EXPRESSIVE POWER OF TRANSFORMERS FOR NESTED LOGICAL FILTERING IN TABLE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the expressive power of Transformers in table reasoning. We approach this question through Nested Logical Filtering (NLF), one of the fundamental operations of table reasoning, which evaluates a Boolean query on every row. Longer chains of thought (CoT) expand what Transformers can compute, but expressive power alone does not ensure reliable answers on growing tables. We identify the core challenges of solving NLF with CoT through theoretical and experimental analysis, spanning evaluating nested queries and reliably answering questions. The decision version of NLF is NC^1-complete even on a one-cell table, yielding a conditional barrier for constant-depth, log-precision Transformers with $O(\log N)$ CoT emissions, where N is the encoded input length. Polynomial-length CoT can nevertheless compute the full answer exactly. We then compare answer generation with program-of-thought (PoT) execution under a common symmetric token-substitution channel. With a fixed vocabulary and positive noise rate, every direct-readout CoT system has exact-answer success at most exponentially small in the row count, regardless of reasoning length. A system emitting a fixed, constant-length program retains a positive success lower bound when execution on clean input and its output are noiseless. Synthetic experiments illustrate the resulting exposure contrast. WikiTableQuestions and DataBench provide complementary evidence on table size and answer structure. The guarantees depend on the output interface, separating computational capability from reliable delivery.

1 INTRODUCTION

Many table questions require deciding which rows qualify before returning a value. For example, "which employees are over 40 and either work in sales or earn less than 50k?" combines comparisons with nested Boolean logic on every row. This operation, *Nested Logical Filtering* (NLF), returns the vector of row decisions (Section 2). This Boolean query test on every row is the basis of many table operations, and a wrong row decision by an LLM can change a downstream answer. Characterizing the expressive power Transformers need to handle NLF therefore addresses their expressive power in table reasoning. To the best of our knowledge, this work is the first to give a formal, complexity-theoretic analysis of this expressive power. Query nesting and table size create distinct challenges: the former can make even one row hard, while the latter lengthens the exact answer. Systems tackle table questions through textual reasoning over serialized tables (Jin & Lu, 2023; Wang et al., 2024; Ye et al., 2023) or generated programs (Gao et al., 2023; Chen et al., 2023; Cheng et al., 2023). Table benchmarks compare these approaches (Li et al., 2025; Wu et al., 2024; 2025), but do not isolate their computational requirements from the reliability of their outputs.

Prior theory shows why reasoning length matters. One constant-depth, log-precision Transformer pass is simulable by constant-depth threshold circuits (Merrill & Sabharwal, 2023b;a). Longer chain-of-thought (CoT) can use generated tokens for sequential computation (Merrill & Sabharwal, 2024; Li et al., 2024). For NLF, this raises two questions. How much generation is needed to compute every row decision? Once those decisions are computed, can the system deliver the full vector reliably?

We first show that deciding whether a row satisfies an NLF query is NC^1-complete even on a one-cell table (Section 3). The worst-case difficulty comes from increasingly nested queries, not more rows. Under the circuit-separation assumption stated below, logarithmically many CoT tokens cannot solve every instance exactly in our Transformer model. Randomized generation cannot achieve a two-thirds

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.

success rate on every input either. A generated program can pass the row loop and nested predicate evaluation to an interpreter. Polynomial-length CoT can also compute the complete row-by-row answer exactly. This shows that enough CoT can overcome the computational barrier, but the full answer still has to carry one decision per row.

One wrong decision makes the full NLF answer incorrect. As the table grows, writing all decisions in text requires a longer output and gives generation errors more chances to change it, even when the model computed the decisions correctly. This is why we compare direct answer text with program execution under the same independent token-error process (Section 4). The chance of an entirely correct direct-readout CoT answer falls exponentially with row count, regardless of reasoning length. A program-of-thought (PoT) system can instead emit a fixed program that loops over the rows and lets an interpreter return the answer without the model transcribing it. We show that this construction retains a fixed positive chance of success as tables grow, provided execution uses clean input and returns its output directly. This result isolates the effect of the output interface under the stated error model. The experiments below examine learned systems separately.

We then test how accuracy changes as answers grow and generated text is corrupted (Section 5). On synthetic depth-two NLF queries, textual CoT exact match falls from 0.300 at 50 rows to zero at 200 rows and above without injected noise, while PoT is correct on every tested instance. Injected character substitutions steepen the textual CoT decline. PoT accuracy falls with noise but changes little with row count. On WikiTableQuestions (Pasupat & Liang, 2015), PoT and textual CoT obtain judge accuracies of 0.875 and 0.625 in the $150+$-row bin. On the 101-row DataBench table (Osés Grijalba et al., 2025), the corresponding scores are 0.975 and 0.638. The synthetic corruption is applied after generation rather than fed back as in the theorem. Public questions often have short answers, and PoT receives execution-feedback repair. These comparisons support the proposed interface explanation within their respective protocols.

Relation to prior work. Prior studies characterize CoT expressiveness (Strobl et al., 2024; Merrill & Sabharwal, 2024; Li et al., 2024), including for Boolean formula evaluation (Yehudai et al., 2025), and derive reasoning-length lower bounds for other tasks (Amiri et al., 2025; Abbe et al., 2024; Peng et al., 2024). We connect this theory to table reasoning through NLF completeness and an exact full-answer construction. Our one-row reduction embeds formula evaluation, but our results concern the full table-level answer and its delivery. Program-aided reasoning offers practical alternatives to textual computation (Chen et al., 2023; Gao et al., 2023; Cheng et al., 2023). Our reliability result adds a formal comparison of how answer text and executed programs expose generated tokens to errors. Full proofs and supplementary results appear after the references.

