC^0. Then there are an integer $\ell \geq 1$, a modulus $q \geq 2$, a block substitution h that replaces every bit of $z \in \{0,1\}^m$ by one of two fixed words of length ℓ, and $t = O_L(1)$ pairs $(a_j, b_j) \in \{0,1\}^* \times \{0,1\}^*$ with signs $\sigma_j \in \{0,1\}$, such that for every $z \in \{0,1\}^m$,*

$$\mathrm{MOD}_q(z) = 1 \quad \Longleftrightarrow \quad f_L\big(a_j h(z) b_j\big) = \sigma_j \text{ for every } j \in [t], \tag{19}$$

where $|h(z)| = \ell m$.

Proof. Let $\mu : \{0,1\}^* \to M$ be the syntactic morphism of L, so that $\mu(s) = \mu(s')$ holds exactly when $asb \in L \iff as'b \in L$ for all contexts (a, b). The stable-image characterization of Barrington et al. (1992) implies that the stable image of μ is not aperiodic. It therefore contains a nontrivial cyclic subgroup with identity e and a generator g of order $q \geq 2$. Choose a stability index ℓ for μ. Both e and g lie in $\mu(\{0,1\}^\ell)$, so fix words $v, u \in \{0,1\}^\ell$ with $\mu(v) = e$ and $\mu(u) = g$, and let $h(z)$ replace each 0 by v and each 1 by u. Since the subgroup is cyclic and commutative, $\mu(h(z)) = g^{|z|_1}$, so $\mu(h(z)) = e$ exactly when $\mathrm{MOD}_q(z) = 1$.

It remains to test the syntactic class of e. For every monoid element $r \neq e$, syntactic distinctness provides a context (a_r, b_r) that distinguishes r from e. Let σ_r be the membership value of the e class in this context. These $|M| - 1 = O_L(1)$ tests give, for every $s \in \{0,1\}^*$,

$$\mu(s) = e \quad \Longleftrightarrow \quad f_L(a_r s b_r) = \sigma_r \text{ for every } r \neq e.$$

The forward direction follows from syntactic equivalence. For the backward direction, use the test indexed by $r = \mu(s)$. Substituting $s = h(z)$ proves the claim. □

Lemma B.3. *Let L be a binary regular language outside AC^0. If a polynomial-size family C_m with unbounded-fan-in AND gates and NOT gates computes f_L, then*

$$\max_{0 \leq m \leq n} \mathrm{depth}(C_m) = \Omega\left(\frac{\log n}{\log \log n}\right).$$

Proof. Apply Lemma B.2. On input $z \in \{0,1\}^m$, substitute the literals forming $a_j h(z) b_j$ into $C_{\ell m + |a_j| + |b_j|}$, complement its output when $\sigma_j = 0$, and take the conjunction over $j \in [t]$. The result is a polynomial-size circuit for MOD_q whose depth is

$$\max_{\delta \in \Delta} \mathrm{depth}(C_{\ell m + \delta}) + O_L(1), \qquad \Delta \coloneqq \{|a_j| + |b_j| : j \in [t]\}.$$

The conclusion follows from Lemma B.1 after rescaling the input length by the constant ℓ and absorbing the finite offsets. □

Lemma B.4. *Let C belong to an AC^0 circuit family and have m input bits and depth at most a fixed $D \geq 2$. Suppose that $C^{-1}(1) \subseteq \mathrm{MOD}_q^{-1}(1)$ for a fixed modulus $q \geq 2$. Then*

$$\Pr\big[C(U) = 1\big] \leq 2^{-\Omega_D\left(m/(\log m)^{D-1}\right)}. \tag{20}$$

The implicit constant may also depend on the polynomial size exponent.

Proof. Let S be the size of C. By Rossman (2017, Theorem 6.1), C is p-critical for $p = 1/(c_D(\log(2+S))^{D-1})$. The theorem excludes NOT gates from its depth count, so its depth parameter is at most D. Proposition 5.2 of the same work gives a decision tree T computing C with $O(2^{(1-p)m})$ leaves.

Every accepting leaf of T fixes all input bits. Otherwise, flipping an unqueried bit would preserve acceptance while changing the Hamming weight by 1 or -1, contradicting $C^{-1}(1) \subseteq \mathrm{MOD}_q^{-1}(1)$. Hence

$$\Pr[C(U) = 1] \leq \frac{O(2^{(1-p)m})}{2^m} = 2^{-\Omega_D\left(m/(\log m)^{D-1}\right)}.$$

□

Lemma B.5. *Let L be a binary regular language outside AC^0 and let $k : \mathbb{N}_{\geq 1} \to \mathbb{N}$. Suppose that for every $n \geq 1$ and $b \in \{0,1\}^{k(n)}$, there is a circuit with unbounded-fan-in AND gates and NOT gates T_b with input $z = (x, c) \in \{0,1\}^{n+1}$, size polynomial in $n + k(n)$, and depth at most a constant d. Suppose also that*

$$f_L(x) = c \quad \iff \quad T_b(x, c) = 1 \text{ for some } b \in \{0,1\}^{k(n)}. \tag{21}$$

Define

$$K(n) \coloneqq \max_{1 \leq r \leq n} k(r).$$

Then there is a constant $C > 0$ such that

$$K(n) = \Omega\left(\frac{n}{(\log n)^C}\right).$$

Proof. Let $z \in \{0,1\}^m$, so $m = |z|$. Apply Lemma B.2, which provides a block length ℓ, the block substitution h, the modulus q, and $t = O_L(1)$ context tests (a_j, b_j, σ_j) with $\mathrm{MOD}_q(z) = 1 \iff f_L(a_j h(z) b_j) = \sigma_j$ for all j. Each input bit is replaced by a word of length ℓ, so $|h(z)| = \ell m$ and $|a_j h(z) b_j| = \ell m + |a_j| + |b_j|$. Choose one transcript $y_j \in \{0,1\}^{k(\ell m + |a_j| + |b_j|)}$ per context test and define

$$G_{(y_1,\ldots,y_t)}(z) \;\coloneqq\; \bigwedge_{j \in [t]} T_{y_j}\big(a_j h(z) b_j,\ \sigma_j\big).$$

Let $M \coloneqq \max_j k(\ell m + |a_j| + |b_j|)$, the largest CoT budget among the finitely many lengths appearing in the context tests. If $M \geq m$, the required bound below already holds. We may therefore assume $M < m$. Each $G_{(y_1,\ldots,y_t)}$ is then a polynomial-size circuit in m of depth $d + O(1)$. The bits of $h(z)$ are the literals z_i and $\neg z_i$, and the context bits are hardwired. If $G_{(y_1,\ldots,y_t)}(z) = 1$, then each T_{y_j} is satisfied, so $\sigma_j = f_L(a_j h(z) b_j)$ for every j and Equation (19) gives $\mathrm{MOD}_q(z) = 1$. Conversely, if $\mathrm{MOD}_q(z) = 1$, every context test holds, and Equation (21) provides a transcript y_j whose circuit accepts the corresponding correctly labeled word. Hence at most 2^{tM} circuits of depth $d + O(1)$ form a one-sided cover of $\mathrm{MOD}_q^{-1}(1)$.

By Lemma B.4, each circuit in this cover accepts at most $2^{m - \Omega_D(m/(\log m)^{D-1})}$ inputs, where $D = d + O(1)$. On the other hand, MOD_q has at least $\beta_q 2^m$ positive instances for a constant $\beta_q > 0$ and all sufficiently large m. The union bound gives

$$\beta_q 2^m \leq 2^{tM}\, 2^{m - \Omega_D(m/(\log m)^{D-1})}.$$

Taking base-two logarithms and using that t is constant yields

$$M \;\geq\; c_L \frac{m}{(\log m)^{\,d + O_L(1)}},$$

where M is the maximum of k over the lengths $\ell m + \delta$ for $\delta \in \Delta_L \coloneqq \{|a_j| + |b_j|\}$. For every sufficiently large n, choose $m = \lfloor (n - \max \Delta_L)/\ell \rfloor$. These lengths are all at most n, so $K(n) \geq M$, while $m = \Theta_L(n)$. Absorbing the constant rescaling and the fixed exponent into the asymptotic notation and C proves the claim. □

B.2 Tight round complexity

Theorem 4.3 (Restated)**.** *For a binary regular language L outside AC^0, sampling Pair_L without revision with a constant CoT budget has round complexity $R^{\times}(\mathrm{Pair}_L) \in \Omega(\log n / \log\log n)$.*

Proof. For a constant CoT budget, the circuits in Lemma 4.2 can be combined into a polynomial-size Boolean circuit family of depth $O(D_n)$ recognizing L. Applying the regular-language depth lower bound (Lemma B.3) yields the claimed round lower bound. □

Theorem 4.4 (Restated)**.** *For every binary regular language L, there is a DLM family without revision that samples $\mathrm{Pair}_{L,n}$ without CoT in $O(\log n / \log\log n)$ decoding rounds for every n. Consequently, the round complexity satisfies $R^{\times}(\mathrm{Pair}_L) \in O(\log n / \log\log n)$.*

Proof. Fix a DFA $\mathcal{A} = (Q, \{0,1\}, \delta, q_0, F)$ recognizing L, and use the transformations T_u of Equation (3). With $A \coloneqq \{T_u : \delta^*(q_0, u) \in F\}$ we have $f_L(x) = \mathbf{1}[T_x \in A]$. Enumerate $M_{\mathcal{A}} = \{a_1, \ldots, a_K\}$; K is a constant depending only on L.

