One Retrieved Token per Step Is Enough: Tight Linear Chain-of-Thought Complexity for Directed Reachability

Anonymous authors
Paper under double-blind review

Abstract

A recent ICML paper asks whether directed reachability in the self-consistent bounded-attention prefix-oracle model can be solved with $O(m + n)$ chain-of-thought tokens, rather than the known $O(n^2)$ construction. We give a sparse-sensitive resolution: for every n-vertex, m-edge input, including arbitrary edge order, self-loops, and repeated tokens, a $(0, 1)$-cBAPO-CoT decides reachability using at most $3m + 2$ nonhalt tokens and an alphabet of size $\Theta(|\Sigma|)$. Together with the existing linear lower bound under fixed-bandwidth, worst-case quantifiers, this gives tight $\Theta(m + 1)$ token complexity. The construction combines split-invariant predecessor attention over present edges with threaded depth-first search: each discovery token records visitedness, a parent, and the continuation needed after backtracking. We also give a $3m + 2$-token simple-path certificate and a $3|Q|m + 2$ bound for regular reachability with a fixed deterministic finite automaton. A separate result shows that arbitrary-oracle cBAPOs can decide nonrecursive length languages even over a binary alphabet, whereas fixed finite-alphabet machines with total-recursive suffix oracles are Turing-simulable. Existing exhaustive and randomized verification covers 14,236 instances. An illustrative single-seed neural study is retained as diagnostic evidence, not as a realization or proof of the oracle model.

1 Introduction

Autoregressive reasoning has become a principal mechanism for improving the accuracy of large language models (LLMs): a model emits intermediate chain-of-thought (CoT) tokens, repeatedly conditions on its own partial computation, and only then produces an answer (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2023; Yao et al., 2023; Besta et al., 2024). This mechanism is computationally expensive. If an input has N_0 tokens and a derivation uses T additional tokens, then even with a key–value cache, the number of token–key score evaluations scales as

$$\mathcal{A}(N_0, T) = \sum_{\tau=0}^{T-1} (N_0 + \tau) = TN_0 + \frac{T(T-1)}{2}. \tag{1}$$

Consequently, reducing T reduces this abstract score-evaluation count. Translating that reduction into wall-clock latency or memory savings requires an implementation-level analysis that is outside our theorem.

The bounded-attention prefix oracle (BAPO) model formalizes a communication bottleneck that arises in causal transformer computation (Schnabel et al., 2025). At every next-token step, an arbitrary split divides the current context into a prefix and suffix. A suffix-side computation receives only a summary bits from the prefix and at most b individually selected prefix tokens. The BAPO-CoT model iterates this constrained next-token operation. Recent work introduced a self-consistent restriction, cBAPO, and proved the bounds

$$\mathrm{TC}^{\mathrm{cBAPO}}_{\mathrm{REACH}}(m) = \Omega(m), \qquad c^{\mathrm{cBAPO}}_{\mathrm{REACH}}(n, m) = O(n^2).$$

It asked whether the upper bound can be reduced to $O(m+n)$ or whether $\Theta(n^2)$ is intrinsic (Tomlinson et al., 2026).

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.

The known construction repeatedly writes an n-token visited/frontier representation. Our central observation is that score attention is stronger than equality lookup. If the current node is u and the next admissible label must be smaller than j, a top-one score can return

$$\mathrm{pred}_E(u, j) := \max\{v < j : (u, v) \in E\}, \tag{2}$$

with $\perp$ for an empty set, without scanning absent labels. A second obstacle is the DFS stack. We instead use a persistent discovery token

$$D_{p,v} \equiv \text{“}v \text{ is visited and its DFS parent is } p\text{”}. \tag{3}$$

When v finishes, attending to $D_{p,v}$ recovers both the parent p and the exact continuation threshold v.

Main result. For the standard edge-list encoding, we prove

$$\mathrm{TC}^{\mathrm{cBAPO}}_{\mathrm{REACH}}(m) = \Theta(m + 1), \tag{4}$$

using zero prefix bandwidth, at most one retrieved prefix token per step, and the uniform bound

$$c^{\mathrm{cBAPO}}_{0,1}(n, m) \le 3m + 2. \tag{5}$$

The construction is independent of n beyond the vocabulary needed to name vertices and edges. On sparse families with $m = \Theta(n)$ it gives $O(n)$ tokens for every instance, while the existing lower-bound family forces $\Omega(n)$.

Contributions. The paper provides the following results.

(i) A canonical top-one split-merge lemma recovers a global maximizer for every split and adversarial tie.

(ii) A predecessor-attention iterator enumerates the distinct present neighbors of u in $\deg^+(u) + 1$ queries rather than n equality probes.

(iii) A $(0, 1)$-cBAPO-CoT implements threaded DFS and halts within $3m + 2$ generated tokens, counting the answer but not HALT; throughout, m is the number of input edge tokens including repetitions.

(iv) The existing $\Omega(m)$ theorem yields the tight $\Theta(m + 1)$ characterization; moreover, no $(0, 0)$-BAPO-CoT can decide reachability.

(v) A certifying variant emits a simple reverse s–t path within $3m + 2$ tokens, and a product-state construction solves fixed-automaton regular-path reachability in at most $3|Q|m + 2$ tokens.

(vi) The literal arbitrary-oracle model admits noncomputable binary languages at bandwidth $(0, 0)$, while fixed finite-alphabet cBAPOs with total-recursive suffix oracles admit a direct Turing simulation.

(vii) A reference implementation validates all split and tie choices on 14,236 instances, including repeated edges and arbitrary edge order.

For diagnostic context only, Appendix G reports an illustrative single-seed study. The main result is theoretical: it does not claim that a trained softmax transformer will spontaneously learn the construction.

2 Related Work and Open-Problem Provenance

2.1 Transformer and CoT expressivity

Prior constructions show that Transformers can simulate rich computations under specified precision, positional, and architectural assumptions (Pérez et al., 2021; Bhattamishra et al., 2020; Giannou et al., 2023), while other restricted self-attention models have formal limitations (Hahn, 2020). CoT can expand expressive power by serializing intermediate states (Feng et al., 2023; Merrill & Sabharwal, 2024; Li et al., 2024); inductive scratchpads and globality barriers provide related accounts of explicit intermediate computation (Abbe et al., 2024). BAPO isolates how much prefix information can cross an arbitrary causal cut (Schnabel et al., 2025), and hard-attention lower bounds study a complementary formal model (Amiri et al., 2025). The cBAPO framework proves $\Omega(m)$ for reachability but gives an $O(n^2)$ direct construction (Tomlinson et al., 2026); our result answers its stated sparse-graph question with $O(m)$.

2.2 Reasoning efficiency

A growing literature measures or optimizes reasoning length through test-time scaling, compression, pruning, adaptive stopping, or length-aware training (Snell et al., 2025; Lee et al., 2025; Sui et al., 2025; Chen et al., 2025; Fan et al., 2025; Feng et al., 2026; Han et al., 2025; Jin et al., 2024; Kim et al., 2025; Liu et al., 2024; Luo et al., 2025; Nayab et al., 2024; Tang et al., 2026; Wu et al., 2026; Xia et al., 2025; Xu et al., 2025; Yi et al., 2025; Zabounidis et al., 2026). Those studies use varied empirical objectives and benchmark regimes. Our result instead concerns an exact worst-case asymptotic quantity in the cBAPO model, stated in Equation (4).

2.3 Graph search and regular paths

Classical DFS is linear in vertices plus edges and stores a visited set and a stack (Tarjan, 1972; Cormen et al., 2022). Our novelty is a representation theorem showing how score-BAPO realizes sparse DFS without copying either structure. The regular-path extension connects the construction to graph-database queries (Mendelzon & Wood, 1995; Barceló, 2013).