2 Problem and Setting

Definition 2.1 (Nested Logical Filtering)**.** A table T has $n \geq 1$ rows and $m \geq 1$ columns. For complexity statements, its cells and query constants are b-bit strings: equality compares codes and order compares unsigned values. A query Q is a fan-in-two Boolean formula over $\wedge, \vee$, with negations pushed to comparison leaves $[T[i][j] \bowtie v]$, where $\bowtie \in \{=, \neq, <, \leq, >, \geq\}$ (categorical values use equality or inequality). Define

$$\mathrm{NLF}(T, Q) = (Q(T[1]), \ldots, Q(T[n])), \qquad \mathrm{NLF}_{\mathrm{dec}} = \{(T, Q, i) : Q(T[i]) = 1\}.$$

The decision language rejects invalid encodings, and the full function returns $\perp$ on them. The total encoded length N includes the table, query, and decision row index when present. Appendix B.2 specifies the encoding.

The output has n bits, but the input size is N: although $n \leq N$, we assume no bound $N \leq \mathrm{poly}(n)$. Indeed, the hardness reduction uses one row and a growing query.

Computational models. A Boolean circuit represents a computation as connected elementary operations, or gates. Its *size* counts the gates, while its *depth* counts the longest chain of dependent operations. We consider circuits whose size grows at most polynomially with input length and use the classes $\mathsf{AC}^0 \subseteq \mathsf{TC}^0 \subseteq \mathsf{NC}^1$. The class AC^0 uses AND, OR, and NOT gates with constant depth. AND and OR gates may combine arbitrarily many inputs. The class TC^0 additionally permits threshold gates, such as majority, which test whether enough inputs are 1. In NC^1, AND and OR

row	dept	age	salary
1	sales	46	62k
2	eng	37	45k
3	sales	29	48k
4	hr	52	44k

$$Q = \text{age} > 40 \wedge (\text{dept} = \text{sales} \vee \text{salary} < 50\text{k})$$
$$\text{NLF}(T, Q) = (1, 0, 0, 1)$$

Figure 1: NLF evaluates the same nested predicate on every row.

gates have two inputs each, and the number of layers may grow logarithmically with input length. These classes describe the computations available under different limits on sequential depth. Their formal definitions appear in Appendix A.

Unless stated otherwise, circuit families are *uniform*: a single efficient rule specifies their structure across input lengths. We write $\mathsf{TC}^0/\mathrm{poly}$ for the non-uniform version, which permits a separate circuit for each input length without a common construction rule. These circuits retain the same size and depth bounds, and each must handle every input of its length. This accommodates length-dependent Transformer parameters and prescribed seeds. Our short-CoT lower bounds use the following separation assumption.

Assumption 2.2.

$$\mathsf{NC}^1 \not\subseteq \mathsf{TC}^0/\mathrm{poly}.$$

Boolean formula evaluation is NC^1-complete, making it a natural model of the nested logical evaluation required by NLF (Buss, 1987). Assumption 2.2 expresses the conjecture that this evaluation cannot always be compressed into a fixed number of threshold-gate layers using only polynomially many gates, even with circuits tailored to each input length. This is a longstanding open separation in circuit complexity (Allender & Koucký, 2010), also used to obtain Transformer lower bounds for arithmetic reasoning (Feng et al., 2023).

Transformer model. We study decoder-only causal Transformers, which predict the next token from the input and the tokens generated so far (Merrill & Sabharwal, 2023b;a; Li et al., 2024). Their precision p_N, layer count L_N, width d_N, and heads per layer H_N satisfy

$$p_N = O(\log N), \qquad L_N = O(1), \qquad d_N, H_N = N^{O(1)}.$$

Thus the number of layers stays fixed as inputs grow, and each numerical value uses at most logarithmically many bits. Width and head count may grow polynomially, allowing more parallel computation without adding layers. Parameters may depend on N but remain fixed throughout a run. The arithmetic, component operations, input encoding, and token-selection rules satisfy the circuit bounds in Appendix A. Together, these assumptions place one forward pass in $\mathsf{TC}^0/\mathrm{poly}$.

Chain of thought. Over a fixed finite vocabulary Σ, each generation step appends one token and makes it available to the next forward pass. Generated tokens can therefore record intermediate results for later computation, even though the Transformer has the same number of layers at every step. Figure 2 illustrates this loop and a transcript-only readout. We write $\mathsf{CoT}[T(N)]$ for the decision problems solvable using $T(N)$ emitted tokens, with the final token specifying acceptance or rejection. The budget includes the answer token, so $T(N) = 1$ is a single pass. This class uses fixed-seed decoding: one seed is prescribed for all inputs of a given length. Random-seed success probabilities are stated separately, with the seed sampled independently of the input.

For the full n-bit NLF answer, an extractor g_N reads the generated transcript W and returns a binary string or $\perp$. It has polynomial-size TC^0 circuits and cannot access the original input, so the transcript must contain the information needed to recover the answer. Appendix A.2 specifies the readout formally. Section 4 introduces the separate noisy interfaces, whose guarantees apply to unrestricted generators.