Layout. Let $B := \lceil \log n \rceil$. Add a unary super-root $r_\star$ whose unique child is the root r of an ordered B-ary tree. For every edge $v \to c$, the local word x^v of the parent contains a marker slot $m_c = (q_1^c, \ldots, q_K^c)$ of K ordinary input positions. The remaining positions are placed in leaf words of length $O(BK)$. The local words partition the first n positions and appear in preorder, so the word of the subtree rooted at v is x^v followed by its child subtrees in order. A balanced recursive partition gives height $h = O(\log_B n) = O(\log n / \log\log n)$. The finitely many smaller lengths are handled directly.

Subtree summaries. For a node v with ordered children $c_1, \ldots, c_d$, define

$$\sigma(v) := \tau(c_d) \circ \cdots \circ \tau(c_1), \qquad \tau(v) := \sigma(v) \circ T_{x^v}, \tag{22}$$

where the empty composition is the identity of $M_{\mathcal{A}}$. Thus $\tau(v)$ is the monoid value of the whole subtree rooted at v. The slot m_v in v's parent stores $\sigma(v) = a_j$ by the one-mask convention

$$q_j^v = \mathtt{M}, \qquad q_\ell^v \neq \mathtt{M} \quad \text{for every } \ell \neq j.$$

The slot lies outside v's subtree. Its sampled bits therefore belong to the parent's local word rather than to the summary it currently stores.

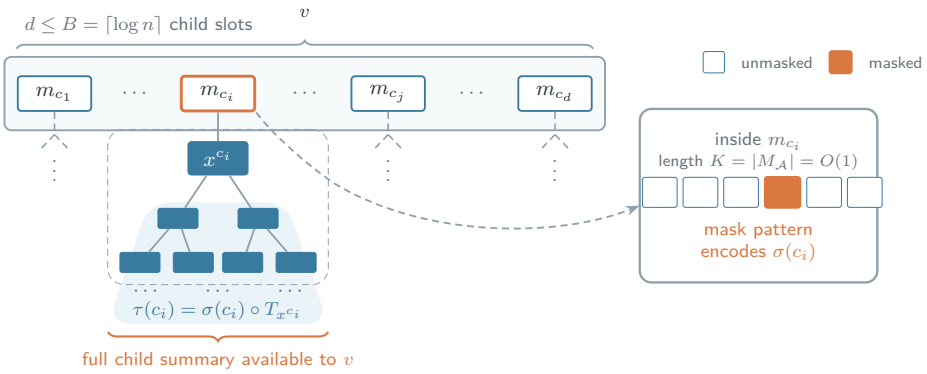

Figure 1: One aggregation step in the no-revision construction. For a child c_i of v, the one-mask pattern in m_{c_i} encodes $\sigma(c_i)$, while the visible local word x^{c_i} and Equation (22) determine the full child summary $\tau(c_i)$.

Rounds. Initially, every leaf is ready. More generally, a node v is ready when each x^{c_i} is visible and each m_{c_i} has one mask. The policy decodes $\sigma(c_i)$ from that mask, obtains $\tau(c_i)$ from Equation (22), and computes $\sigma(v)$. In the same round it creates the one-mask pattern for $\sigma(v)$ in m_v, reveals the remaining mask in every m_{c_i}, and reveals any leaf positions in x^v. All newly revealed input positions are sampled as independent fair bits. After the round, x^v is visible and m_v encodes $\sigma(v)$, which is the same invariant required by v's parent.

The computation at a node is in AC^0. Each local word has $O(BK) = O(\log n)$ bits, so its monoid value is obtained by a polynomial-size table lookup. The product of at most B child summaries uses a second table with $|M_{\mathcal{A}}|^B = n^{O(1)}$ entries. All ready nodes at one level are processed in parallel. The readiness rule depends only on the current mask pattern, so the policy does not need the round index. After h rounds, the ordinary root r has been processed. Its local word x^r is visible, and the unique slot m_r in the unary super-root has one mask and encodes $\sigma(r)$. Hence the current sequence determines $\tau(r)$.

Terminal rounds. Let U be the hidden bit in m_r, let $m_r(u)$ denote its completion by $u \in \{0,1\}$, and define

$$g(u) := \mathbf{1}\big[\tau(r) \circ T_{m_r(u)} \in A\big].$$

This composition respects the input order because the super-root slot precedes its unique child subtree. Round $h+1$ samples the label Y according to

$$\Pr\big[Y = 1 \mid \text{current sequence}\big] = \tfrac{1}{2} \sum_{u \in \{0,1\}} g(u) \in \{0, \tfrac{1}{2}, 1\}, \tag{23}$$

and round $h+2$ reveals U from the posterior $\Pr[U = u \mid Y = b] \propto \frac{1}{2}\mathbf{1}[g(u) = b]$. Therefore

$$\Pr\big[Y = b, U = u \mid \text{current visible bits}\big] = \tfrac{1}{2}\mathbf{1}[g(u) = b].$$

Thus X is uniform on $\{0,1\}^n$ and $Y = f_L(X)$. The output law is $\mathrm{Pair}_{L,n}$, and the construction uses $h + 2 = O(\log n / \log\log n)$ rounds without revision. □

B.3 CoT COMPLEXITY

Theorem 4.6 (Restated)**.** *For a binary regular language $L \notin \mathsf{AC}^0$, consider a DLM family S without revision that exactly samples $\mathrm{Pair}_{L,n}$ and has $R_S(n) \in O(1)$. There is a fixed integer $i_S \geq 0$ such that*

$$K_S(n) = \Omega_S\left(\frac{n}{(\log n)^{i_S}}\right).$$

In particular, S does not have CoT complexity $O(n^{1-\theta})$ for any constant $\theta > 0$.

Proof of Theorem 4.6. Apply Lemma 4.2. The resulting circuits satisfy the assumptions of Lemma B.5, with $K(n) = K_S(n)$. The lemma gives a constant $C > 0$ such that

$$K_S(n) = \Omega_S\left(\frac{n}{(\log n)^{C}}\right).$$

Taking $i_S = \lceil C \rceil$ proves the claim. □

Lemma B.6. *Let $\mathcal{A} = (Q, \{0,1\}, \delta, q_0, F)$ be a DFA with transition monoid $M_{\mathcal{A}}$. Let $\mathbf{P}_{\mathrm{bool}}$ and a be as in the proof of Lemma 3.5, so that $\mathbf{P}^a_{\mathrm{bool}}$ is idempotent. Define the set of transformations realized at positive multiples of this length by*

$$M^{(a)}_{\mathcal{A}} \coloneqq \{T_w : w \in \{0,1\}^{ka} \text{ for some } k \geq 1\}.$$

There are a block length $b = ac$ for some constant $c \geq 1$, a finite state space H, a surjective projection $\phi : H \to Q$, and a DFA

$$\mathcal{B} = (H, \{0,1\}^b \times M^{(a)}_{\mathcal{A}}, \tau)$$

with the following properties:

(i) for every $h \in H$ and $g \in M^{(a)}_{\mathcal{A}}$, there is a bijection $\pi_{h,g} : \{0,1\}^b \to \{0,1\}^b$ such that, for every $u \in \{0,1\}^b$,

$$\phi\big(\tau(h,(u,g))\big) = \delta^*\big(g(\phi(h)), \pi_{h,g}(u)\big); \tag{24}$$

(ii) the transition monoid of $\mathcal{B}$ is aperiodic.

Proof. Use the notation and hidden states from the proof of Lemma 3.5, and set $b = ac$. For $g \in M^{(a)}_{\mathcal{A}}$, define

$$\mathbf{R}_g(q,r) \coloneqq \mathbf{N}^b(g(q), r).$$

Since g is induced by a word of length ka, $\mathbf{P}^{ka}_{\mathrm{bool}} = \mathbf{P}_{\mathrm{aper}}$. Thus g preserves every terminal component and cannot move backward in the condensation graph.

On a terminal component R, the distribution $\mathbf{R}_g(q,\cdot)/2^b$ converges, as $c \to \infty$, to the same stationary distribution π_R as in Lemma 3.5; the limit is independent of q and g. Hence the same ordered-census construction, with $\mathbf{N}^b(q,r)$ replaced by $\mathbf{R}_g(q,r)$, is order-compatible for all sufficiently large c.

On a transient component C,

$$2^{-b} \sum_{q,t \in C} \mathbf{R}_g(q,t) \longrightarrow 0.$$

Hence, for large enough c, the internal census is smaller than 2^b and can be assigned disjoint seed sets exactly as in Lemma 3.5. Because there are only finitely many components and controls g, one value of c works simultaneously for all of them. The resulting maps are order-preserving on terminal

components and have at most one internal edge on transient components, so every mixed composition has no nontrivial cycle. By Lemma A.2, the transition monoid of $\mathcal{B}$ is aperiodic.