3 Model and Notation

3.1 Score-BAPO

Let $y = y_1 \cdots y_N \in \Gamma^N$ be a current context and let $0 \le k < N$ be an arbitrary split. The prefix is $y_{1:k}$ (empty when $k = 0$) and the nonempty suffix is $z = y_{k+1:N}$. A score-BAPO with prefix bandwidth a and attention bandwidth b consists of

$$f : \Gamma^* \to \{0,1\}^a, \tag{6}$$

$$g : \Gamma^* \times \mathbb{N} \times \Gamma \times \mathbb{N} \to [0,1], \tag{7}$$

$$h : \{0,1\}^a \times \bigcup_{r=0}^{b} (\Gamma \times \mathbb{N})^r \times \Gamma^* \times \mathbb{N} \to \Gamma. \tag{8}$$

For $i \le k$, write $g_i = g(z, k, y_i, i)$. The attended set may be any

$$G \in \arg \max_{\substack{S \subseteq \{(y_i,i) : i \le k,\ g_i > 0\} \\ |S| \le b}} \sum_{(y_i,i) \in S} g_i. \tag{9}$$

Ties are adversarial; G is passed to h in increasing position order. A next-token map $p' : \Gamma^* \to \Gamma$ is solved when

$$h(f(y_{1:k}), G, z, k) = p'(y) \tag{10}$$

for every k and valid G. A BAPO-CoT iterates

$$y^{(0)} = x, \qquad y^{(r+1)} = y^{(r)} p'(y^{(r)}), \tag{11}$$

until p' emits Halt, with the answer immediately before Halt. We impose

$$|\Gamma| = \Theta(|\Sigma|). \tag{12}$$

A cBAPO additionally requires f to be BAPO-consistent. Our construction uses $a = 0$ and $f(y) = \epsilon$.

3.2 Reachability encoding

For fixed $n \ge 1, m \ge 0$, an input is $x = e_1 \cdots e_m s t$, where $e_i = (u_i, v_i) \in [n]^2$ and $s, t \in [n]$. Thus m is the edge-token count, including repeated copies of the same ordered pair. The tagged input alphabet is

$$\Sigma_n = ([n] \times [n]) \dot\cup [n], \qquad |\Sigma_n| = n^2 + n. \tag{13}$$

Repeated tokens are removed semantically by

$$E_x = \{(u,v) : \exists i \le m,\ e_i = (u,v)\}. \tag{14}$$

Write $s \rightsquigarrow_x t$ when t is reachable from s in $([n], E_x)$, and let

$$\mathcal{I}_{n,m} = ([n] \times [n])^m \times [n] \times [n] \tag{15}$$

be the full edge-list domain, including repetitions. For fixed bandwidths a, b, let $\mathcal{B}_{a,b}^{\mathrm{cBAPO}}(n,m)$ be the cBAPO-CoTs that are correct on every $x \in \mathcal{I}_{n,m}$ and satisfy $|\Gamma_n| = \Theta(|\Sigma_n|)$, and define

$$c_{a,b}^{\mathrm{cBAPO}}(n,m) := \inf_{M \in \mathcal{B}_{a,b}^{\mathrm{cBAPO}}(n,m)} \sup_{x \in \mathcal{I}_{n,m}} T_M(x). \tag{16}$$

Here $T_M(x)$ is the worst-case number of generated tokens before HALT, over all splits and valid attended sets at every step; it includes the answer token immediately preceding HALT and excludes HALT itself. Following the constant-bandwidth convention of the prior work, $\mathrm{TC}_{\mathrm{REACH}}^{\mathrm{cBAPO}}(m) = \Theta(m+1)$ means that one fixed bandwidth pair gives the $O(m+1)$ upper bound, whereas for every fixed constant pair (a, b) the worst case $\sup_n c_{a,b}^{\mathrm{cBAPO}}(n,m)$ is $\Omega_{a,b}(m)$. The hidden constant in the lower bound may depend on (a, b).

4 A Canonical Top-One Split-Merge Lemma

Definition 1. *Fix $y \in \Gamma^N$ and a query state q readable from y_N. Each position has a score $\omega_q(y_i, i, N) \in [0,1]$ and semantic value $v_q(y_i, i, N) \in V$. The query is canonical if all positive global maximizers have one semantic value.*

Lemma 2. *For every canonical score query, the suffix side of a $(0,1)$-score-BAPO recovers the globally maximizing semantic value internally for every split $k < N$ and every valid top-one tie choice. If every score is zero, it recovers $\perp$.*

Proof. Set $g(z, k, y_i, i) = \omega_q(y_i, i, N)$, where $N = k + |z|$ and q is read from y_N. Let

$$M_P = \max(\{0\} \cup \{\omega_q(y_i, i, N) : i \le k\}), \tag{17}$$

$$M_S = \max(\{0\} \cup \{\omega_q(y_i, i, N) : i > k\}). \tag{18}$$

If $M_P > 0$, every valid G contains one prefix item of score M_P; otherwise $G = \varnothing$. The suffix oracle computes M_S, compares the two values, and selects an item of score $M = \max\{M_P, M_S\}$ when $M > 0$. Otherwise it uses $\perp$. Canonicality makes its semantic value independent of the split and tie breaking. □

The lemma yields the primitives Index(r), Exists(P), Rightmost(P), and MaxValue(P, κ) by using respectively the scores

$$\mathbf{1}[i = r], \quad \mathbf{1}[P(y_i)], \quad \mathbf{1}[P(y_i)]\frac{i}{N+1}, \quad \mathbf{1}[P(y_i)]\rho(\kappa(y_i)), \tag{19}$$

where κ takes values in a finite totally ordered set K and $\rho : K \to (0,1]$ is any strictly increasing map.

5 Sparse Predecessor Attention

For $u \in [n]$ and $j \in [n+1]$, define

$$\mathrm{pred}_{E_x}(u, j) = \begin{cases} \max\{v < j : (u, v) \in E_x\}, & \text{if nonempty,} \\ \perp, & \text{otherwise.} \end{cases} \tag{20}$$

Let $\alpha_n(v) = \frac{1}{2} + \frac{v}{2(n+1)}$. On a state containing (u, j), score an original edge token (u, v) with $v < j$ by $\alpha_n(v)$ and all other edge tokens by zero.

Lemma 3. *A $(0,1)$ score query returns $\mathrm{pred}_{E_x}(u, j)$ for every split, independent of edge-list order and duplicate edge tokens.*

Lemma 4. *Set $j_0 = n+1$ and, while $v_r = \mathrm{pred}_{E_x}(u, j_r) \neq \perp$, set $j_{r+1} = v_r$. If $N^+(u) = \{w_1 > \cdots > w_d\}$, then $(v_0, \ldots, v_{d-1}) = (w_1, \ldots, w_d)$ and $v_d = \perp$. Thus the iterator uses exactly $d+1$ queries and never probes an absent label.*

Proof. Equation (20) returns the greatest remaining neighbor. Strict threshold decrease removes both that endpoint and every repeated copy; induction gives the claim. □

6 The Threaded DFS cBAPO-CoT

6.1 CoT alphabet

Introduce tagged tokens

$$R_{s,t}: \text{ root and target record}, \qquad D_{p,v}: \text{ discovery record}, \tag{21}$$

$$S_{u,j}: \text{ resume below } j, \qquad C_{u,v}: \text{ check } u \to v, \tag{22}$$

together with Yes, No, and Halt. The full alphabet is

$$\Gamma_n = \Sigma_n \dot\cup \{R_{s,t}\} \dot\cup \{D_{p,v}\} \dot\cup \{S_{u,j}\} \dot\cup \{C_{u,v}\} \dot\cup \{\textsc{Yes}, \textsc{No}, \textsc{Halt}\}, \tag{23}$$

with indices in their natural ranges. Hence

$$|\Gamma_n| = 5n^2 + 2n + 3 = \Theta(n^2) = \Theta(|\Sigma_n|). \tag{24}$$

A record-token type for v belongs to the set

$$\mathrm{Rec}(v) = \{R_{v,t'} : t' \in [n]\} \cup \{D_{p,v} : p \in [n]\}, \tag{25}$$

with parent decoder $\pi(R_{v,t'}) = 0$ and $\pi(D_{p,v}) = p$. For a concrete context y, let $\mathrm{rec}_y(v)$ be its rightmost token in $\mathrm{Rec}(v)$, or $\perp$ if none exists. On legal rollout contexts Lemma 7 makes this the unique record of v; the rightmost convention also makes the oracle total on contexts containing conflicting records. Active tokens decode as

$$\mathrm{act}(R_{s,t}) = (s, n+1), \qquad \mathrm{act}(D_{p,u}) = (u, n+1), \tag{26}$$

$$\mathrm{act}(S_{u,j}) = (u, j). \tag{27}$$

6.2 Attention scores

If the last token is active with $\mathrm{act}(y_N) = (u, j)$, define

$$g_{\mathrm{act}}(y_i, i) = \begin{cases} \frac{1}{2} + \frac{v}{2(n+1)}, & i \le m,\ y_i = (u, v),\ v < j, \\ \frac{i}{4(N+1)}, & y_i \in \mathrm{Rec}(u), \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

The score bands are disjoint, so a qualifying edge always outranks a parent record; if no edge qualifies, the rightmost record is canonical. If the final token is $C_{u,v}$, define

$$g_{\mathrm{check}}(y_i, i) = \begin{cases} 1, & y_i = R_{a,v} \text{ for some } a, \\ \frac{1}{2}, & y_i \in \mathrm{Rec}(v), \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

On the initial context define $g_{\mathrm{init}}(y_i, i) = \mathbf{1}[i = m+1]$.