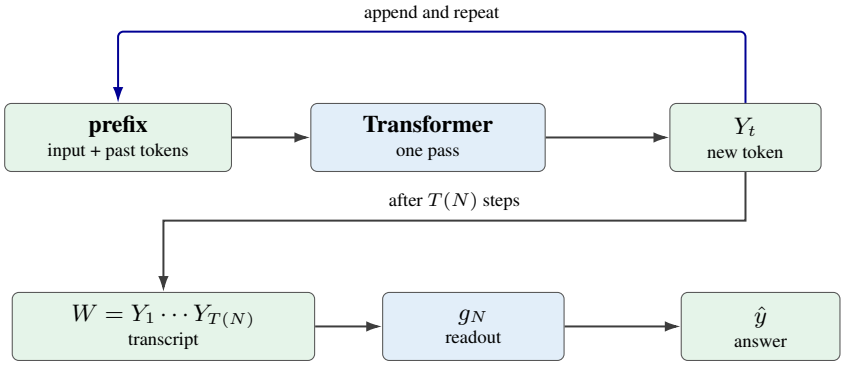


Figure 2: CoT generation and full-answer readout. Each new token extends the prefix before the next Transformer pass. After $T(N)$ steps, g_N reads the transcript to produce an answer.

3 Computational Landscape

The complexity of NLF determines which generation budgets can support exact table reasoning. We establish a conditional barrier for single-pass and logarithmic-CoT Transformers, together with a constructive result for polynomial CoT. Table 1 summarizes the guarantees and the remaining gap between them.

Emissions	Computational guarantee	Consequence for NLF
Single pass	Fixed-seed maps lie in $\mathsf{TC}^0/\mathrm{poly}$	No exact or per-input bounded-error solver
$O(\log N)$	The whole transcript has a $\mathsf{TC}^0/\mathrm{poly}$ simulation	Same barrier, including full answers and logarithmic-depth queries
$\Theta(N)$	An NC^1-complete language is decidable in the cited models	Full NLF with a linear budget remains unresolved here
$\mathrm{poly}(N)$	P/poly or P in the respective cited models	**Exact full NLF is attainable** (Proposition 3.3)

Table 1: The computational landscape, measured in total encoded input length N. The single-pass and logarithmic-CoT barriers are conditional on Assumption 2.2. Bounded error means success at least $2/3$ on every input. The linear and polynomial decision results retain the distinct model assumptions of Theorem A.10.

3.1 Complexity of Nested Logical Filtering

Theorem 3.1. $\mathrm{NLF}_{\mathrm{dec}}$ *is* NC^1*-complete under* $\mathsf{DLOGTIME}$*-uniform* AC^0 *many-one reductions.*

The result connects NLF to Boolean formula evaluation: comparisons on a row can be evaluated in parallel, after which their truth values are combined according to the query. Conversely, even a query over one Boolean table cell can represent a hard formula-evaluation instance. Hardness already holds for equality queries of logarithmic depth. All bits of the full answer are also computable in uniform NC^1, and the one-row instances transfer decision hardness to full NLF. Appendix B contains the proof and reduction diagram.

The complexity parameter is the joint encoded length N: a growing query can carry a hard instance even when the table has only one row. Increasing the number of rows for a fixed shallow predicate does not create this worst-case difficulty. Our depth-two synthetic queries therefore examine full-answer reliability as n grows, while the completeness result concerns the joint family of tables and queries.

3.2 SINGLE-PASS AND LOGARITHMIC-COT BARRIERS

Under the model assumptions of Section 2, a single forward pass is simulable by polynomial-size, constant-depth threshold circuits. Logarithmically many emissions retain this upper bound (Li et al., 2024): with a fixed vocabulary, there are only polynomially many possible transcripts of that length, and their consistency with generation can be checked in parallel. Consequently,

$$\mathsf{CoT}[O(\log N)] \subseteq \mathsf{TC}^0/\mathrm{poly} \qquad \text{(Theorem A.7)}.$$

The simulation also covers the full-answer extractor, with circuit bounds independent of the chosen seed. Combining it with NLF completeness gives the following barrier under Assumption 2.2.

Theorem 3.2 (Short-CoT barrier)**.** *Under Assumption 2.2,* $\mathrm{NLF}_{\mathrm{dec}} \notin \mathsf{CoT}[O(\log N)]$. *No random-seed decision system in the same model with* $O(\log N)$ *emissions succeeds with probability at least* $2/3$ *on every input.*

The non-uniform formulation accommodates length-dependent model parameters and prescribed seeds. The random-seed result follows by amplification and seed fixing. The full argument is in Appendix C. The same barrier holds for full answers and logarithmic-depth queries, with a single pass as the special case $T(N) = 1$. These worst-case statements give neither a benchmark error rate nor a row-count threshold. An eventual average-case ceiling requires the separate distributional hardness hypothesis of Proposition C.3.

3.3 POLYNOMIAL-LENGTH COT

Longer CoT allows intermediate results to be reused over more sequential steps, and the logarithmic-CoT simulation no longer supplies a constant-depth upper bound. In the cited deterministic models, linear CoT can decide an NC^1-complete language, while polynomial CoT reaches P/poly or P (Theorem A.10). These are model-specific characterizations: the former uses Li et al.'s non-uniform family convention, and the latter uses Merrill and Sabharwal's uniform model with its attention and normalization assumptions. Appendix A.6 gives those conventions. For full NLF, we obtain the following constructive result.

Proposition 3.3 (Exact full-answer computation)**.** *With the fixed-point arithmetic of Li et al. (2024, Section 3.1), a constant-depth, constant-precision Transformer family can emit a polynomial-length transcript from which a string readout as in Appendix A.2 returns* NLF *exactly. Greedy decoding achieves success* 1 *under either seed convention.*

The construction uses generated tokens to record intermediate circuit values and extracts all answer bits from the completed transcript. Its proof and model compatibility are given in Appendix A.6.

The results establish a gap between logarithmic-CoT impossibility and polynomial-CoT sufficiency, without identifying the minimum budget for full NLF. In particular, the cited linear-CoT result for another complete language does not establish a linear-budget full-answer solver. Expressive sufficiency also supplies no training or length-generalization guarantee. It does establish that failure at arbitrary polynomial lengths cannot be attributed solely to the short-CoT circuit barrier. Reliable delivery of the computed answer requires a separate analysis.