Finally, for fixed h, g, r, the hidden census and the set $\{v : \delta^*(g(\phi(h)), v) = r\}$ both have size $\mathbf{R}_g(\phi(h), r)$. Matching them in a fixed order for every r gives the bijection $\pi_{h,g}$ and Equation (24). □

Lemma B.7. *Let $\mathcal{A} = (Q, \{0,1\}, \delta, q_0, F)$ be a DFA and fix an integer $i \geq 0$. For every sufficiently large n, every $u \in \{0,1\}^n$ admits a decomposition*

$$\begin{aligned} &u = A_1 U_1 \cdots A_m U_m R \in \{0,1\}^n, \\ &|U_j| = O_{\mathcal{A}}(1), \qquad |A_j| = \Theta_{\mathcal{A}}((\log n)^i), \quad |R| = O_{\mathcal{A}}((\log n)^i + 1). \end{aligned} \tag{25}$$

when $i \geq 1$; for $i = 0$, take every A_j to be empty. There are functions

$$f_{n,i} : \{0,1\}^n \to \{0,1\}^n, \qquad s_{n,i} : \{0,1\}^n \to Q$$

such that:

- *(i) $f_{n,i}$ is a bijection;*
- *(ii) $f_{n,i}(u) = A_1 V_1 \cdots A_m V_m R$, that it agrees with u on every position in the blocks $A_1, \ldots, A_m$ and in the tail R;*
- *(iii) $s_{n,i}(u) = \delta^*(q_0, f_{n,i}(u))$.*

For every fixed i, the families $\{f_{n,i}\}_n$ and $\{s_{n,i}\}_n$ are computable by AC^0 circuit families. For $i = 0$, this is the all-short-block reparameterization used in Proposition 3.2.

Proof. The case $i = 0$ is the bijection implicit in the proof of Proposition 3.2: its source blocks are the fair seeds, its image is the projected input word, and the construction computes the entire DFA run in AC^0.

Now let $i \geq 1$ and apply Lemma B.6. Choose B to be the largest positive multiple of a below $(\log n)^i$. Then $B = \Theta((\log n)^i)$ and, for every long block A_j, $g_j := T_{A_j} \in M_{\mathcal{A}}^{(a)}$.

Use the decomposition in Equation (25), with $|U_j| = b$ and $|R| < B + b$. Starting from a fixed $h_0 \in \phi^{-1}(q_0)$, set

$$h_j = \tau(h_{j-1}, (U_j, g_j)), \qquad V_j = \pi_{h_{j-1}, g_j}(U_j),$$

and define $f_{n,i}(u) = A_1 V_1 \cdots A_m V_m R$. For fixed long blocks, this map is a bijection: once the preceding seeds determine h_{j-1}, the jth short block is transformed by a permutation. By definition, it fixes every long block and the tail. Equation (24) gives

$$\delta^*(q_0, f_{n,i}(u)) = \delta^*(\phi(h_m), R).$$

It remains to check the circuit depth. Each T_{A_j} is computable by an i-level product tree over the fixed monoid $M_{\mathcal{A}}$: a node multiplies $O(\log n)$ monoid elements by an AC^0 table lookup. The aperiodicity above then makes all prefix hidden states h_j computable in AC^0. The blocks V_j, the copied positions, and the final transition through R are fixed lookups and another $O(i)$-depth monoid product. Thus both $f_{n,i}$ and $s_{n,i}$ are computable in AC^0. □

Theorem 4.7 (Restated)**.** *For every binary regular language L and every fixed integer $i \geq 0$, there is a two-round DLM family without revision that samples $\mathrm{Pair}_{L,n}$ using $O_L(n/(\log n)^i)$ CoT positions. For $i = 0$, the construction can use exactly n CoT positions.*

Proof of Theorem 4.7. Apply Lemma B.7 and use its decomposition $u = A_1 U_1 \cdots A_m U_m R$. The CoT positions store the short source blocks $U_1, \ldots, U_m$. In the visible output, the long blocks $A_1, \ldots, A_m$ and the tail R retain their source values, while the lemma replaces each U_j by V_j and computes the final DFA state. The number of CoT positions is

$$O_L(m) = O_L\left(\frac{n}{(\log n)^i}\right).$$

Thus a two-round DLM samples the long blocks and tail into output positions and the short source blocks into CoT positions in the first round. In the second round, it fills the replaced blocks and the label. Since $f_{n,i}$ is an AC^0-computable bijection, the visible word is uniform and receives its correct label. When $i = 0$, the construction uses n CoT positions to store the source bits. □

B.4 Round complexity of autoregressive models

An autoregressive model (ARM) is a DLM without revision whose policy reveals the leftmost masked position in each round. Its predictors satisfy the same randomized-AC^0 restriction as DLM predictors and read the generated prefix. For an output of length ℓ_n and k_n CoT positions, the number of rounds is $D_n = \ell_n + k_n$. We define the round complexity of an ARM family S by

$$R_S(n) \coloneqq \max_{1 \le m \le n} (\ell_m + k_m).$$

For a distribution family $\mathcal{D}$, define $R^{\mathrm{ARM}}(\mathcal{D})$ using the same O and Ω conventions as in Section 2.2, ranging over ARM families that sample $\mathcal{D}$ with any CoT budget. Here the CoT budget may grow with n, and each CoT position adds one round. In particular, sampling $\mathrm{Pair}_{L,n}$ requires at least $n + 1$ rounds.

Lemma B.8. *For every binary regular language L, there is an ARM family that samples $\mathrm{Pair}_{L,n}$ using n CoT positions in $2n + 1$ rounds. Consequently, $R^{\mathrm{ARM}}(\mathrm{Pair}_L) \in \Theta(n)$.*

Proof. Let $G_n \colon \{0,1\}^n \to \{0,1\}^{n+1}$ be the AC^0 map from the proof of Theorem 3.1, so that $G_n(U)$ has law $\mathrm{Pair}_{L,n}$ for n independent fair bits U. Use Algorithm 3.

Algorithm 3 ARM sampling with n CoT positions as a prefix

Input: length n
$z \leftarrow \mathtt{M}^{2n+1}$
for $i = 1, \dots, n$ **do** ▷ CoT prefix, one token per round
    sample $z_i \sim \mathrm{Unif}(\{0,1\})$ independently
for $j = 1, \dots, n+1$ **do** ▷ output, one token per round
    $z_{n+j} \leftarrow \big[G_n(z_1, \dots, z_n)\big]_j$
return $(z_{n+1}, \dots, z_{2n+1})$

Every prediction is a fair bit or an AC^0 function of the generated prefix, and the returned sequence has law $\mathrm{Pair}_{L,n}$. The algorithm uses n CoT positions and $2n + 1$ rounds, which together with the $n + 1$ lower bound gives the claimed round complexity. □

C Approximate Conditional Sampling

C.1 TV and dyadic-rounding tools

Lemma C.1. *Let $\mu, \widetilde{\mu}$ be distributions on a finite set and let $K_x, \widetilde{K}_x$ be conditional distributions. Write $\mu \otimes K$ for the joint law obtained by drawing $x \sim \mu$ and then $y \sim K_x$. Then*

$$d_{\mathrm{TV}}(\mu \otimes K, \widetilde{\mu} \otimes \widetilde{K}) \le d_{\mathrm{TV}}(\mu, \widetilde{\mu}) + \mathbb{E}_{x \sim \widetilde{\mu}} d_{\mathrm{TV}}(K_x, \widetilde{K}_x). \tag{26}$$

Consequently, sequential conditional samplers with uniform local errors $a_1, \dots, a_m$ have joint error at most $\sum_i a_i$. Independent products satisfy

$$d_{\mathrm{TV}}\left(\bigotimes_i \mu_i, \bigotimes_i \nu_i\right) \le 1 - \prod_i (1 - d_{\mathrm{TV}}(\mu_i, \nu_i)) \le \sum_i d_{\mathrm{TV}}(\mu_i, \nu_i).$$

Applying the same deterministic map cannot increase TV distance. For an event G with $\mu(G) > 0$, conditioning satisfies $d_{\mathrm{TV}}(\mu, \mu(\cdot \mid G)) = \mu(G^c)$.