6.3 Suffix transition

Initialization emits Yes if $s = t$ and $R_{s,t}$ otherwise. For active A with $\mathrm{act}(A) = (u, j)$, let $v^* = \mathrm{pred}_{E_x}(u, j)$ and set

$$p'(yA) = \begin{cases} C_{u,v^*}, & v^* \neq \perp, \\ \textsc{No}, & v^* = \perp \text{ and } \pi(\mathrm{rec}_{yA}(u)) = 0, \\ S_{p,u}, & v^* = \perp \text{ and } \pi(\mathrm{rec}_{yA}(u)) = p \in [n]. \end{cases} \tag{30}$$

If $v^* = \perp$ and $\mathrm{rec}_{yA}(u) = \perp$, the totalized transition outputs No. An edge check transitions by

$$p'(yC_{u,v}) = \begin{cases} \textsc{Yes}, & v = t, \\ S_{u,v}, & v \neq t \text{ and } \mathrm{rec}_{yC_{u,v}}(v) \neq \perp, \\ D_{u,v}, & v \neq t \text{ and } \mathrm{rec}_{yC_{u,v}}(v) = \perp. \end{cases} \tag{31}$$

Finally $p'(y\textsc{Yes}) = p'(y\textsc{No}) = \textsc{Halt}$.

Algorithm 1: Semantic view of the generated CoT

1. If $s = t$, emit YES and stop; otherwise emit $R_{s,t}$.
2. Decode the last active token as (u, j) and set $v \leftarrow \mathrm{pred}_E(u, j)$.
3. If $v \neq \perp$, emit $C_{u,v}$. Then emit YES if $v = t$; otherwise emit $S_{u,v}$ if $\mathrm{rec}(v) \neq \perp$, and $D_{u,v}$ if not.
4. If $v = \perp$, retrieve $\mathrm{rec}(u)$. Emit NO if it is the root record; otherwise, if its parent is p, emit $S_{p,u}$.
5. Repeat from step 2 until an answer is emitted; the next token is HALT.

Theorem 5. *The preceding functions, totalized as specified in Appendix A, define a* $(0, 1)$*-cBAPO-CoT over* Γ_n.

Proof. The prefix oracle is empty and self-consistent. Initialization uses $\mathrm{Index}(m + 1)$. In an active state, edge and record scores lie in disjoint bands; duplicates of the maximum edge have the same semantics, while absent edges expose the unique rightmost record. Lemma 2 therefore makes Equation (30) split invariant. For $C_{u,v}$, all score-one maximizers mean "target" and all score-one-half maximizers mean "visited," making Equation (31) split invariant. The alphabet condition follows from the count above. □

7 CORRECTNESS

Definition 6. *At time r, let* $V_r = \{v : \mathrm{rec}_{y^{(r)}}(v) \neq \perp\}$ *and define* $P_r(s) = 0$*, while* $P_r(v) = p$ *iff* $\mathrm{rec}_{y^{(r)}}(v) = D_{p,v}$.

Lemma 7. *Every vertex has at most one record on a valid trace. Every nonroot record* $D_{p,v}$ *satisfies* $p \in V_r$ *and* $(p, v) \in E_x$*. Once* $R_{s,t}$ *is emitted, parent records form a directed tree rooted at s over* V_r.

Proof. A nonroot record is emitted only after an active query selected (u, v) and the check query found no existing record of v. Parent pointers strictly follow discovery time, so they are acyclic. □

Lemma 8. *Whenever the final token is active with state* (u, j)*: (a)* $u \in V_r$*; (b) every distinct out-neighbor* $w \geq j$ *has already been checked; and (c) no distinct out-neighbor* $w < j$ *has yet been checked from u.*

Proof. A first activation has threshold $n + 1$ and no earlier outgoing check. After selecting $v = \mathrm{pred}_{E_x}(u, j)$, either the search succeeds before returning to u, or the next activation of u is $S_{u,v}$. Maximality leaves no unchecked neighbor in $[v, j)$; induction preserves (a)–(c). □

Lemma 9. *If a nonroot current vertex u finishes and its record is* $D_{p,u}$*, then* $S_{p,u}$ *is exactly the suspended parent state that decreasing-label DFS must resume. At each active or check state, the unexhausted records are precisely the current vertex and its ancestors.*

Proof. Initially only the root is unexhausted. A discovery appends a new child to the current chain; a visited check leaves it unchanged; exhaustion removes its last vertex. Since $D_{p,u}$ was created when p selected u, Lemma 8 gives the continuation $S_{p,u}$. Induction proves both claims. □

Lemma 10. *For every* $(u, v) \in E_x$*, the trace contains at most one* $C_{u,v}$.

Proof. After $C_{u,v}$, either u is never active again, or its next activation has threshold v and all later queries require a destination strictly smaller than v. □

Theorem 11. *The threaded DFS emits* YES *iff* $s \rightsquigarrow_x t$*; otherwise it emits* NO*. It halts on every input.*

Proof. If $s = t$, the empty path is valid. Otherwise YES is emitted only from $C_{u,t}$; Lemma 7 makes the record chain from s to u a path of input edges, and $C_{u,t}$ supplies its last edge.

For completeness, suppose the root is exhausted but t is reachable. On a shortest s–t path choose the first vertex not in the final discovered set. Its predecessor is discovered, and Lemmas 4 and 8 force that outgoing endpoint to be checked, either producing YES or a discovery record—a contradiction.

Termination follows because check and discovery tokens are unique, thresholds strictly decrease, and exhausted vertices move up the finite parent chain. □

8 Exact Token Complexity

Let e be the number of distinct examined directed edges, r the number of records, and q the number of exhausted vertices. By Lemma 10, $e \le |E_x| \le m$. Every examined edge contributes exactly two tokens, while every exhausted vertex contributes one and initialization contributes one.

Theorem 12. *The construction emits at most $3m + 2$ nonhalt CoT tokens; if $s \neq t$ and the answer is* YES*, at most $3m$.*

Proof. If $s = t$, the trace is the one-token answer. On a no-instance, $q = r$ and every nonroot record is created by a distinct examined edge, so $r - 1 \le e$ and

$$T_{\text{no}} = 1 + 2e + r \le 3e + 2 \le 3m + 2. \tag{32}$$

On a yes-instance with $s \neq t$, the terminal edge creates no record and the root is unexhausted, so $r \le e$ and $q \le r - 1$, whence

$$T_{\text{yes}} = 1 + 2e + q \le 3e \le 3m. \tag{33}$$

□

Proposition 13. *For every fixed constant bandwidth pair (a, b), the $\Omega(m)$ reachability lower bound of Tomlinson et al. (2026) applies to $\mathcal{I}_{n,m}$ above: for all sufficiently large m, there is $n(m) = m + O_a(1)$ such that*

$$c^{\text{cBAPO}}_{a,b}(n(m), m) = \Omega_{a,b}(m), \qquad \textit{and hence} \qquad \sup_n c^{\text{cBAPO}}_{a,b}(n, m) = \Omega_{a,b}(m). \tag{34}$$