4 RELIABILITY UNDER NOISE

Although polynomial CoT can compute NLF exactly, our synthetic experiments show declining CoT accuracy as tables grow, while PoT remains more stable (Section 5.1). This contrast motivates examining how each approach delivers its answer. Direct-readout CoT must emit the full n-bit answer, exposing more tokens to possible errors as the table grows. PoT can generate a fixed program that loops over the rows, so producing a longer answer need not require more generated tokens.

To compare their exposure to generation errors, we apply the same token-noise channel to both approaches. The generated program can also be corrupted, but the input, execution, and executed output remain noiseless. The definitions below formalize this comparison for unrestricted generators, isolating the role of the output interfaces.

4.1 CHANNEL AND OUTPUT INTERFACES

Definition 4.1 (Noisy generation)**.** Fix a finite vocabulary Σ, $|\Sigma| \geq 2$, and $\varepsilon \in (0, 1/2]$. At step t, a generator chooses $W_t = f_t(x, \rho, R_{<t})$ on input x, with an internal seed ρ prescribed for its input length or sampled independently of x. The channel $\mathcal{N}_\varepsilon$ returns $R_t = W_t$ with probability $1 - \varepsilon$. Otherwise it substitutes a uniformly chosen symbol of $\Sigma \setminus \{W_t\}$. Each step uses fresh channel randomness, independent of the input, internal seed, and earlier channel draws. Realized tokens R_t are fed back. Input, readout, execution, and execution output are noiseless. The maps f_t and generation length are unrestricted.

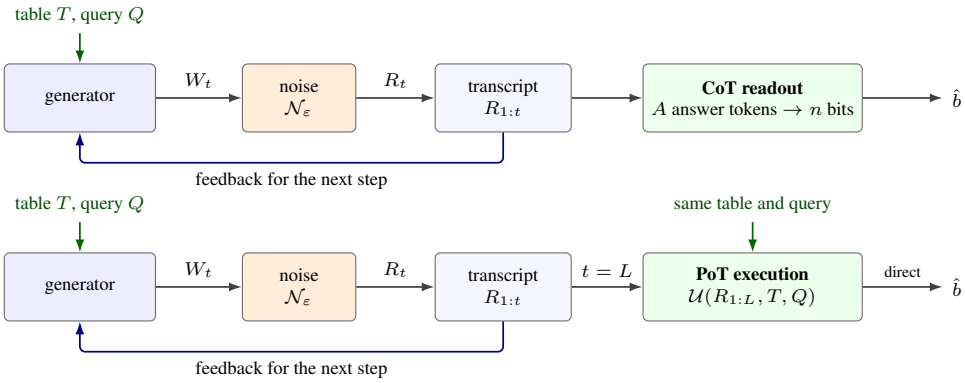


Figure 3: Both interfaces corrupt generated tokens and feed them back. PoT executes the realized program on clean input and returns its output directly.

Definition 4.2 (Direct-readout CoT)**.** Let n denote the answer length in bits and A the number of answer tokens. For each n, fix positive block sizes $\beta_1, \ldots, \beta_A$ summing to n, and bijections $\varphi_i : S_i \to \{0,1\}^{\beta_i}$ with $S_i \subseteq \Sigma$. A direct-readout system commits at ordered times $\tau_i \in \mathbb{N}_{\geq 1} \cup \{\infty\}$, strictly increasing whenever the later time is finite. Each time is *predictable*: whether $\tau_i = t$ is determined by $(x, \rho, R_{<t})$ before R_t is observed. Its answer is

$$\hat{b} = \varphi_1(R_{\tau_1}) \cdots \varphi_A(R_{\tau_A}).$$

An unreached slot or token outside S_i returns $\perp$.

Each answer token encodes at most $\log_2 |\Sigma|$ bits, so $A \geq n / \log_2 |\Sigma|$. Appendix E.1 gives the formal specification.

Direct readout needs tokens carrying every row decision. PoT can instead generate a program that loops over the table. We pass that code through the same noisy channel, then run it with a fixed interpreter on the clean table and query. The interpreter returns the answer directly, without the model transcribing it. We choose program length before generation and bound execution time.

Definition 4.3 (Program-of-thought)**.** Fix a deterministic universal interpreter $\mathcal{U}$ with polynomial simulation overhead and a polynomial execution budget p. A PoT system chooses $L = L(x)$ before generation, emits L noisy tokens, and executes $\mathcal{U}(R_{1:L}, T, Q)$ on clean input for at most $p(N + L)$ steps. It returns the resulting n-bit string directly, or $\perp$ on invalid code, timeout, or invalid output. The interpreter and budget are independent of the input.

4.2 SEPARATION UNDER TOKEN NOISE

Theorem 4.4 compares these interfaces for unrestricted generators on every NLF instance, without a circuit-separation assumption.

Theorem 4.4 (Noise-robustness separation). *Fix $\varepsilon \in (0, 1/2]$ and finite $|\Sigma| \geq 2$. Under $\mathcal{N}_\varepsilon$, with either internal-seed convention:*

(a) Every direct-readout CoT system, on every NLF *instance with n rows, has exact-answer success at most*

$$(1-\varepsilon)^A \leq (1-\varepsilon)^{n/\log_2 |\Sigma|}.$$

(b) There exists a PoT system intending a fixed correct NLF *program P_0 of constant length $L_0 = |P_0|$, whose success on every instance is at least*

$$(1-\varepsilon)^{L_0} > 0, \qquad \textit{independently of } n, m, \textit{ and } |Q|.$$

The PoT guarantee is an existence result for a fixed correct evaluator that receives the clean table and query as input, so its program length need not grow with either. For each fixed positive noise rate, its success lower bound eventually exceeds the upper bound for every direct-readout CoT system as the number of rows grows. This does not guarantee correctness of arbitrary learned programs. Execution output must be returned directly, since making the generator transcribe it exposes the answer tokens to noise again.