Proof. Insert $\widetilde{\mu} \otimes K$ in the triangle inequality. The first joint distance equals $d_{\mathrm{TV}}(\mu, \widetilde{\mu})$, and the second equals the expectation in Equation (26). Iteration gives the sequential bound. Independent maximal couplings give the product bound. Summing probabilities over fibers proves contraction under a deterministic map, and expanding the TV sum proves the conditioning identity. □

Lemma C.2. *Let $0 < \eta \leq 1$. A distribution on $M \geq 1$ outcomes has a support-preserving approximation with TV error at most η, generated by at most $\lceil \log_2 M + \log_2(1/\eta) \rceil$ fair bits. For hardwired outcomes and probabilities, the sampler is in AC^0, with size polynomial in M, the seed length, and the output length. Here support-preserving means that the approximate law assigns no mass outside the original support.*

Proof. For $m = \lceil \log_2 M + \log_2(1/\eta) \rceil$, largest-remainder rounding gives integers $a_x \geq 0$ with $\sum_x a_x = 2^m$ and $|a_x - 2^m p_x| \leq 1$. Start from $\lfloor 2^m p_x \rfloor$ and distribute the remaining units among the largest fractional remainders. No unit needs to be assigned to an outcome with $p_x = 0$, so $a_x = 0$ there. The rounded law has TV error at most $M/2^{m+1} \leq \eta$. Partition the m-bit integers into consecutive intervals of lengths a_x. Comparisons with hardwired endpoints and selection of the corresponding output are computable in constant depth. □

C.2 Two-round sampling with revision

Lemma C.3. *Let $\mathcal{G}$ be a finite deterministic binary labeled graph, possibly with missing transitions, with a start state and an accepting set. Suppose every strongly connected component (SCC) with an internal cycle has adjacency spectral radius less than 2. For every nonempty length-n accepting slice and $0 < \epsilon \leq 1/2$, a support-preserving randomized-AC^0 sampler approximates its uniform law within TV error ϵ using $n + O_{\mathcal{G}}(\log(1/\epsilon))$ fair bits. Its size is $\mathrm{poly}_{\mathcal{G}}(n, 1/\epsilon)$ and its depth depends only on $\mathcal{G}$.*

Proof. An accepting trajectory has a unique category consisting of its SCC itinerary, crossing edges, segment lengths, and final accepting state. There are $(n+1)^{O_{\mathcal{G}}(1)}$ categories and at most $|\mathcal{G}|$ segments per category. Select a category with probability proportional to its path count, rounded to TV error $\epsilon/4$ by Lemma C.2. This costs $O_{\mathcal{G}}(\log((n+1)/\epsilon))$ bits. Given the category, its SCC segments are independent uniform bridges between specified endpoints.

Consider a cyclic SCC adjacency matrix $\mathbf{A}$ with spectral radius λ, period d, and positive left and right Perron vectors $\mathbf{l}, \mathbf{r}$, normalized by $\mathbf{l}^\top \mathbf{r} = 1$. For compatible endpoints and lengths, periodic Perron–Frobenius convergence gives

$$(\mathbf{A}^b)_{uv} = d\lambda^b \mathbf{r}_u \mathbf{l}_v (1 + e_b(u,v)), \qquad |e_b(u,v)| \leq C\rho^{\lfloor b/d \rfloor}, \quad \rho < 1. \tag{27}$$

To justify this periodic form, define the stochastic matrix $\mathbf{K}(u,v) = \mathbf{A}(u,v)\mathbf{r}_v/(\lambda \mathbf{r}_u)$, whose stationary law is $\pi(v) = \mathbf{l}_v \mathbf{r}_v$. It has the same cyclic classes as $\mathbf{A}$, each with stationary mass $1/d$. On each class, $\mathbf{K}^d$ is irreducible and aperiodic. The convergence theorem of Levin & Peres (2017, Theorem 4.9) therefore gives $\mathbf{K}^b(u,v) = d\pi(v) + O(\rho^{\lfloor b/d \rfloor})$ for compatible endpoints. Since π is positive on this finite SCC and $\mathbf{A}^b(u,v) = \lambda^b \mathbf{r}_u \mathbf{K}^b(u,v)/\mathbf{r}_v$, this implies Equation (27) with a uniform relative error bound.

Choose $B \geq a \log_2((n+1)/\epsilon)$ to be a multiple of all SCC periods with $B = a \log_2((n+1)/\epsilon) + O_{\mathcal{G}}(1)$, for a sufficiently large constant a. A segment of length at least $2B$ is split into blocks of length B, with the last length in $[B, 2B)$. Shorter segments use one block. Singleton SCCs without a cycle have only zero-length segments.

For a long segment with endpoints s, t, the boundary tuple $y = (y_1, \ldots, y_{k-1})$ has weight

$$(\mathbf{A}^B)_{s,y_1} \cdots (\mathbf{A}^B)_{y_{k-2},y_{k-1}} (\mathbf{A}^{b_k})_{y_{k-1},t}.$$

Substitution of Equation (27) makes the normalized leading term a product with each boundary drawn from $\nu(v) = d\mathbf{l}_v \mathbf{r}_v$ on the cyclic class of s. This law is normalized because that class has stationary mass $1/d$, and every intermediate boundary is in that class because B is a multiple of d. With $\Delta = C\rho^{\lfloor B/d \rfloor}$, normalization bounds the likelihood ratio by $[((1-\Delta)/(1+\Delta))^k, ((1+\Delta)/(1-\Delta))^k]$. Its log magnitude is at most $4k\Delta$ when $\Delta \leq 1/2$. Since the total number of blocks is $O_{\mathcal{G}}(n+1)$, choosing a large enough makes the combined boundary-replacement error at most $\epsilon/4$. Increasing B

past the finite reachability threshold also ensures that every selected boundary tuple admits compatible bridges.

Round every boundary law and every uniform conditional block law to TV error $\delta = \epsilon/[C_0(n+1)]$, choosing C_0 so the total number of each type of draw is at most $C_0(n+1)/4$. All boundary draws run in parallel, followed by all conditional block draws. By Lemma C.1, category selection, boundary replacement, boundary rounding, and block rounding contribute at most $\epsilon/4$ each. Concatenation does not increase the resulting error.

It remains to count fair bits. Choose $1 < \gamma < 2$ larger than every SCC spectral radius. The identity $\mathbf{A}^b\mathbf{r} = \lambda^b\mathbf{r}$ implies

$$N_{u,v,b} = (\mathbf{A}^b)_{uv} \le \lambda^b \mathbf{r}_u/\mathbf{r}_v \le C_{\mathcal{G}}\gamma^b.$$

Thus a length-b block needs at most

$$b\log_2\gamma + \log_2(1/\delta) + O_{\mathcal{G}}(1) \tag{28}$$

fair bits. **Since $\gamma < 2$, the path-count contribution is fewer than one fair bit per step. This linear saving pays for boundary sampling and approximation error.**

Put $h = \log_2\gamma < 1$ and $t = \log_2((n+1)/\epsilon)$. There are $O_{\mathcal{G}}(n/B+1)$ block and boundary draws. Including the category seed, Equation (28) gives

$$r \le hn + C_1\frac{nt}{B} + C_2 t \le hn + \frac{C_1}{a}n + C_2 t,$$

where C_1, C_2 can be chosen independently of a. Indeed, each draw has $O_{\mathcal{G}}(t)$ rounding and boundary overhead, and the sum of the sampled block lengths is at most n. Take $C_1/a \le (1-h)/2$ and set $\kappa = (1-h)/2 > 0$. Since $C_2\log_2(n+1) - \kappa n$ is bounded above,

$$r \le (1-\kappa)n + C_2\log_2(n+1) + C_2\log_2(1/\epsilon) \le n + O_{\mathcal{G}}(\log(1/\epsilon)).$$

The category seed is independent of the remaining seed. Within each category, reserve disjoint slots using the maximum block budget over its possible endpoints. These budgets and output positions depend only on the category and block lengths. The remaining seed is shared across category branches, so its budget is the maximum conditional budget. Block lengths are below $2B$, hence all bridge supports have polynomial size in $n, 1/\epsilon$. Category selection, parallel boundary sampling, parallel bridge sampling, and output selection have constant total depth. □

Lemma C.4. *Let $\mathcal{A} = (Q, \{0,1\}, \delta, q_0, F)$ be a complete DFA for L, and write $L_n = L \cap \{0,1\}^n$. For every n with $L_n \neq \varnothing$ and every $0 < \epsilon \le 1/2$, a nonuniform randomized-AC^0 circuit samples an accepting trajectory within TV distance ϵ of the trajectory law induced by $\mathrm{Unif}(L_n)$, using at most*

$$r_{n,\epsilon} \le n + C_L\log_2\frac{1}{\epsilon} \tag{29}$$

fair bits. Its size is $\mathrm{poly}_L(n, 1/\epsilon)$ and its depth depends only on the chosen DFA, as do all constants indexed by L.

Proof. Expand its states to $(q,a) \in Q \times \{0,1\}$, with transition $(q,a) \mapsto (\delta(q,b),b)$ on symbol b, and add an initial copy of q_0 with the corresponding transitions and acceptance status. A trajectory now records its labels, and each step has two distinct successors of probability $1/2$. We use Z_i for this expanded chain and $\widehat{F}$ for its accepting states. Its law conditioned on $E_n = \{Z_n \in \widehat{F}\}$ is the trajectory law of $\mathrm{Unif}(L_n)$.

Let $\mathbf{A}_{\mathrm{bool}}$ be the Boolean adjacency matrix. As in Appendix A, choose a constant p with $\mathbf{A}^p_{\mathrm{bool}} = \mathbf{A}^{2p}_{\mathrm{bool}}$. In particular, p is a multiple of every SCC period. For each residue $a \pmod p$, the limit

$$\alpha_a = \lim_{j\to\infty}\Pr(E_{jp+a})$$

exists. It is positive precisely when an accepting trajectory of that residue can enter a terminal SCC. Indeed, terminal SCCs are closed, transient-state probabilities decay exponentially, and within each terminal cyclic class the chain converges exponentially to a positive stationary law by applying Levin & Peres (2017, Theorem 4.9) to the p-step chain on each terminal cyclic class. A positive accepting trajectory ending in a terminal SCC can be extended by multiples of p steps back to its accepting

endpoint with probability bounded away from zero. It therefore gives a positive limiting contribution. If $\alpha_a = 0$, every accepting trajectory of that residue avoids all terminal SCCs.