Proof. Definitions 2.6 and 3.1 of arXiv:2602.02909v2 use our input, alphabet, and pre-HALT conventions. Its Theorem 4.3 uses $p = 2a + 4$ layered paths. Appendix C gives the pointwise argument, retaining the current CoT in every attention query and padding nondivisible m by isolated loops. The resulting inputs lie in $\mathcal{I}_{n,m}$ with $n = m + O_a(1)$. The lower bound holds for BAPO-CoTs and hence for the cBAPO subclass. □

Corollary 14. *Under $|\Gamma| = \Theta(|\Sigma|)$, $c^{\text{cBAPO}}_{0,1}(n, m) \le 3m + 2$ for every (n, m). Together with Proposition 13, this yields $\mathrm{TC}^{\text{cBAPO}}_{\text{REACH}}(m) = \Theta(m + 1)$ under the same fixed-bandwidth quantifiers as the prior lower bound.*

8.1 Attention bandwidth one is zero-prefix optimal

Proposition 15. *No $(0, 0)$-BAPO-CoT decides directed reachability, even on two vertices and one edge.*

Proof. Compare $x^+ = ((1, 2), 1, 2)$ with $x^- = ((1, 1), 1, 2)$. At the split immediately before the final token, both suffixes and split indices agree, while the prefix oracle and attended set are empty. Thus the first generated token is identical. Repeating the argument before each last token makes the complete traces identical, contradicting their different answers. □

8.2 Sparse inference consequence

With $N_0 = m + 2$ and $T \le 3m + 2$, Equation (1) gives the following count of token–key score evaluations for the nonhalt outputs in the stated dense-attention accounting:

$$\mathcal{A}(N_0, T) \le (3m + 2)(m + 2) + \frac{(3m + 2)(3m + 1)}{2} \tag{35}$$

$$= \frac{15}{2}m^2 + \frac{25}{2}m + 5. \tag{36}$$

Including the final HALT call adds $N_0 + T$. Under the same accounting, a $\Theta(n^2)$-token trace on a sparse $m = \Theta(n)$ family would induce $\Theta(n^4)$ cumulative score work. This comparison is not a wall-clock latency claim: kernels, batching, retrieval overhead, and memory traffic are not modeled.

9 A Linear-Token Path Certificate

Add tokens $W_{p,v}$, certifying a selected path edge $p \to v$, and B_v, marking the root. Modify the target branch to emit $W_{u,t}$. On $W_{p,v}$ retrieve the unique record of p and emit

$$p'(yW_{p,v}) = \begin{cases} B_p, & \mathrm{rec}_{yW_{p,v}}(p) = R_{p,t'}, \\ W_{q,p}, & \mathrm{rec}_{yW_{p,v}}(p) = D_{q,p}, \end{cases} \qquad p'(yB_s) = \textsc{Yes}. \tag{37}$$

Theorem 16. *On every yes-instance with $s \neq t$, let $v_0 = s \to \cdots \to v_\ell = t$ be the simple parent-chain path found by the decreasing-label threaded DFS. A $(0,1)$-cBAPO-CoT emits*

$$W_{v_{\ell-1},v_\ell}, W_{v_{\ell-2},v_{\ell-1}}, \ldots, W_{v_0,v_1}, B_{v_0}, \textsc{Yes}. \tag{38}$$

When $s = t$, it emits B_s, Yes. *The total nonhalt count, including* Yes *and excluding* Halt*, is at most $3m + 2$, also on no-instances. Appendix D gives the complete token and transition definitions.*

Proof. Parent queries strictly decrease discovery time, so the path is simple. At success, the ℓ recorded ancestors are exactly the unexhausted vertices, hence $r = q + \ell$. Replacing direct Yes adds $\ell + 1$ tokens, giving $T = 2 + 2e + q + \ell = 2 + 2e + r \le 3m + 2$. The empty path uses two tokens. Appendix D gives equality cases. □

10 Regular-Path Reachability

Let $\mathcal{A} = (Q, \Lambda, \delta, q_0, F)$ be a fixed DFA, with $q = |Q|$ and $d = |\Lambda|$ constants. Input edges are $(u, \lambda, v) \in [n] \times \Lambda \times [n]$. Regular reachability asks for an accepted labeled walk, allowing repeated vertices. Use the implicit product graph $\widehat{V} = [n] \times Q$ with

$$(u, r) \longrightarrow (v, r') \iff \exists \lambda \in \Lambda : \ (u, \lambda, v) \in E_x \text{ and } r' = \delta(r, \lambda). \tag{39}$$

Fix an injective key $\kappa : \widehat{V} \to [nq]$. At (u, r) score an admissible input edge by

$$\widehat{\alpha}(v, \delta(r, \lambda)) = \frac{1}{2} + \frac{\kappa(v, \delta(r, \lambda))}{2(nq + 1)}. \tag{40}$$

Different labeled edges mapping to the same product child tie with the same semantics. The root is (s, q_0), and (v, r') is accepting exactly when $v = t$ and $r' \in F$.

Theorem 17. *For every fixed DFA $\mathcal{A}$ with q states, regular-path reachability on an m-edge-token labeled digraph is solved by a $(0,1)$-cBAPO-CoT with $|\Gamma| = \Theta(|\Sigma|)$ and $T \le 3qm + 2$. For every fixed $q \ge 1$, some q-state DFA has worst-case token complexity $\Theta(m + 1)$. Complete transitions, the empty-path case, multiple accepting states, and product-neighbor ties are specified in Appendix D.*

Proof. Run the threaded construction on the implicit product graph. For a fixed automaton state, every original edge contributes at most one distinct product candidate, so at most qm product edges are examined. The lower bound follows from a one-label DFA whose accepting start state self-loops, augmented by $q - 1$ unreachable states. □

11 Constant Alphabets and Turing Simulation

Theorem 18. *For every $A \subseteq \mathbb{N}_{\ge 2}$, the language $L_A = \{x \in \{0,1\}^* : |x| \in A\}$ is recognized by a $(0,0)$-cBAPO. Consequently, some constant-alphabet, zero-bandwidth cBAPOs are not simulable by any Turing machine.*

Proof. Let $f(x) = \epsilon$, $g \equiv 0$, and define the arbitrary suffix oracle $h_A(\epsilon, \varnothing, z, k) = \mathbf{1}[k + |z| \in A]$. Since $k + |z| = |x|$ at every nonempty-input split, the output is split invariant; set the output on ϵ to 0. Choosing a nonrecursive A gives the claim. □

Corollary 19. *There are $2^{\aleph_0}$ zero-bandwidth cBAPO-recognizable languages over $\{0,1\}$ but only $\aleph_0$ Turing-decidable languages.*

Table 1: Exact and randomized verification.

Test family	Instances
All simple digraphs, $n = 1$	2
All simple digraphs, $n = 2$	108
All simple digraphs, $n = 3$	9,126
Random multigraphs, $n \le 12, m \le 50$	5,000
Total	14,236

Definition 20. *A score-cBAPO is effective if f and h are total recursive and $g : \Gamma^* \times \mathbb{N} \times \Gamma \times \mathbb{N} \to \mathbb{Q} \cap [0, 1]$ is total recursive.*

Theorem 21. *Every fixed finite-alphabet score-BAPO with total-recursive suffix oracle h is Turing-simulable, without computability assumptions on f or g. In particular, every fixed effective BAPO-CoT is simulable token by token on every input on which it halts.*

Proof. For nonempty y, choose $k = 0$: correctness gives $p'(y) = h(f(\epsilon), \varnothing, y, 0)$. Hard-code the fixed finite string $f(\epsilon)$ and the token $p'(\epsilon)$; then evaluate h and iterate. No score is evaluated. This is a fixed-machine simulation, not a uniform claim about arbitrary nonuniform families. □

12 Executable Split-by-Split Verification

The exact transition system was implemented in `verify_bapo.py` using integer-scaled rational scores. It uses the original input alphabet: both s and t are tokens of the same node type and are distinguished only by their positions after the m edges. For every rollout context it executes all splits $k \in \{0, \ldots, |y| - 1\}$ and all valid adversarial top-one prefix choices; it additionally enumerates all equal-scoring suffix maximizers. It then checks that the answer agrees with independent BFS, $T \le 3m + 2$, and the trace ends in YES or NO. Every simple graph with more than one edge is tested in canonical and reversed edge order. Random tests include shuffled order, self-loops, and repeated edge tokens. The score routine accepts only (z, k, y_i, i), while the suffix transition accepts only (G, z, k); neither interface accepts the hidden prefix. The implementation passed all 14,236 cases after this information-flow restriction was enforced structurally rather than by convention. Appendix B specifies the corpus, assertions, and reproduction commands for the anonymized supplementary archive.