Appendix E proves the output-interface separation. A direct-readout system commits to each answer token before noise is applied. Fresh channel randomness limits the chance that the token decodes correctly, even when generation adapts to earlier corruptions. Since every answer token must be correct, full-answer success falls exponentially with row count. The PoT proof sends a fixed evaluator through the same channel. If its short program survives, clean execution returns the full answer without the model writing its bits. The appendix also extends the bound to limited token redundancy and analyzes programs that embed the query, whose guarantee depends on query length. External majority voting over repeated bits and choosing answer positions after generation lie outside direct readout (Appendix E.4).

Appendix D studies transcript length and model errors without injected noise. Its transcript bound shows that even an easy NLF query needs output proportional to the row count when the answer varies freely and the decoder cannot see the input. This cost persists even with perfect computation. For model errors, an average row-error rate that stays bounded away from zero after earlier correct decisions directly yields exponential decay. Exact polynomial-length CoT solvers show why this error condition cannot be inferred from the Transformer model. Positive-temperature sampling alone does not imply the injected-noise channel.

5 Experiments

We study two questions: how exact-answer accuracy changes with table size and injected noise on synthetic NLF, and whether related differences appear on public table benchmarks. All experiments use a DeepSeek backbone. *Textual CoT* (TCoT) reads the serialized table and writes its answer in text. *PoT* emits a program that executes on the clean table, with its output passed directly to the scorer. *Strict* scoring compares the answer with the gold string after whitespace normalization. An additional LLM *judge* accepts representational variants such as `1.0` for `1` and reordered list items. We report disagreements between these metrics explicitly.

The theoretical comparison motivates two measurements: sensitivity to the number of exposed answer positions, and sensitivity to the length and correctness of generated programs. The theorem supplies an upper bound for direct readout and an existence lower bound for PoT, so it does not require measured accuracies to equal either bound. In particular, an empirical PoT curve can decrease with table size if the model writes longer or less accurate programs. We therefore examine the scaling pattern together with observed program execution and correctness, and separate full-vector exact match from the public benchmarks' more varied answer formats.

5.1 Synthetic filtering under injected noise

Design. We generate 20 instances at each $n \in \{50, 100, 200, 400, 800\}$, using a depth-2 filter with selectivity in $[0.2, 0.8]$ and an exact n-bit indicator answer. Both modes return one line beginning with `ANSWER:`, followed by the indicator string. The first occurrence is scored. TCoT emits this string,

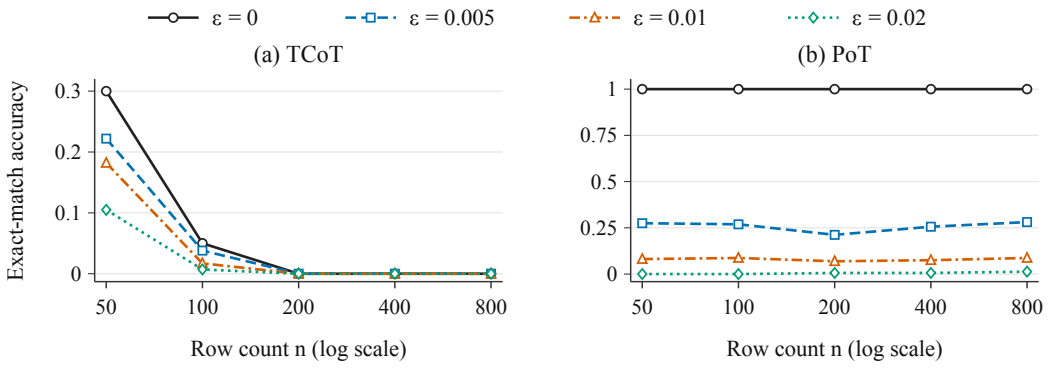


Figure 4: Synthetic NLF exact match. After generation, each character produced by the model is independently replaced with probability ε. Thus $\varepsilon = 0$ means no injected noise. Interpreter output is unaffected. Panels use separate linear accuracy scales, and row counts are logarithmically spaced.

whereas PoT emits a pandas program whose printed output is scored. Synthetic PoT is single-shot: an execution failure scores zero.

Transcripts are generated once at temperature 0. Independent symmetric substitutions are then applied per emitted character at rate ε, with Monte-Carlo resampling. Interpreter output is exempt. This measures corruption of completed answers or programs. Unlike Definition 4.1, corrupted characters are not fed back during generation. The experiment therefore does not test the full feedback-coupled channel, and its results need not bound the effect of that channel in either direction.

Results. Without injected noise, TCoT exact match falls from 0.300 at 50 rows to 0.050 at 100 rows and zero at larger sizes. PoT is correct on all tested instances (Figure 4). With noise, PoT accuracy decreases with ε but exhibits little row-count dependence: its fitted absolute log-accuracy slopes are at most 0.0012. At $\varepsilon = 0.005$, accuracy ranges from 0.212 to 0.281, consistent in scale with $(1-\varepsilon)^{270} \approx 0.26$ for a typical program of about 270 characters. These measurements illustrate the different exposed output lengths. They do not establish that a learned system emits a correct program on other inputs.