Suppose $\alpha_a > 0$. For all sufficiently large $n \equiv a \pmod p$, $\Pr(E_n) \geq c_L > 0$. Let $\mathcal{R}$ be the union of terminal SCCs and write

$$h_t(q) = \Pr_q(Z_t \in \widehat{F}).$$

Choose $T, R = O_L(\log(1/\epsilon))$ large enough for the estimates below, and put $M = n - T - R$. Bounded exceptional lengths and lengths $n < T + R$ are handled by direct lookup on at most 2^n accepted words, using Lemma C.2. In the latter case, $n = O_L(\log(1/\epsilon))$, so $2^n = \mathrm{poly}_L(1/\epsilon)$. This has polynomial size in $n, 1/\epsilon$ and satisfies Equation (29).

Set $G = \{Z_T \in \mathcal{R}\}$. Finiteness gives constants $m \geq 1$ and $\eta > 0$ such that, from every transient state, the chain reaches $\mathcal{R}$ within m steps with probability at least η. Consequently $\Pr(G^c) \leq (1-\eta)^{\lfloor T/m \rfloor}$, allowing us to choose T such that

$$\Pr(G^c \mid E_n) \leq \frac{\Pr(G^c)}{\Pr(E_n)} \leq \epsilon/4. \tag{30}$$

Thus replacing the target by its conditional law given G costs at most $\epsilon/4$. For a length-T head v ending at $q(v) \in \mathcal{R}$, its marginal under this restricted target is

$$\pi(v) = \frac{2^{-T} h_{M+R}(q(v))}{\Pr(E_n \cap G)}.$$

Only heads with positive numerator are used.

First consider the ideal construction that draws $v \sim \pi$, samples an unconditioned length-M trajectory from $q = q(v)$ using Proposition 3.2, and then draws a uniform accepting length-R tail from its endpoint s. For every resulting word vwy, its probability relative to the restricted target is

$$\frac{h_{M+R}(q)}{h_R(s)}. \tag{31}$$

If q is in cyclic class i of a terminal SCC of period d, every reachable middle endpoint s is in class $i + M \pmod d$. Both acceptance probabilities therefore refer to the final class $i + M + R \pmod d$. For that class, their common limiting acceptance probability is $\beta = d\pi_{\mathrm{SCC}}(\widehat{F} \cap \text{final class})$, where π_{SCC} is the stationary law on the SCC. The positive head weight ensures $\beta > 0$. On every such compatible class, periodic convergence gives $h_t(u) = \beta(1 + O_L(\rho^{\lfloor t/p \rfloor}))$ with $0 < \rho < 1$. There are only finitely many classes. We may therefore choose $R = O_L(\log(1/\epsilon))$ so that every reachable s has $h_R(s) > 0$ and Equation (31) lies in $[1 - \epsilon/4, 1 + \epsilon/4]$. The ideal construction is within TV distance $\epsilon/4$ of the restricted target. The middle itself uses M fair bits and introduces no sampling error.

Apply Lemma C.2 to π and, uniformly over s, to the accepting tail law, giving each TV error $\epsilon/4$. Both supports have size at most 2^T or 2^R. By Lemma C.1, the total error is bounded by

$$\underbrace{\epsilon/4}_{\text{head restriction}} + \underbrace{\epsilon/4}_{\text{middle replacement}} + \underbrace{\epsilon/4}_{\text{head rounding}} + \underbrace{\epsilon/4}_{\text{tail rounding}} \leq \epsilon.$$

The seed length is

$$[T + O_L(\log(1/\epsilon))] + M + [R + O_L(\log(1/\epsilon))] = n + O_L(\log(1/\epsilon)).$$

The short boundary lookups have polynomial size. For each possible middle start state, use the DFA-run circuit with the same middle seed and select the appropriate output. The three stages have constant total depth.

Suppose $\alpha_a = 0$. Delete the terminal SCCs. This preserves all accepting trajectories of the current residue and their weights 2^{-n}. The remaining transition matrix is substochastic. We retain its edge weights rather than normalizing rows, which would change relative path probabilities.

For each cyclic SCC remaining after deleting terminal SCCs, let $\mathbf{A}$ be its internal adjacency matrix, λ its Perron root, and $\mathbf{l}$ a positive left Perron vector. The row sums of $\mathbf{A}$ are at most 2, with at least one strictly smaller. Let $\mathbf{1}$ be the all-ones column vector of the appropriate dimension. Then

$$\lambda \mathbf{l}^\top \mathbf{1} = \mathbf{l}^\top \mathbf{A} \mathbf{1} < 2\mathbf{l}^\top \mathbf{1},$$

so $\lambda < 2$. Lemma C.3 applies and proves Equation (29) for zero-density residues.

All lookup tables, length residues, and probability thresholds are hardwired. Projection to the labels preserves the error bound by Lemma C.1. □

Theorem 5.1 (Restated)**.** *For every binary regular language L and error tolerance $0 < \epsilon \le 1/2$, there is a DLM family with revision that ϵ-approximately samples $\mathrm{Word}_{L,n}$ in two rounds using $O_L(\log(1/\epsilon))$ CoT positions. The policy and predictor circuits have size $\mathrm{poly}_L(n, 1/\epsilon)$ and depth $O_L(1)$.*

Proof. By Lemma C.4, an AC^0 map of at most $n + C_L \log_2(1/\epsilon)$ fair bits samples an accepting trajectory with TV error at most ϵ. Projecting to its input symbols gives the same error bound for $\mathrm{Word}_{L,n}$ by Lemma C.1. Use the n output positions and $O_L(\log(1/\epsilon))$ CoT positions to store the fair bits in round one, filling any unused positions with zeros. In round two, evaluate this map and overwrite the output positions with the sampled word. Leave the CoT positions unchanged. The policy distinguishes the initial masked sequence from the filled sequence, and all deterministic maps have size $\mathrm{poly}_L(n, 1/\epsilon)$ and depth $O_L(1)$. □

C.3 Round lower bound without revision

Lemma C.5. *Let $P_n = \mathrm{Unif}\{0^j 1^{n-j} : 0 \le j \le n\}$. The output distribution Q of any D-round no-revision sampler with K CoT positions satisfies*

$$1 - d_{\mathrm{TV}}(P_n, Q) \le \frac{2^K 64^D}{\sqrt{n+1}}. \tag{32}$$

Consequently, constant K and TV error at most any constant $\epsilon < 1$ require $\Omega_{K,\epsilon}(\log n)$ rounds, even with unrestricted computation.

Proof. Define the affinity $\mathrm{Aff}(P, Q) = \sum_x \sqrt{P(x)Q(x)}$. Since $\min\{P(x), Q(x)\} \le \sqrt{P(x)Q(x)}$, it suffices to bound $\mathrm{Aff}(P_n, Q)$.

First, for every product distribution q on m bits,

$$\sum_{j=0}^{m} \sqrt{q(0^j 1^{m-j})} \le 64. \tag{33}$$

To see this, let $p_i = q(X_i = 1)$. Swapping an adjacent inversion $p_i > p_{i+1}$ increases only the summand with cut at i and leaves the others unchanged. We may assume $p_1 \le \cdots \le p_m$. Let r be the number of p_i in $[1/4, 3/4]$. The $r + 1$ cuts in or adjacent to this interval each contribute at most $(\sqrt{3}/2)^r$. Moving a cut outward through either tail decreases its contribution by a factor at most $1/\sqrt{3}$. The sum is therefore at most

$$\frac{2}{1 - 1/\sqrt{3}} + (r+1)(\sqrt{3}/2)^r \le \frac{2}{1 - 1/\sqrt{3}} + \frac{1}{(1 - \sqrt{3}/2)^2} < 64.$$

Degenerate probabilities follow by continuity.

Next allow arbitrary history-dependent control and free deterministic completion after the last round. This only enlarges the sampler class. Without CoT, we prove by induction that its affinity with a uniform target on s possible change points is at most $64^D/\sqrt{s}$. For $D = 0$, a deterministic output has affinity at most $1/\sqrt{s}$. In the first round, the revealed positions have a product law q. A legal revealed pattern $0^j 1^{m-j}$ leaves s_j possible change points, uniformly distributed, with $\sum_j s_j = s$. Coordinates already determined by the target can be supplied freely to the controller. Conditional on each legal pattern, induction and Equation (33) give

$$\mathrm{Aff}(P, Q) \le \sum_{j : s_j > 0} \sqrt{\frac{s_j}{s} q(0^j 1^{m-j})} \frac{64^{D-1}}{\sqrt{s_j}} \le \frac{64^D}{\sqrt{s}}.$$

Illegal patterns contribute zero. The same argument covers a subset of the available change points and any adaptive choice of positions.