13 Discussion

13.1 Why the result is sparse-sensitive

The prior $O(n^2)$ approach represents graph-search state as a dense vector over the node universe. Ours represents only discovery records and distinct examined-edge checks. Since every nonroot discovery is charged to an edge,

$$|V_{\text{discovered}}| \le |E_{\text{examined}}| + 1, \tag{41}$$

and the accounting has no independent n term.

13.2 Interpretation for LLM reasoning systems

Within the formal construction, three mechanisms are responsible for the bound: selecting the next present relation instead of incrementing through absent labels, writing each discovery record once, and storing the parent continuation in that record. Whether analogous mechanisms improve a learned system requires separate empirical evidence.

13.3 Limitations

The theorem concerns cBAPO, not end-to-end learnability. It assumes exact scores and positions, nonuniformity in (n, m), a $\Theta(n^2)$ vocabulary, and an unbounded suffix oracle; it establishes neither finite-precision or softmax robustness nor immunity to tokenization or learned-retrieval errors.

AI USE STATEMENT

We used generative AI tools to assist with drafting and reviewing experimental code, clarifying mathematical notation and proof exposition, organizing supplementary material, and editing the manuscript for clarity and LaTeX formatting. AI tools also provided feedback on the description of experimental methods and the interpretation of results, and assisted in formulating and checking the sharpened token bounds and completing proof details. The authors reviewed the AI-assisted material and take responsibility for the final manuscript, proofs, code, and reported results.

REFERENCES

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, Colin Sandon, and Omid Saremi. How far can transformers reason? the globality barrier and inductive scratchpad. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0874.

Alireza Amiri, Xinting Huang, Mark Rofin, and Michael Hahn. Lower bounds for chain-of-thought reasoning in hard-attention transformers. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.02393.

Pablo Barceló. Querying graph databases. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 175–188, 2013. doi: 10.1145/2463664.2465216.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Höfler. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17682–17690, 2024. doi: 10.1609/aaai.v38i16.29720.

Satwik Bhattamishra, Arkil Patel, and Navin Goyal. On the computational power of transformers and its implications in sequence modeling. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pp. 455–475, 2020. doi: 10.18653/v1/2020.conll-1.37.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? on the overthinking of O1-like LLMs. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2412.21187.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 4th edition, 2022.

Siqi Fan, Bowen Qin, Peng Han, Shuo Shang, Yequan Wang, and Aixin Sun. The price of a second thought: On the evaluation of reasoning efficiency in large language models, 2025. URL https://arxiv.org/abs/2505.22017. arXiv preprint arXiv:2505.22017.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.15408.

Yunzhen Feng, Julia Kempe, Cheng Zhang, Parag Jain, and Anthony Hartshorn. What characterizes effective reasoning? revisiting length, review, and structure of CoT. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.19284.

Angeliki Giannou, Shashank Rajput, Jy yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 2023. URL https://proceedings.mlr.press/v202/giannou23a.html.

Michael Hahn. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171, 2020. doi: 10.1162/tacl_a_00306.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24842–24855, 2025. doi: 10.18653/v1/2025.findings-acl.1274.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. URL https://arxiv.org/abs/2401.04925.

Juno Kim, Denny Wu, Jason D. Lee, and Taiji Suzuki. Metastable dynamics of chain-of-thought reasoning: Provable benefits of search, RL and distillation. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.01694.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22199–22213, 2022.

Ayeong Lee, Ethan Che, and Tianyi Peng. How well do LLMs compress their own chain-of-thought? a token complexity approach, 2025. URL https://arxiv.org/abs/2503.01141. arXiv preprint arXiv:2503.01141.

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. URL https://arxiv.org/abs/2402.12875.

Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. Can language models learn to skip steps? In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2411.01855.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for O1-like reasoning pruning, 2025. URL https://arxiv.org/abs/2501.12570. arXiv preprint arXiv:2501.12570.

Alberto O. Mendelzon and Peter T. Wood. Finding regular simple paths in graph databases. In *Proceedings of the 21st International Conference on Very Large Data Bases*, pp. 185–196, 1995.

William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.07923.

Sania Nayab, Giulio Rossolini, Marco Simoni, Andrea Saracino, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. Concise thoughts: Impact of output length on LLM reasoning and cost, 2024. URL https://arxiv.org/abs/2407.19825. arXiv preprint arXiv:2407.19825.

Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is Turing-complete. *Journal of Machine Learning Research*, 22(75):1–35, 2021.

Tobias Schnabel, Kiran Tomlinson, Adith Swaminathan, and Jennifer Neville. Lost in transmission: When and why LLMs fail to reason globally. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2505.08140.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.03314.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research*, 2025. URL https://arxiv.org/abs/2503.16419.

Yuntian Tang, Bohan Jia, Wenxuan Huang, Lianyue Zhang, Jiao Xie, Wenxi Li, Wei Li, Jie Hu, Xinghao Chen, Rongrong Ji, and Shaohui Lin. Towards efficient large language reasoning models via extreme-ratio chain-of-thought compression. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.08324.

Robert E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2): 146–160, 1972. doi: 10.1137/0201010.

Kiran Tomlinson, Tobias Schnabel, Adith Swaminathan, and Jennifer Neville. Reasoning about reasoning: BAPO bounds on chain-of-thought token complexity in LLMs. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.02909v2. arXiv:2602.02909v2, revised 27 May 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in LLMs. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2502.07266.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. TokenSkip: Controllable chain-of-thought compression in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3351–3363, 2025. doi: 10.18653/v1/2025.emnlp-main.165.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less, 2025. URL https://arxiv.org/abs/2502.18600. arXiv preprint arXiv:2502.18600.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jingyang Yi, Jiazheng Wang, and Sida Li. ShorterBetter: Guiding reasoning models to find optimal inference length for efficient reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2504.21370.

Renos Zabounidis, Aditya Golatkar, Michael Kleinman, Alessandro Achille, Wei Xia, and Stefano Soatto. Re-FORC: Adaptive reward prediction for efficient chain-of-thought reasoning. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2511.02130.

A Complete Oracle Specification

Fix (n, m). This section defines total oracles on all contexts, not only those reached from a valid input. The prefix oracle is $f(w) = \epsilon$. Consequently it reveals neither the prefix nor any validity bit. Set $g(z, k, \gamma, i) = 0$ when $z = \epsilon$ or $i \notin [1, k + |z|]$, and set $h = \text{NO}$ on syntactically malformed arguments. The score oracle receives only (z, k, y_i, i), where $z \neq \epsilon$, $N = k + |z|$, and the last token $z_{|z|} = y_N$ determines the query type. It assigns the scores of Section 6.2: position $m + 1$ for initialization; the disjoint edge and rightmost-record bands for an active token; and the target/visited bands for a check token. For an answer, HALT, or any other final-token type it assigns zero everywhere. In particular, it never reads $y_{1:k}$ except through the single candidate (y_i, i) whose score it is asked to evaluate.

The suffix oracle receives only (G, z, k). It recomputes the scores of all suffix positions, compares their maximum with the score of the sole element of G when $G \neq \emptyset$, and decodes a global maximizer. Equal maximizers are handled as follows: repeated copies of the maximum edge all decode to the

same $C_{u,v}$; all target-record maximizers decode to YES; and all visited-record maximizers decode to the same $S_{u,v}$. The rightmost-record score is strict in the position, so conflicting parent records have a unique global maximizer. If every score is zero, the candidate is $\perp$. This is precisely the split–merge operation of Lemma 2, implemented using only the permitted interface.