For TCoT, an ordinary-least-squares fit of $\log \mathrm{EM}$ against n, restricted to positive exact-match rates, gives slope -0.0358 without injected noise. The reported first-error estimate is $37/1053 = 0.0351$ errors per row-slot reached with a correct prefix. Their numerical agreement is consistent with the small-hazard approximation $-\log(1-h) \approx h$, rather than an exact equality between hazard and log loss. At $\varepsilon = 0.01$ and 0.02, fitted slopes are -0.0478 and -0.0551, respectively, compared with -0.046 and -0.056 from adding $\log(1-\varepsilon)$ to the noiseless slope. Only two row counts have positive TCoT accuracy, so these fits are descriptive diagnostics of a small sample, not evidence of a universal exponential law. Additional diagnostics appear in Appendix F.

Repeated channel draws reuse the same query-table instances. They do not add independent tasks. Observed zeros do not establish zero population accuracy. Appendix F provides the full grid and diagnostics.

5.2 Public benchmarks

Public-benchmark questions include filtering followed by projection or aggregation, alongside other operations, and often have short answers. They do not require recovery of the full NLF indicator vector, so their results provide complementary empirical comparisons rather than direct tests of Theorem 4.4. No noise is injected.

On these benchmarks, PoT receives execution feedback for up to 8 rounds: after an execution error, the error message and table schema are appended to the context and a replacement program is generated. The process stops at the first completed execution. Failure after the final round scores zero. The answer is never re-verbalized. This procedure differs from single-shot synthetic PoT and

(a) WikiTableQuestions: judge accuracy by table size

table size	TCoT	PoT	PoT execution
< 25 rows	0.750	0.825	0.975
25–60	0.788	0.838	1.000
60–150	0.700	0.825	0.975
150+	0.625	0.875	1.000

(b) DataBench: accuracy by answer type and table size

	TCoT		PoT		PoT
stratum	judge	strict	judge	strict	execution
`boolean`	0.975	0.900	0.975	0.975	0.95–1.0
`category`	0.825	0.750	0.950	0.900	1.000
`number`	0.625	0.600	0.875	0.850	1.000
`list[category]`	0.750	0.250	0.900	0.650	1.000
`list[number]`	0.813	0.500	0.925	0.450	1.000
full table (101 rows)	0.638	0.475	0.975	0.900	1.000

Table 2: Public benchmarks: 40 matched instances per bin or stratum. PoT uses execution-feedback repair.

gives PoT an additional repair budget. Execution rates below denote successful execution, not answer correctness.

WikiTableQuestions. We sample 40 identical instances per mode from each of four row-count bins, with a 50K-character context cap so both modes can access the complete table (Pasupat & Liang, 2015). Table 2(a) shows the largest judge-score difference in the 150+-row bin: 0.875 versus 0.625. Across the smallest and largest bins, TCoT strict match falls from 0.775 to 0.550. PoT leads under strict scoring as well. TCoT's per-instance token cost grows approximately $15\times$ across bins, while PoT's is approximately constant. These comparisons suggest a table-size effect in this sample, but do not isolate it from differences in question composition or establish the full-vector theorem for short answers.

DataBench. We use the official lite track, with 20-row tables and 40 matched instances for each of five typed-answer strata (Osés Grijalba et al., 2025). A sixth stratum contains 40 questions from the only full dataset fitting the context cap (101 rows). The other 15 candidate full datasets are excluded. Judge scores tie on Boolean answers and favor PoT in the remaining strata (Table 2(b)). The largest difference is on the full table: 0.975 versus 0.638 by judge and 0.900 versus 0.475 by strict match. TCoT uses $3.6\times$ as many tokens there. For `list[number]`, the scoring direction reverses: PoT has higher judge accuracy (0.925 versus 0.813), but lower strict accuracy (0.450 versus 0.500), with float representations such as `1.0` versus `1` contributing to mismatches. Answer type, task difficulty, and output formatting all vary across strata, so these results do not identify answer length alone as the cause of the differences.

6 Conclusion

Nested Logical Filtering has a two-sided computational landscape: its decision version is NC^1-complete even on a one-cell table, logarithmic CoT is insufficient under $\mathsf{NC}^1 \not\subseteq \mathsf{TC}^0/\mathrm{poly}$, and polynomial CoT can compute the full answer exactly. The minimum sufficient budget for full NLF remains unresolved between these bounds. Under symmetric token substitution, reliable delivery introduces a separate constraint: every direct-readout generator has exact-answer success exponentially small in the row count, while a fixed correct program retains a positive success lower bound through clean execution. The experiments illustrate this exposure contrast within their stated protocols. Together, the results distinguish the computation needed to evaluate a query from the interface needed to deliver its answer. Appendix G discusses limitations and open questions.

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 (NeurIPS)*, 2024.

Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *Journal of the ACM*, 57(3), 2010. doi: 10.1145/1706591.1706594.

Alireza Amiri, Xinting Huang, Mark Rofin, and Michael Hahn. Lower bounds for chain-of-thought reasoning in hard-attention transformers. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. arXiv:2502.02393.

Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

Samuel R. Buss. The Boolean formula value problem is in ALOGTIME. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 123–131, 1987.

Samuel R. Buss, Stephen A. Cook, Arvind Gupta, and Vijaya Ramachandran. An optimal parallel algorithm for formula evaluation. *SIAM Journal on Computing*, 21(4):755–780, 1992. doi: 10.1137/0221046.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2211.12588.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Binding language models in symbolic languages. In *International Conference on Learning Representations (ICLR)*, 2023.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006.

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 (NeurIPS)*, volume 36, 2023.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *International Conference on Machine Learning (ICML)*, 2023.