Finally, let X be the output word and B the random string in the K CoT positions. For $b \in \{0,1\}^K$, let $Q_b(x) = \Pr(X = x, B = b)$. Force each CoT draw to its prescribed value in b, keeping the output kernels unchanged, and denote the resulting output law by $\widehat{Q}_b$. Any histories that cannot occur in the original sampler may be completed arbitrarily. A complete output and CoT assignment determines at most one execution because the policy is deterministic and tokens are never revised. Forcing removes only CoT probability factors, each at most one, so $Q_b(x) \le \widehat{Q}_b(x)$. The forced process can be simulated by a controller without CoT that knows b, hence

$$\mathrm{Aff}(P_n, Q) \le \sum_b \mathrm{Aff}(P_n, \widehat{Q}_b) \le \frac{2^K 64^D}{\sqrt{n+1}}.$$

This proves Equation (32). In particular, TV error at most ϵ implies $6D + K \ge \frac{1}{2}\log_2(n+1) + \log_2(1-\epsilon)$, proving Theorem 5.2. □

Theorem 5.2 (Restated)**.** *For the language $L = 0^*1^*$, every constant CoT bound K, and every error tolerance $0 < \epsilon < 1$, any DLM family $S = (S_n)_{n\ge1}$ without revision that ϵ-approximately samples $\mathrm{Word}_{L,n}$ with $k_n \le K$ CoT positions must use $D_n = \Omega_{K,\epsilon}(\log n)$ decoding rounds. This lower bound holds even when the policies and predictors are computationally unrestricted.*

Proof. Apply Lemma C.5 with $D = D_n$ and at most K CoT positions. TV error at most ϵ implies $6D_n + K \ge \frac{1}{2}\log_2(n+1) + \log_2(1-\epsilon)$. For constant K and $\epsilon < 1$, this gives $D_n = \Omega_{K,\epsilon}(\log n)$. The bound does not restrict the computation performed by the policy or predictors. □

D Transformer Realization

D.1 Circuit-to-Transformer implementation

We adopt the finite-precision arithmetic of Li et al. (2024, Section 3.1), with ties rounded toward zero and overflow saturated at the representable endpoints. Iterated sums round after each addition in the prescribed order.

For constant precision $s \ge 2$, let $[\cdot]_s$ denote rounding to the nearest element of $\mathcal{F}_s$, where

$$\mathcal{F}_s = \{j2^{-s} : j \in \mathbb{Z},\ |j| \le 2^{2s} - 1\}, \qquad B_s = \max \mathcal{F}_s = 2^s - 2^{-s}, \qquad [\exp(-B_s)]_s = 0.$$

Lemma D.1. *A map from a length-N token sequence to the logits at all positions is computable by a polynomial-width, constant-depth, constant-precision Transformer with full attention if and only if the binary representations of its logits are jointly AC^0-computable. For causal attention, the same equivalence holds for logits that depend only on the available prefix.*

Proof. The layerwise AC^0 simulation of Li et al. (2024, Theorem 3.1) also applies when attention sums range over all positions, and computes the binary representations of all logits jointly. For the converse, we adapt their circuit-to-Transformer construction. One-hot positional encodings and a feed-forward layer place each token's binary encoding in a disjoint value block. Set every visible attention score to B_s. Since $[\exp(B_s)]_s = B_s$ and a rounded sum of copies of B_s equals B_s, every visible attention weight is 1. Full attention therefore gathers the entire input at every position, while causal attention gathers the available prefix.

Retaining the destination's positional encoding lets a shared feed-forward network select its output circuits and compute their logit digits in parallel. For each logit, use the signed binary expansion

$$z = \sum_{k=-s}^{s-1} d_k 2^k, \qquad d_k \in \{-1, 0, 1\},$$

with all nonzero digits having the sign of z. These digits are AC^0-computable from the logit's binary representation. Store them in hidden coordinates and use output projection weights 2^k. Each partial sum lies in $\mathcal{F}_s$, so the projection recovers z without rounding error or overflow. The construction preserves constant depth and precision and polynomial width, and adds no token positions or decoding rounds. □

D.2 TRANSFER OF THE SAMPLING BOUNDS

For a constant-sized vocabulary $\Sigma = \{v_1, \ldots, v_t\}$, token sampling and unmasking are implemented by the following procedures.

Algorithm 4 Sampling with logits

Input: token logits $z_1, \ldots, z_t \in \mathcal{F}_s$
for all $j = 1, \ldots, t$ **do**
    $w_j \leftarrow [\exp(z_j)]_s$
$W \leftarrow [w_1 + \cdots + w_t]_s$
if $W = 0$ **then**
    return token v_1
for all $j = 1, \ldots, t$ **do**
    $p_j \leftarrow [w_j / W]_s$
sample r uniformly from $\{0, 1, \ldots, 2^s - 1\}$
$u \leftarrow r / 2^s$
if $u \geq \sum_{i=1}^{t} p_i$ **then**
    return token v_t
$J \leftarrow \min\{k \in [t] : u < \sum_{i=1}^{k} p_i\}$
return token v_J

Algorithm 5 Unmasking with logits

Input: policy logits $a_{\text{keep}}, a_{\text{unmask}} \in \mathcal{F}_s$
if $a_{\text{keep}} < a_{\text{unmask}}$ **then**
    return unmask
else
    return keep

Theorem 2.1 (Restated)**.** *All round and CoT upper and lower bounds in Sections 3 to 5 remain valid for Transformer-realized families under their stated assumptions.*

Proof of Theorem 2.1. Lower bounds. By Lemma D.1, Transformer logits are AC^0-computable. At constant precision and vocabulary size, Algorithm 4 and Algorithm 5 use constant-size computations and $O(1)$ fair bits per selected position. Thus Transformer-realized families satisfy the circuit restrictions in Section 2.1, so all lower bounds apply.

Upper bounds. It suffices to show that the logits implementing our constructions are AC^0-computable, then apply Lemma D.1. For the vocabulary $\{\mathtt{M}, 0, 1\}$, the distributions concentrated on 0, concentrated on 1, and uniform on $\{0, 1\}$ are realized by the respective logit vectors

$$(-B_s, 0, -B_s), \qquad (-B_s, -B_s, 0), \qquad (-B_s, 0, 0). \tag{34}$$

For remasking, $(0, -B_s, -B_s)$ realizes the distribution concentrated on $\mathtt{M}$. These claims follow from $[\exp(-B_s)]_s = 0$. All upper-bound constructions choose among these distributions in AC^0. This includes the label marginal in Equation (23) and the posterior of the last hidden bit, which is deterministic or uniform on $\{0, 1\}$ on every reachable state. Their predictor logits are therefore AC^0-computable. Their policies are also in AC^0 and are implemented by choosing between the logit pairs $(0, -B_s)$ and $(-B_s, 0)$. Hence Lemma D.1 realizes the same policies and conditional distributions, preserving the round and CoT bounds and the output law.

Causal attention for ARMs. In Lemma B.8, all source bits appear in the CoT prefix, and each subsequent prediction is an AC^0 function of that prefix. Thus the required logits depend only on already generated tokens, and the causal version of Lemma D.1 applies, preserving the n CoT positions and $2n + 1$ rounds. The $n + 1$ round lower bound follows because an ARM generates one token per round. □

E RANDOMIZED-AC^0 SAMPLING OF FINITE-STATE DISTRIBUTIONS

E.1 MOTIVATION

Uniform sampling of accepted words supports model-based testing and Monte Carlo model checking. Basset et al. (2017) study this task for networks of automata, where choosing transitions uniformly can strongly bias the distribution over complete executions. In their discussion of sampling in NC^0 and AC^0, Horacsek et al. (2026, Section 1.1) write that "it would be interesting to sample random walks on graphs (equivalently, Markov chains)".

Motivated by these questions, we obtain the following four randomized-AC^0 sampling results.

- For every binary regular language L, the input–output pair distribution $\mathrm{Pair}_{L,n}$ can be sampled exactly using n fair bits.
- For every binary regular language L with $L \cap \{0,1\}^n \neq \varnothing$, the uniform distribution on $L \cap \{0,1\}^n$ can be sampled to TV error at most ϵ using at most $n + C_L \log_2(1/\epsilon)$ fair bits, with every output in that set.
- For every finite Markov chain $\mathcal{M}$, its complete trajectories can be sampled exactly if and only if every finite trajectory has dyadic probability. When exact sampling is possible, it uses $O_{\mathcal{M}}(n+1)$ fair bits.
- For every finite Markov chain $\mathcal{M}$ and set of states F with $\Pr[X_n \in F] > 0$, trajectories conditioned on $X_n \in F$ can be sampled to TV error at most ϵ using $O_{\mathcal{M}}(n + \log(1/\epsilon))$ fair bits. Every output is a positive-probability trajectory ending in F.

E.2 SAMPLING DISTRIBUTIONS INDUCED BY REGULAR LANGUAGES

Theorem E.1 (Exact sampling of input–output pairs)**.** *For every binary regular language L, there is a polynomial-size randomized-AC^0 circuit family that samples $\mathrm{Pair}_{L,n}$ exactly.*

Proof. Fix a DFA recognizing L. By Proposition 3.2, a randomized-AC^0 circuit samples its complete run on a uniform $X \sim \mathrm{Unif}(\{0,1\}^n)$ exactly. Output X together with the indicator that the final state is accepting. This constant-depth postprocessing gives $\mathrm{Pair}_{L,n}$. □

Theorem E.2 (Approximate uniform sampling from regular languages)**.** *For every binary regular language L, there is a nonuniform family $(C_{n,\epsilon})$ of randomized-AC^0 circuits such that, for every n with $L \cap \{0,1\}^n \neq \varnothing$ and every $0 < \epsilon \leq 1/2$, the output distribution $Q_{n,\epsilon}$ of $C_{n,\epsilon}$ satisfies*

$$d_{\mathrm{TV}}(Q_{n,\epsilon}, \mathrm{Word}_{L,n}) \leq \epsilon.$$

Each circuit uses at most $n + C_L \log_2(1/\epsilon)$ fair bits and has size polynomial in n and $1/\epsilon$. Its depth depends only on L, and $Q_{n,\epsilon}$ is supported on $L \cap \{0,1\}^n$.