For completeness, the transition table is:

1. *Initialization.* If the last token is a node token, recover position $m + 1$. If it contains a node s and the last token contains t, emit YES when $s = t$ and $R_{s,t}$ otherwise. If the position is absent or is not a node token, emit NO.
2. *Active.* For final token A with $\mathrm{act}(A) = (u, j)$, an edge maximizer (u, v) emits $C_{u,v}$. If no edge qualifies, a root record emits NO and a discovery record $D_{p,u}$ emits $S_{p,u}$. If neither edge nor record exists, emit NO.
3. *Check.* For final token $C_{u,v}$, a matching target record $R_{a,v}$ emits YES; otherwise any record in $\mathrm{Rec}(v)$ emits $S_{u,v}$; if no such record exists, emit $D_{u,v}$.
4. *Answer and fallback.* After YES or NO, emit HALT. After HALT, emit HALT. For every remaining final-token type, emit NO. Define $p'(\epsilon) = \text{NO}$, although no legal BAPO split queries the empty context.

On legal inputs, the first transition is applied to exactly $e_1 \cdots e_m s t$. When $m = 0$, initialization is unchanged and the first active query either immediately answers NO or the $s = t$ branch has already answered YES. Repeated edges only create equal-scoring copies with identical decoded semantics. Empty prefixes give $G = \varnothing$ and are merged with the visible suffix maximum in the same way as every other split.

The totalization does not rely on declaring malformed contexts invalid inside a hidden prefix. Every fallback is determined by the visible last token, k, G, and suffix maxima. Conflicting records are resolved by the globally rightmost position; missing records use the all-zero case; and check-state conflicts collapse to the target/visited semantic classes above. Hence each case has one decoded semantic output for every split and every valid top-one tie choice. On reachable contexts, Lemma 7 removes the conflict cases and the table reduces to Equations (30)–(31). This proves split invariance for both legal rollouts and the stated total extension.

B VERIFICATION AND REPRODUCIBILITY DETAILS

An anonymized code archive is included in the supplementary material. The accompanying README provides dependencies, execution commands, expected running times, and instructions for reproducing the verification and experimental results. The verifier uses only the Python standard library and exact integer-scaled rational scores; its fixed random seed is 20260829. The 14,236-instance run invokes `python verify/verify_bapo.py` with arguments `--mode full` and `--json-output verify/verification_results.json`.

The exhaustive corpus enumerates every subset of the n^2 possible ordered pairs, including self-loops, and every $(s, t) \in [n]^2$. Edge subsets of size zero or one use their sole canonical order; every larger subset is tested in canonical and reverse order. Thus the counts are 2 for $n = 1$, 108 for $n = 2$, and 9,126 for $n = 3$. The randomized corpus adds 5,000 instances: n is uniform on $[1, 12]$, m is uniform on $[0, 50]$, each endpoint and (s, t) are sampled uniformly, and the edge-token list is shuffled. Sampling with replacement permits repeated edges; self-loops and the cases $m = 0$ and $s = t$ are included. The total is therefore $2 + 108 + 9{,}126 + 5{,}000 = 14{,}236$.

At every generated context, including the call that emits HALT, the verifier enumerates all $k \in \{0, \ldots, |y| - 1\}$. For each split it enumerates every equal-scoring positive prefix maximizer allowed by top-one attention, and, as a strengthening, every equal-scoring positive suffix maximizer. It asserts that all such choices and all splits produce one next token. Across the complete corpus this checks 2,479,232 split contexts and 2,987,909 tie outcomes.

Each rollout is also checked against an independently implemented BFS on the semantic set E_x, with duplicate tokens removed only for BFS adjacency. The verifier asserts termination, a final token in $\{\text{YES}, \text{NO}\}$ immediately before HALT, agreement with BFS, and $T \le 3m + 2$ where T includes the answer and excludes HALT. The recorded run passed all assertions; observed T ranged from 1 to 88

with mean 5.3953. Interface signatures enforce the information restriction: the score routine accepts only (z, k, y_i, i) and the suffix routine only (G, z, k), so neither can inspect the hidden prefix.

C Compatibility with the Existing Lower Bound

The comparison is to the accepted ICML 2026 paper *Reasoning about Reasoning: BAPO Bounds on Chain-of-Thought Token Complexity in LLMs*, arXiv:2602.02909v2 (27 May 2026) (Tomlinson et al., 2026). Its Definition 2.6 encodes reachability as $([n] \times [n])^m \times [n] \times [n]$: an arbitrarily ordered list of m edge tokens followed by s, t. This is exactly $\mathcal{I}_{n,m}$; our allowance of repeated pairs states explicitly what the Cartesian-power domain already permits.

Its Definition 3.1 minimizes over machines separately for each size, imposes $|\Gamma| = \Theta(|\Sigma|)$, maximizes over inputs and over split/attention choices, and counts generated tokens before the halt token. Our T uses the same convention: the answer is counted and HALT is not. Theorem 4.3 states that every constant-bandwidth BAPO-CoT for m-edge reachability needs $\Omega(m)$ tokens. Because cBAPO-CoTs form a subclass of BAPO-CoTs, any such lower bound also applies to our upper-bound class. We spell out the adaptive argument with the current CoT included in every score query.

Fix a and $b \geq 1$, put $p = 2a + 4$, $d = \lfloor m/p \rfloor$, and $m_0 = pd$. Take $d + 1$ layers of p vertices each. Between successive layers choose a bijection; its edges form p disjoint directed paths from the first layer $s_1, \ldots, s_p$ to the last layer $t_1, \ldots, t_p$. Each nonterminal vertex u has a fixed input position $I(u)$ containing its outgoing edge. Append $m - m_0$ copies of a self-loop at one additional isolated vertex. There are exactly m edge tokens and $n = m_0 + p + 1 = m + O_a(1)$ vertices. Padding never affects a query $s_i t_j$.

Suppose a correct machine uses at most T nonhalt tokens on every such input. Maintain partial bijections that always admit completion; padding tokens are fixed from the start. For each summary $\sigma \in \{0, 1\}^a$ and each query $s_i t_j$, simulate at most $T + 1$ next-token calls, stopping at HALT. Reset the simulated CoT z at the start of each simulation. At each call use the cut $k = m$ and the actual current suffix $w = s_i t_j z$.

To choose G, greedily select up to b distinct positive-score positions. At each selection maximize $g(w, m, e, i)$ over still-unselected positions i and edge values e consistent with some completion of the current partial bijections, also including the fixed padding positions. If the selected position is not fixed, commit its selected edge. Stop if no positive candidate remains. Then append $h(\sigma, G, w, m)$ to z. Each call fixes at most b new core edges. Its selected G is valid in every subsequent completion: each surviving competitor was feasible when a selected maximizer was chosen and had no larger score. If fewer than b positions were selected, all remaining feasible scores were zero. Adversarial ties permit the same selected G in every completion.

After all simulations, the number of fixed core edges is at most

$$K(T + 1), \qquad K = 2^a p^2 b.$$

If $K(T + 1) < m_0/2$, some layer has at least $p/2 = a + 2$ unassigned sources and equally many unused destinations. Only now complete the other layers, leaving these unmatched positions free. Permuting the remaining destinations at this layer produces at least $(a + 2)! > 2^a$ full inputs with pairwise distinct source–target reachability relations. Two have the same actual prefix summary $\sigma = f(e_1 \cdots e_m)$ but disagree on some query $s_i t_j$. All selected edges and padding positions agree on these two inputs. The stored simulation for this (σ, i, j) is therefore a valid, identical rollout on both: inductively its suffix, chosen G, and next token agree. Within $T + 1$ calls it must include the answer and HALT, contradicting the differing answers.

Consequently every correct machine satisfies

$$T \geq \frac{m_0}{2^{a+1} p^2 b} - 1 = \Omega_{a,b}(m)$$

for all sufficiently large m, not merely along multiples of p. The argument is pointwise in the chosen machine and size, so it survives the infimum in Proposition 13. For $b = 0$, a zero-attention machine is also a bandwidth-one machine by taking all scores to be zero, and that lower bound applies. No effectiveness assumption or restriction on how f depends on the edge list is used.