András Hajnal, Wolfgang Maass, Pavel Pudlák, Mário Szegedy, and György Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46(2):129–154, 1993. doi: 10.1016/0022-0000(93)90001-D.

William Hesse, Eric Allender, and David A. Mix Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *Journal of Computer and System Sciences*, 65(4): 695–716, 2002. doi: 10.1016/S0022-0000(02)00025-9.

Ziqi Jin and Wei Lu. Tab-CoT: Zero-shot tabular chain of thought. In *Findings of the Association for Computational Linguistics (ACL)*, 2023.

Ce Li, Xiaoxuan Liu, Zhao Song, et al. TReB: A comprehensive benchmark for evaluating table reasoning capabilities of large language models. *arXiv preprint arXiv:2506.18421*, 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 (ICLR)*, 2024. URL https://arxiv.org/abs/2402.12875. arXiv:2402.12875.

William Merrill and Ashish Sabharwal. A logic for expressing log-precision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a.

William Merrill and Ashish Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11:531–545, 2023b.

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

Jorge Osés Grijalba, Luis Alfonso Ureña-López, Eugenio Martínez Cámara, and Jose Camacho-Collados. SemEval-2025 task 8: Question answering over tabular data. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pp. 2512–2522, Vienna, Austria, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.semeval-1.324/.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1470–1480, Beijing, China, 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1142. URL https://aclanthology.org/P15-1142/.

Binghui Peng, Srini Narayanan, and Christos Papadimitriou. On limitations of the transformer architecture. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2402.08164.

Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996.

Philip M. Spira. On time-hardware complexity tradeoffs for Boolean functions. In *Proceedings of the 4th Hawaii International Symposium on System Sciences*, pp. 525–527, 1971.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? A survey. *Transactions of the Association for Computational Linguistics*, 12:543–561, 2024.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *International Conference on Learning Representations (ICLR)*, 2024.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xinrun Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, Guanglin Niu, Tongliang Li, and Zhoujun Li. TableBench: A comprehensive and complex benchmark for table question answering. *arXiv preprint arXiv:2408.09174*, 2024.

Zhenhe Wu, Jian Yang, Zhongjiang He, Changzai Pan, Jie Zhang, Jiaheng Liu, Xianjie Wu, Yu Zhao, Shuangyong Song, Yongxiang Li, Zhoujun Li, and Xueling Li. Table-R1: Region-based reinforcement learning for table understanding. *arXiv preprint arXiv:2505.12415*, 2025.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of SIGIR*, 2023.

Gilad Yehudai, Noah Amsel, and Joan Bruna. Compositional reasoning with transformers, RNNs, and chain of thought. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2503.01544.

Appendix

Appendix A gives the Transformer semantics and simulation proofs, including the uniform refinements. Appendices B and C prove NLF completeness and the short-CoT lower bounds. Appendix D covers transcript information and endogenous errors. Appendix E proves the noise separation and its readout extensions. Appendix F supplies additional experimental details, and Appendix G discusses limitations and open problems.

Table 3 collects notation used throughout the paper.

N	total encoded instance length	Σ	vocabulary, $\lvert\Sigma\rvert = O(1)$
n	answer bits (rows in NLF)	ε	channel substitution rate
m	number of table columns	L_0	token length of the fixed program
b	field width in bits	A	number of answer tokens
$T(N)$	CoT length in tokens	β_i, τ_i	block size and position of answer token i
p_N, d_N	precision, width	q_r	per-slot base under r-redundancy
Q	nested logical query	λ	conditional per-row hazard

Table 3: Notation. N is the size parameter for every complexity-theoretic statement. n is the size parameter for every noise-robustness and experimental statement. We have $n \le N$ but assume no bound $N \le \mathrm{poly}(n)$.

A Transformer Semantics and Circuit Simulations

This appendix fixes the model and encoding conventions, proves the non-uniform single-pass simulation lemma and logarithmic-CoT simulation theorem, and then establishes their uniform refinements. The longer-CoT compatibility discussion appears last.

A.1 Circuit and Transformer definitions

Definition A.1. AC^0 consists of languages decided by polynomial-size, constant-depth circuits with NOT gates and unbounded fan-in AND and OR gates. TC^0 additionally permits unbounded fan-in threshold (majority) gates. NC^1 uses polynomial-size circuits of depth $O(\log N)$ with NOT gates and fan-in-2 AND and OR gates. Unless marked *non-uniform*, families are $\mathsf{DLOGTIME}$-uniform under the random-access convention of Merrill & Sabharwal (2023a) (Appendix A). For NC^1, uniformity is imposed on the extended connection language. For AC^0 and TC^0, direct connection uniformity suffices. We write $\mathsf{TC}^0/\mathrm{poly}$ for non-uniform TC^0.

Definition A.2. $A \le_m^{\mathsf{AC}^0} B$ if some $\mathsf{DLOGTIME}$-uniform AC^0-computable function f satisfies $x \in A \Leftrightarrow f(x) \in B$. L is NC^1*-complete* if $L \in \mathsf{NC}^1$ and $L' \le_m^{\mathsf{AC}^0} L$ for every $L' \in \mathsf{NC}^1$. Both TC^0 and $\mathsf{TC}^0/\mathrm{poly}$ are closed under $\le_m^{\mathsf{AC}^0}$.