Proof. Apply Lemma C.4 to a complete DFA recognizing L and project each sampled accepting trajectory to its sequence of input symbols. This projection produces a distribution on $L \cap \{0,1\}^n$ and does not increase total variation distance. □

E.3 EXACT SAMPLING OF MARKOV TRAJECTORIES

Lemma E.3. *For a finite Markov chain $\mathcal{M} = (Q, \mu, \mathbf{P})$ with state set Q, initial distribution μ, and transition matrix $\mathbf{P}$, let $\mathrm{Path}_n(\mathcal{M})$ denote the law of $(X_0, \ldots, X_n)$, where $X_0 \sim \mu$ and X_{i+1} is drawn from $\mathbf{P}(X_i, \cdot)$. For a trajectory $\gamma = (x_0, \ldots, x_n)$, write $p_{\mathcal{M}}(\gamma) \coloneqq \mu(x_0) \prod_{i=1}^{n} \mathbf{P}(x_{i-1}, x_i)$. Call $\mathcal{M}$ **path-dyadic** if every finite trajectory has dyadic probability, and **transition-dyadic** if every entry of μ and $\mathbf{P}$ is dyadic. For every transition-dyadic chain $\mathcal{M}$, there are a DFA $\mathcal{B} = (Q_{\mathcal{B}}, \{0,1\}, \delta_{\mathcal{B}}, q_0, F)$, constants $v, s \geq 1$, and a map $\phi : Q_{\mathcal{B}} \to Q$ such that, for every n, the run $(X, y_0, y_1, \ldots, y_{v+sn})$ of $\mathcal{B}$ on a uniform $X \sim \mathrm{Unif}(\{0,1\}^{v+sn})$ satisfies*

$$\big(\phi(y_v), \phi(y_{v+s}), \ldots, \phi(y_{v+sn})\big) \overset{d}{=} \mathrm{Path}_n(\mathcal{M}). \tag{35}$$

Proof. Fix $s \geq 1$ with $2^s\mathbf{P}(q,r) \in \mathbb{Z}$ for all q, r, and let $\Lambda := \{0,1\}^s$. Build a DFA $\mathcal{C} = (Q, \Lambda, \delta_C, q_0, F)$ in which exactly $2^s\mathbf{P}(q,r)$ letters move q to r. Then a uniform letter moves q to r with probability $\mathbf{P}(q,r)$. Fix $v \geq 1$ with $2^v\mu(q) \in \mathbb{Z}$ for all q, and a map $g : \{0,1\}^v \to Q$ such that $|\{e : g(e) = q\}| = 2^v\mu(q)$ for every q.

We encode $\mathcal{C}$ as a binary DFA by buffering blocks of s bits. Fix the binary tree of depth v with 2^v leaves and edges labeled 0 and 1. Let $\mathcal{B}$ have states

$$Q_{\mathcal{B}} := \big(\bigcup_{j<v}\{0,1\}^j\big) \sqcup \big(Q \times \bigcup_{j<s}\{0,1\}^j\big),$$

where a node at depth $j < v$ is named by the word $e \in \{0,1\}^j$ labeling its path from the root, the 2^v leaves $e \in \{0,1\}^v$ are identified with the states $(g(e), \varepsilon)$, and ε denotes the empty buffer. The start state is the root. The transition $\delta_{\mathcal{B}}$ moves down the tree for the first v steps. From a buffer state (q, w) with $|w| < s-1$ it appends the input bit, and with $|w| = s-1$ it reads the last bit, obtains a letter $u \in \Lambda$, and moves to $(\delta_C(q,u), \varepsilon)$. Set $\phi(q, \varepsilon) := q$ and define ϕ arbitrarily on the remaining states.

On a uniform word, the first v bits select a leaf e uniformly, so the buffer state after v steps is $(g(e), \varepsilon)$ with $g(e) \sim \mu$, and each subsequent block of s bits is then a uniform letter of Λ, independent of the others. Hence, at times $v, v+s, v+2s, \ldots$, the states of $\mathcal{B}$ are $(Y_0, \varepsilon), (Y_1, \varepsilon), \ldots$ where (Y_j) is the chain of $\mathcal{M}$ started from μ. Applying ϕ gives Equation (35). □

Lemma E.4. *Every path-dyadic Markov chain $\mathcal{M} = (Q, \mu, \mathbf{P})$ is the projection of a finite transition-dyadic chain $\widetilde{\mathcal{M}} = (\widetilde{Q}, \widetilde{\mu}, \widetilde{\mathbf{P}})$: there is a map $\phi : \widetilde{Q} \to Q$ such that, for every n,*

$$\big(\phi(\widetilde{X}_0), \ldots, \phi(\widetilde{X}_n)\big) \overset{d}{=} (X_0, \ldots, X_n), \tag{36}$$

where $(\widetilde{X}_i)$ is the chain with initial distribution $\widetilde{\mu}$ and transition matrix $\widetilde{\mathbf{P}}$, and (X_i) is the chain of $\mathcal{M}$.

Proof. Delete the states that are unreachable from the support of μ. This does not change the trajectory law, and afterwards every state carries a positive trajectory. Every positive transition probability is then rational, since $\mathbf{P}(q,r) = p_{\mathcal{M}}(\gamma r)/p_{\mathcal{M}}(\gamma)$ for any positive trajectory γ ending at q, since both trajectory probabilities in this ratio are dyadic.

Fix an odd prime p and, for a positive rational a/b in lowest terms, let $v_p(a/b)$ denote the exponent of p in a minus the exponent of p in b. Let $\mathcal{G}(r)$ be the set of positive trajectories ending at r, and define

$$\alpha_r^p := \min_{\gamma \in \mathcal{G}(r)} v_p\big(p_{\mathcal{M}}(\gamma)\big) \geq 0, \tag{37}$$

which is finite and attained, since $\mathcal{G}(r)$ is nonempty and v_p is nonnegative on positive dyadic rationals.

Appending the edge $q \to r$ to a trajectory that attains α_q^p gives

$$\alpha_r^p \leq \alpha_q^p + v_p\big(\mathbf{P}(q,r)\big) \qquad \text{whenever } \mathbf{P}(q,r) > 0. \tag{38}$$

The length-zero trajectory at r gives

$$\alpha_r^p \leq v_p\big(\mu(r)\big) \qquad \text{whenever } \mu(r) > 0. \tag{39}$$

Only finitely many odd primes occur in the numerators or denominators of the positive entries of μ and $\mathbf{P}$. Every positive trajectory probability is a product of these entries, so $\alpha_q^p = 0$ for every other odd prime. Hence

$$w_q := \prod_{p \text{ odd prime}} p^{\alpha_q^p} \tag{40}$$

is a finite positive integer for every q. For every odd prime p, Equation (38) gives $v_p\big(w_q\mathbf{P}(q,r)/w_r\big) = \alpha_q^p + v_p(\mathbf{P}(q,r)) - \alpha_r^p \geq 0$ when $\mathbf{P}(q,r) > 0$, and Equation (39) gives $v_p\big(\mu(q)/w_q\big) \geq 0$ when $\mu(q) > 0$. Therefore,

$$\frac{\mu(q)}{w_q} \quad \text{and} \quad \frac{w_q\mathbf{P}(q,r)}{w_r} \quad \text{are dyadic.} \tag{41}$$

Now in $\widetilde{\mathcal{M}}$ each $q \in Q$ corresponds to w_q states, $\mathcal{F}_q \coloneqq \{(q,1),\dots,(q,w_q)\}$. Let $\widetilde{Q} \coloneqq \bigsqcup_q \mathcal{F}_q$ and $\phi(q,i) \coloneqq q$, and set $\widetilde{\mu}(q,i) \coloneqq \mu(q)/w_q$. We construct $\widetilde{\mathbf{P}}$ so that

$$\sum_{h' \in \widetilde{Q}} \widetilde{\mathbf{P}}(h,h') = 1, \qquad \sum_{h \in \mathcal{F}_q} \widetilde{\mathbf{P}}(h,h') = \frac{w_q \mathbf{P}(q,r)}{w_r} \tag{42}$$

for every $h \in \widetilde{Q}$, every $q, r \in Q$ and every $h' \in \mathcal{F}_r$.