D Complete Certificate and Regular-Path Extensions

D.1 Path certificate

Extend Γ_n by $W_{p,v}$ for $(p, v) \in [n]^2$ and B_v for $v \in [n]$. The new size is $6n^2 + 3n + 3 = \Theta(|\Sigma_n|)$. Replace only the successful target transition $C_{u,t} \mapsto \textsc{Yes}$ by $C_{u,t} \mapsto W_{u,t}$. When the last token is $W_{p,v}$, use the rightmost-record score on $\mathrm{Rec}(p)$. A root record $R_{p,t'}$ emits B_p, a discovery record $D_{q,p}$ emits $W_{q,p}$, and a missing record emits No. Finally $B_s \mapsto \textsc{Yes}$ and $\textsc{Yes} \mapsto \textsc{Halt}$. For $s = t$, initialization emits B_s rather than Yes.

Thus the emitted W tokens are exactly the reverse of the parent chain created by this DFS, not a user-chosen or arbitrary simple path. Each consecutive pair is an input edge, discovery time strictly decreases along the reverse chain, and the chain ends at s. If its length is ℓ, the target has no record and the unexhausted recorded vertices are exactly the ℓ ancestors from s to the final current vertex. Thus $r = q + \ell$ and

$$T_{\mathrm{cert}} = (1 + 2e + q) + (\ell + 1) = 2 + 2e + r \le 3e + 2 \le 3m + 2.$$

Equivalently, each examined edge contributes its check and response, and each recorded vertex contributes exactly one additional transition: exhaustion if it is off the final chain, or a parent-query output if it is on the chain. Initialization and the final Yes contribute the remaining two tokens. The empty path uses B_s, Yes and has $T = 2$; no-instance behavior and the $3m + 2$ bound are unchanged.

The bounds are attained by these constructions. An m-edge directed chain with an isolated target uses $3m + 2$ decision tokens. For $m \ge 2$, let the root first explore a dead chain of $m - 1$ edges and then a direct edge to a smaller-labeled target; for $m = 1$, take just the target edge. Here $e = r = m$ and $q = m - 1$, giving $T_{\mathrm{yes}} = 3m$. An m-edge simple chain from s to t has $e = r = \ell = m$ and $q = 0$, so its certifying trace uses exactly $3m + 2$ tokens. For $m = 0$, the empty-path certificate has two tokens. These equality claims concern the stated constructions, not the optimal constant over all cBAPOs.

D.2 Regular-path reachability

If $\Lambda = \varnothing$, legal inputs have $m = 0$. A source-index query outputs Yes exactly when $s = t$ and $q_0 \in F$, otherwise No, followed by Halt. This uses one nonhalt token and $|\Gamma| = n + 3 = \Theta(|\Sigma|)$. Below assume $|\Lambda| \ge 1$.

For fixed $\mathcal{A} = (Q, \Lambda, \delta, q_0, F)$, replace every vertex field in R, D, S, C by a product vertex $(v, r) \in [n] \times Q$; write the root as $\widehat{R}_{(s,q_0),t}$ so that it additionally stores the target t. Initialization answers Yes exactly when $s = t$ and $q_0 \in F$. Otherwise it creates the product root (s, q_0). An active product state (u, r) scores each labeled input edge (u, λ, v) by the key of its product child $(v, \delta(r, \lambda))$ and uses the same decreasing-threshold iterator, record check, discovery, and parent-resume transitions as before.

Let $\widehat{\mathrm{Rec}}(v, r')$ be the root-or-discovery record class of product vertex (v, r'). For a final check token whose child is (v, r'), use the single top-one query

$$g^{\mathcal{A}}_{\mathrm{check}}(y_i, i) = \begin{cases} 1, & y_i = \widehat{R}_{(a,r_0),t'} \text{ for some } (a, r_0),\ v = t',\ r' \in F, \\ \frac{1}{2}, & y_i \in \widehat{\mathrm{Rec}}(v, r'), \\ 0, & \text{otherwise.} \end{cases} \tag{42}$$

Thus an accepting target match outranks visitedness and emits Yes; otherwise a product record emits the resume token, and no positive candidate emits the discovery token. The final check token supplies (v, r'), while the retrieved root token supplies t', so the test uses no extra retrieval. It also handles any number of accepting states.

Different labeled edges that induce the same product child are semantic ties and decode identically; distinct product children have distinct κ keys. Each of the m input edge tokens induces at most one product edge for each $r \in Q$, so the number of distinct examined product edges is at most qm, even with repeated input tokens. Applying the main accounting gives $T \le 3qm + 2$. With fixed q and $|\Lambda|$, the input alphabet and the extended token alphabet are both $\Theta(n^2)$ (constants may depend on q and $|\Lambda|$), preserving $|\Gamma| = \Theta(|\Sigma|)$.

E Worked Trace

Consider $E = \{(1,4),(1,2),(4,4),(2,3),(3,5)\}$ and $(s,t) = (1,5)$. Decreasing-label DFS explores the dead branch through 4 and backtracks without an explicit pop token:

$$R_{1,5} \to C_{1,4} \to D_{1,4} \to C_{4,4} \to S_{4,4} \to S_{1,4} \\ \to C_{1,2} \to D_{1,2} \to C_{2,3} \to D_{2,3} \to C_{3,5} \to \textsc{Yes}. \quad (43)$$

The transition $S_{4,4} \to S_{1,4}$ occurs because no edge from 4 has destination below 4 and the lower score band retrieves $D_{1,4}$. The trace has 12 tokens, below $3m + 2 = 17$.

F Impact Statement

This work advances theoretical understanding of inference-time reasoning efficiency. A tighter upper bound can motivate shorter and more structured traces, reducing computation and energy use. Conversely, exact retrieval should not be interpreted as a guarantee that deployed LLMs will reliably solve safety-critical graph-reasoning tasks. Practical use should retain explicit verification of emitted path certificates.

G Illustrative Single-Seed Study

This experiment is descriptive support for one mechanism, not a contribution to the asymptotic theorem. We evaluate the step-60,000 `latest.pt` checkpoints of three six-layer Transformers trained from scratch on deterministic online graphs with seed 42. Step 60k was the common budget endpoint fixed by the training protocol, not a validation-selected optimum; no test suite is used for checkpoint selection. *Retrieval soft* (12.80M parameters) receives 10k oracle-retrieval warm-up steps followed by 50k soft-retrieval steps and is evaluated with hard argmax retrieval. *Vanilla threaded* and *vanilla answer-only* (12.60M parameters each) use unrestricted causal attention and predict, respectively, the canonical DFS trace or only the answer and halt. All models have width 256, eight heads, FFN width 1024, maximum length 2304, and a composite $Q_{s,t}$ query token. Training uses AdamW for 60k updates, a 32,768-token global batch budget, a cosine learning-rate schedule from 3×10^{-4} to 3×10^{-5} after 2k warm-up steps. Rollout correction is an orthogonal data-generation intervention during updates 50k–60k; it is not a hard-retrieval phase.

The retrieval architecture contains exactly one global retrieval module, applied once per predicted position after six width-one local blocks. At inference, that module forms one one-hot vector at the argmax and takes one value-vector weighted sum. Because a width-one block attends only to its own position, each candidate value depends on one semantic context token rather than a contextual mixture. The mask includes the diagonal, so the argmax may select the current position; across any prefix–suffix cut this is therefore *at most* one selected prefix position, not necessarily exactly one. This is a useful architectural analogue of $b \le 1$ at inference, but not a literal cBAPO realization: training retrieval is dense and soft, the network is not tested for adversarial split consistency, and its composite query encoding differs from the formal edge-list encoding.

Training graphs have 8–32 active vertices and token-edge density $m/n \in \{1,2,4\}$. The balanced generator alternates conditioned-random and constructed positive and negative instances and includes repeated edges and self-loops. The held-out suites test the same distribution (IID), sizes 40/48/64 (Size), density $m/n = 8$ on 16–64 active vertices (Dense), 30% repeated edge tokens on 16–64 vertices (Robust.), reversed IID edge order (Reverse), and oracle traces with at least m non-halt tokens (Long). Semantic adjacency hashes separate train, validation, and nonpaired test suites. We report final-answer accuracy and strict equality to the deterministic canonical trace; path-certificate validity is computed separately for predicted positive traces. Appendix I gives the precise generators, metrics, and stopping rules.