Choose σ such that $m_{q,r} \coloneqq 2^\sigma w_q \mathbf{P}(q,r)/w_r$ is an integer for every q, r. For each q we need an integer matrix $\mathbf{C}^{(q)}$ with rows indexed by $\mathcal{F}_q$ and columns by $\widetilde{Q}$, whose rows all sum to 2^σ and whose column sums satisfy $\sum_{h \in \mathcal{F}_q} \mathbf{C}^{(q)}(h,h') = m_{q,r}$ for every $h' \in \mathcal{F}_r$ and $r \in Q$. Both totals equal $2^\sigma w_q$, since $\sum_r w_r m_{q,r} = 2^\sigma w_q \sum_r \mathbf{P}(q,r) = 2^\sigma w_q$. List each column label as many times as its required column sum, and partition the resulting list into w_q consecutive groups of size 2^σ. The counts in each group form one row of $\mathbf{C}^{(q)}$, giving the required nonnegative integer matrix. For $h \in \mathcal{F}_q$, set $\widetilde{\mathbf{P}}(h,h') \coloneqq \mathbf{C}^{(q)}(h,h')/2^\sigma$.

We claim by induction on n that for every positive trajectory $\gamma = (x_0,\dots,x_n)$ and every $h \in \mathcal{F}_{x_n}$,

$$\Pr\big[\phi(\widetilde{X}_0) = x_0, \dots, \phi(\widetilde{X}_{n-1}) = x_{n-1}, \widetilde{X}_n = h\big] = \frac{p_{\mathcal{M}}(\gamma)}{w_{x_n}}. \tag{43}$$

For $n = 0$ this is the definition of $\widetilde{\mu}$. For the step, let $\gamma^- = (x_0,\dots,x_{n-1})$, $q = x_{n-1}$, $r = x_n$, and $h' \in \mathcal{F}_r$. Summing Equation (43) at level $n-1$ over $h \in \mathcal{F}_q$ and moving one step,

$$\Pr\big[\phi(\widetilde{X}_0) = x_0, \dots, \widetilde{X}_n = h'\big] = \sum_{h \in \mathcal{F}_q} \frac{p_{\mathcal{M}}(\gamma^-)}{w_q} \widetilde{\mathbf{P}}(h,h') = \frac{p_{\mathcal{M}}(\gamma^-)\,\mathbf{P}(q,r)}{w_r} = \frac{p_{\mathcal{M}}(\gamma)}{w_r}. \tag{44}$$

The second equality uses Equation (42), and the last uses $p_{\mathcal{M}}(\gamma) = p_{\mathcal{M}}(\gamma^-)\mathbf{P}(q,r)$. This proves Equation (43). Summing it over $h \in \mathcal{F}_{x_n}$ gives $\Pr[\phi(\widetilde{X}_0) = x_0, \dots, \phi(\widetilde{X}_n) = x_n] = p_{\mathcal{M}}(x_0,\dots,x_n)$ for every positive trajectory, and both sides vanish for infeasible ones, which is Equation (36). □

Theorem E.5 (Exact sampling of Markov trajectories)**.** *Let $\mathcal{M}$ be a finite Markov chain. The complete trajectory family $(\mathrm{Path}_n(\mathcal{M}))_{n\ge 0}$ admits an exact randomized-AC^0 sampler family of size polynomial in $n+1$ if and only if every finite trajectory of $\mathcal{M}$ has dyadic probability. The depth bound is independent of n.*

Proof. For necessity, fix a length n and let $C_n : \{0,1\}^{r(n)} \to Q^{n+1}$ be an exact sampler for $\mathrm{Path}_n(\mathcal{M})$. Each trajectory in its output receives one preimage for every random string mapped to it, so its probability is an integer multiple of $2^{-r(n)}$. Exactness therefore forces every trajectory probability to be dyadic.

For sufficiency, let $\mathcal{M}$ be path-dyadic. By Lemma E.4 it is the projection of a transition-dyadic chain $\widetilde{\mathcal{M}}$ under a map ϕ. By Lemma E.3 there are a binary DFA $\mathcal{B}$ with start state and constants v, s such that the run of $\mathcal{B}$ on a uniform word of length $v + sn$ projects, at times $v, v+s, \dots, v+sn$, to $\mathrm{Path}_n(\widetilde{\mathcal{M}})$. By Proposition 3.2, $\mathcal{B}$ has an exact randomized-AC^0 sampler for its run using $v + sn$ fair bits and polynomial size. Reading off those positions and applying ϕ is a constant-depth postprocessing, yielding an exact randomized-AC^0 sampler for $\mathrm{Path}_n(\mathcal{M})$ of polynomial size. □

E.4 Approximate sampling of acceptance-conditioned Markov trajectories

Theorem E.6 (Approximate sampling of acceptance-conditioned Markov trajectories)**.** *Let $\mathcal{M} = (Q, \mu, \mathbf{P})$ be a finite Markov chain and $F \subseteq Q$. For every $n \ge 0$ with $\Pr[X_n \in F] > 0$ and every $0 < \epsilon \le 1/2$, there is a nonuniform randomized-AC^0 sampler for $\mathrm{Law}(X_0,\dots,X_n \mid X_n \in F)$ with TV error at most ϵ. It uses $O_{\mathcal{M}}(n + \log(1/\epsilon))$ fair bits, has size $\mathrm{poly}_{\mathcal{M}}(n+1, 1/\epsilon)$ and depth depending only on $\mathcal{M}$, and outputs only positive-probability trajectories ending in F.*

Proof. The SCC decomposition and block construction of Lemma C.3 also apply to weighted trajectories. Without the coefficient-one random-bit requirement, no subcriticality assumption is

needed. We adapt the construction in Lemma C.3, replacing path counts by path probabilities. Decompose the support graph of $\mathbf{P}$ into SCCs. A category records the initial state, SCC itinerary, crossing transitions, segment lengths, and final state in F. There are $(n+1)^{O_{\mathcal{M}}(1)}$ categories. Select a category according to its probability conditioned on $X_n \in F$, rounded to TV error $\epsilon/4$ using Lemma C.2. These conditional probabilities are hardwired, so no lower bound on $\Pr[X_n \in F]$ is required. Given the category, the SCC segments are independent weighted bridges.

For a cyclic SCC, let $\mathbf{W}$ be the restriction of $\mathbf{P}$ to that SCC. This irreducible nonnegative matrix may be substochastic. Let $\lambda > 0$ be its Perron root, d its period, and $\mathbf{l}, \mathbf{r}$ positive left and right Perron vectors normalized by $\mathbf{l}^\top \mathbf{r} = 1$. The stochastic matrix $\mathbf{K}(u,v) = \mathbf{W}(u,v)\mathbf{r}_v/(\lambda \mathbf{r}_u)$ gives exactly the argument leading to Equation (27), now with $\mathbf{W}$ in place of $\mathbf{A}$. In particular, for compatible endpoints and sufficiently large b,

$$(\mathbf{W}^b)_{uv} = d\lambda^b \mathbf{r}_u \mathbf{l}_v \big(1 + O_{\mathcal{M}}(\rho^{\lfloor b/d \rfloor})\big), \qquad 0 < \rho < 1.$$

Choose $B = \Theta_{\mathcal{M}}(\log((n+1)/\epsilon))$ divisible by all cyclic SCC periods and beyond their reachability thresholds. Split long segments into blocks of lengths in $[B, 2B)$, as in Lemma C.3. The exact boundary weight is a product of entries of powers of $\mathbf{W}$. Substitution of the displayed estimate again leaves the product law $\nu(v) = d\mathbf{l}_v \mathbf{r}_v$ on the appropriate cyclic class. The same likelihood-ratio bound makes the total boundary-replacement error at most $\epsilon/4$. Every boundary tuple in this product support admits positive-weight bridges. Segments shorter than $2B$ use one block, and acyclic singleton SCCs require no internal transitions.

Conditioned on the boundaries, a block from u to v of length b has path probabilities proportional to $\prod_{j=1}^{b} \mathbf{P}(x_{j-1}, x_j)$, with normalizer $(\mathbf{W}^b)_{uv}$. Apply Lemma C.2 to the boundary laws and these weighted block laws with local error $\delta = \epsilon/[C(n+1)]$, where C depends only on $\mathcal{M}$ and is large enough that each type of rounding contributes at most $\epsilon/4$. Draw all boundaries in parallel and then all blocks in parallel. By Lemma C.1, the four error contributions sum to at most ϵ. All rounding preserves support, so every output is a positive-probability accepting trajectory.

For the bit count, put $t = \log_2((n+1)/\epsilon)$. A length-$b$ block has at most $|Q|^{b+1}$ possible state sequences and therefore uses $O_{\mathcal{M}}(b+t)$ fair bits. Each boundary uses $O_{\mathcal{M}}(t)$ bits, and there are $O_{\mathcal{M}}(n/B+1)$ blocks and boundaries. Including the category seed, the total is

$$O_{\mathcal{M}}(n + (n/B+1)t) = O_{\mathcal{M}}(n + \log(1/\epsilon))\,.$$

As in Lemma C.3, category branches share the same remaining random-bit slots, whose total budget is the maximum over categories. Each block has length below $2B$, so its support is polynomial in $n+1, 1/\epsilon$. This also covers $B > n$, when whole segments are sampled directly. Polynomially many category and endpoint branches, constant-depth rounding circuits, and output multiplexers give the claimed size and depth bounds. The case $n = 0$ is a direct application of Lemma C.2 to μ conditioned on F. Taking $F = Q$ also gives approximate sampling of unconditioned trajectories. □