IID, robustness, reverse-order, and long-trace suites contain 20k examples, Size 15k, and Dense 10k. Teacher-forced retrieval accuracy is 99.37% IID, 99.19% Size, 97.95% Dense, 99.39% Robust., 99.37% Reverse, and 99.06% Long. Answer-only is near chance on most suites and reaches 68.60% on Dense; the present design does not isolate whether density, length, label balance, or another correlated feature explains that difference. Vanilla threaded nearly solves the IID canonical transduction and

Table 2: Descriptive single-seed autoregressive accuracy (%) at the common step-60k budget endpoint. Each entry is answer/exact-trace accuracy. Answer-only exact outputs are not comparable to threaded traces and are shown as "–".

Model	IID	Size	Dense	Robust.	Reverse	Long
Retrieval soft	99.1/89.3	98.0/77.7	97.3/24.8	98.7/87.4	99.1/89.5	96.9/63.8
Vanilla threaded	100/100	98.9/93.3	81.8/39.2	99.6/96.7	100/100	98.5/90.7
Vanilla answer-only	55.5/–	58.4/–	68.6/–	55.0/–	55.8/–	54.8/–

has higher exact-trace accuracy on every split. Retrieval obtains 97.32% answer accuracy on Dense, compared with 81.76% for vanilla threaded, while its exact-trace rate is 24.79%. This gap is consistent with some noncanonical local decisions leaving the final answer or an implicit certificate intact, but aggregate metrics do not identify a unique error-accumulation mechanism. Dense-suite certificate validity is 95.61% (4,897 of 5,122 predicted-YES traces). These comparisons describe one seed and these fixed suites, not average training behavior or causal effects.

These experiments probe finite-sample learnability rather than establish the theorem: learned soft retrieval only approximates the exact top-one oracle, and none of the neural results imply asymptotic length generalization.

H CHECKPOINT AND CONVERGENCE AUDIT

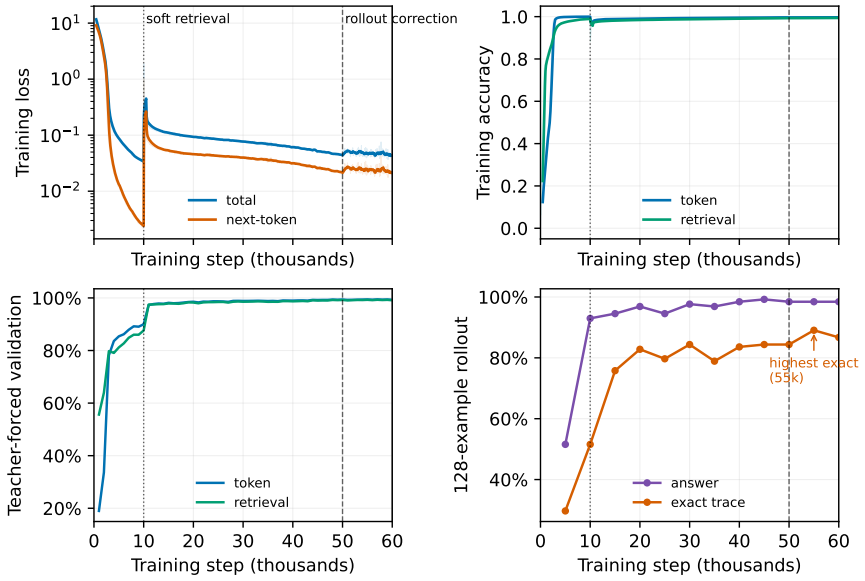


Figure 1: From-scratch soft retrieval. Thin training curves are raw 50-step logs and thick curves are ten-point means. Lines mark the end of oracle retrieval (10k) and start of rollout correction (50k). Teacher-forced validation is stable by 40k–60k; the 128-example rollout fluctuates at $1/128$ resolution and peaks in exact trace at 55k, not 60k.

The table uses the 60k endpoint for every model because training budget was the selection rule. The logged checkpoint saver instead defines `best.pt` by a strict improvement in exact-output accuracy on the fixed 128-example rollout. Under that rule the from-scratch retrieval run first peaks at 55k (exact trace 0.890625), while vanilla threaded and answer-only first peak at 25k. For answer-only, the selection criterion is exact matching of the two-token answer-and-halt target. Its value coincides with answer accuracy at the logged checkpoints, but this is not an identity of the metrics: a correct answer with a missing halt or extra token is not exact. No hidden trace-based criterion is applied. We

do not use any of these `best.pt` files in Table 2, avoiding both post-hoc test selection and unequal training budgets. The 60k point should therefore be read as a common fixed-budget endpoint near a teacher-forced plateau, not as three coincident validation optima.

I Neural Metrics, Data, and Training Details

I.1 Metrics and stopping

Autoregressive decoding is greedy. It stops at the first HALT, at the model's maximum sequence length, or after $\min(3m + 4, \text{remaining context capacity})$ generated tokens. Output after the first HALT is ignored. *Answer accuracy* extracts the first generated YES or NO and compares it with semantic BFS reachability; a trace containing neither answer is incorrect. This metric is reported independently of halting, so emitting a correct answer without a later HALT can count as answer-correct but not exact. *Exact-trace accuracy* requires the entire generated sequence through HALT to equal the deterministic decreasing-label DFS target. Any illegal token, early stop, missing HALT, extra pre-HALT token, or noncanonical but semantically valid trace fails this metric. Answer-only targets are exactly (YES, HALT) or (NO, HALT).

Retrieval accuracy is teacher-forced, not autoregressive: at each supervised position with a nonempty retrieval target set, the hard argmax retrieval position must belong to that set. The set contains every duplicate-edge position that is semantically valid, and accuracy is micro-averaged over such positions. Positions requiring no retrieval are excluded.

Certificate validity is evaluated only when the first predicted answer is YES. The implementation scans the generated sequence through its first HALT (not merely the prefix ending at the first YES), reads each predicted discovery token $D_{p,v}$ as a parent assignment, and takes the last relevant check $C_{u,t}$ as the final edge into t. If the same child v occurs in multiple discovery tokens, the later assignment overwrites the earlier one. The checker then requires that the final edge occurs in the semantic input graph, that following the resulting parent map reaches s without a repeated vertex, and that every consecutive path edge occurs in the graph. For $s = t$, the empty certificate is valid when the first generated token is YES. The reported Dense value 95.61% has denominator 5,122 predicted-YES traces and numerator 4,897 valid certificates; it is not divided by all 10,000 Dense examples and does not require an exact canonical trace.

I.2 Data construction and separation

Dataset index modulo four fixes equal counts of positive/negative and conditioned-random/constructed examples. Conditioned-random examples rejection-sample an adjacency with the requested label; if rejection fails, positive cases fall back to a planted path and negative cases to a planted cut. Constructed positives plant an s–t path before filling remaining edges, whereas constructed negatives fill edges subject to a cut separating s from t. The configured token count is $m = \text{round}(dn)$ for density d, with at least one token; a configured fraction of tokens are repeated copies, edge order is shuffled, and self-loops are sampled at the stated rate. The generator verifies the requested label by graph search.

Separation uses a hash of semantic adjacency, not edge order or multiplicity. The first hexadecimal hash digit assigns disjoint buckets: train 0–7, validation 8, IID 9, Size 10, Dense 11, Robustness 12, hard-negative 14, and Long 15. A process-local registry rejects duplicate semantic adjacencies within the generated datasets. Reverse is a paired perturbation of IID and intentionally reuses bucket 9 with the same semantic graphs in reversed token order; it is therefore excluded from the claim of nonpaired train/test adjacency separation.

I.3 ROLLOUT CORRECTION

All three reported runs—retrieval soft, vanilla threaded, and vanilla answer-only—use rollout correction during updates 50k–60k. On every fifth optimization update in that interval, each microbatch is replaced by four deterministically indexed training examples. The current model is rolled out greedily. If its first mismatch with the canonical target is at position j, the generated prefix through $j-1$ is retained and only the correct token at j (and its retrieval target when applicable) is supervised. By definition of the first mismatch, that retained prefix equals the canonical prefix; this intervention selects an error location for next-token supervision and does not train recovery from an already divergent prefix. If there is no mismatch within the target, the ordinary example is retained. The procedure neither changes inference-time decoding nor creates a hard-retrieval phase.