Definition A.3 (Log-precision Transformer family)**.** Fix a finite ordered vocabulary Σ, $\lvert\Sigma\rvert \ge 2$, and a numerical convention from Appendix A.2. A *log-precision, constant-depth causal Transformer family* $\mathcal{T} = \{\mathcal{T}_N\}_{N \ge 1}$ consists of decoder-only Transformers acting on every nonempty finite token string. Each $\mathcal{T}_N$ has precision p_N, hidden dimension d_N, L_N layers, and at most H_N heads per layer, satisfying

$$p_N = O(\log N), \qquad L_N = O(1), \qquad d_N, H_N = N^{O(1)}.$$

For each N, one finite description of length $N^{O(1)}$ determines the parameters and positional rule at every prefix length. At precision p_N and prefix length m, every output bit of its numerical operations and its embedding, attention, layer, and readout maps has a constant-depth threshold circuit of size $(N+m)^{O(1)}$. A forward pass uses $(N+m)^{O(1)}$ scalar primitive operations. For $z \in \Sigma^+$, write $\ell_{\mathcal{T}_N}(z)$ for its next-token score vector.

A.2 CoT decision class and model conventions

Definition A.4 (CoT decision class)**.** Let $T(N) \ge 1$ be an integer-valued, polynomially bounded emission budget. Fix a finite ordered vocabulary Σ with distinguished symbols $\mathbf{0}, \mathbf{1}$, a Transformer

family as in Definition A.3, an input encoder enc_N, and token-selection rules $S_{N,t}$ satisfying the bounds below. On input $x \in \{0,1\}^N$ and seed $r_N = (r_{N,1}, \ldots, r_{N,T(N)})$, set $z_0 = \mathrm{enc}_N(x)$ and generate

$$F_{N,t}^{r_N}(z) = S_{N,t}\big(\ell_{\mathcal{T}_N}(z); r_{N,t}\big),$$
$$Y_t = F_{N,t}^{r_N}(z_{t-1}), \qquad z_t = z_{t-1}Y_t, \qquad 1 \le t \le T(N).$$

Write $W(x; r_N) = Y_1 \cdots Y_{T(N)}$. The decision bit $D_N(x; r_N)$ is 1 if $Y_{T(N)} = \mathbf{1}$ and 0 otherwise.

A language $L \subseteq \{0,1\}^*$ belongs to $\mathsf{CoT}[T(N)]$ if some such system and one prescribed seed r_N per input length satisfy

$$D_N(x; r_N) = \mathbf{1}\{x \in L\} \qquad \text{for every } N \ge 1 \text{ and } x \in \{0,1\}^N.$$

The empty input has a separately fixed decision. Unless explicitly qualified as uniform, the Transformer descriptions and prescribed seeds may depend non-uniformly on N. All model choices are fixed independently of the particular input. The vocabulary and numerical convention are fixed across input lengths. For a set of budgets, such as $O(\log N)$, the notation denotes the union over those budgets.

Input representation. For circuit simulations, vocabulary symbols have distinct fixed-width binary codes of length $w = \lceil \log_2 |\Sigma| \rceil$. The encoder $\mathrm{enc}_N : \{0,1\}^N \to \Sigma^+$ has $\mathsf{DLOGTIME}$-uniform AC^0 circuits and token length at most $N^{c_{\mathrm{enc}}}$ for a fixed constant c_{enc}. Its circuit representation includes the length and padded token codes, as specified below. The size parameter remains $N = |x|$, before this encoding. Representing each input bit by the corresponding token $\mathbf{0}$ or $\mathbf{1}$ is an allowed encoder.

Decoding rules and seeds. Fix an integer $K_0 \ge 1$ and let $h(N) = K_0 \lceil \log_2(N+1) \rceil$. Each rule $S_{N,t}(s; u) \in \Sigma$ takes only the score vector s and a seed word $u \in \{0,1\}^{h(N)}$. For every fixed u, it has polynomial-size TC^0 circuits, with size and depth bounds independent of the step and seed. A run uses $h(N)T(N)$ seed bits. In fixed-seed mode, one seed is prescribed for all inputs of length N. Greedy decoding ignores the seed and uses $F_{N,t}^{r_N} = F_N$, where F_N selects the first token in vocabulary order with maximal score.

In random-seed mode, the seed words are independent and uniform, independently of the input. A decision system has per-input success at least p for L if

$$\Pr_{r_N}\big[D_N(x; r_N) = \mathbf{1}\{x \in L\}\big] \ge p \qquad \text{for every } N \ge 1 \text{ and } x \in \{0,1\}^N.$$

Random-seed guarantees are stated separately from membership in $\mathsf{CoT}[T(N)]$. Conditioning on a seed gives a deterministic computation, whose selection rule may still depend on t. We count all emissions, including the answer token: $T(N) = 1$ is a single pass, and $T(N)$ emissions correspond to $T(N) - 1$ intermediate steps in Li et al. (2024). Uniformity requirements are given in Appendix A.5.

String-valued outputs. For Transformer computations of full answers, such as NLF, use the same generation process and a fixed extractor

$$g_N : \Sigma^{T(N)} \longrightarrow \{0,1\}^{\le q(N)} \cup \{\bot\}, \qquad q(N) \le N^{c_{\mathrm{out}}},$$

where c_{out} is fixed and g_N has polynomial-size TC^0 circuits. It depends on N but has no access to x or the seed. The same extractor is used for all inputs and seed realizations. The extracted answer is $g_N(W(x; r_N))$, and its length may vary with the transcript. Exact correctness for a target function f_N means equality to $f_N(x)$ on every input. Random-seed success is the probability of this equality. No uniformity is required of g_N unless explicitly stated. This convention defines full-answer computations without introducing a function-valued CoT class.

Figure 5 depicts generation with transcript-only extraction.

Finite-precision arithmetic. To relate our circuit model to the finite-precision CoT models of Merrill & Sabharwal (2024) and Li et al. (2024), we allow either of their numerical conventions below. Each convention specifies the representable numbers and their encoding, scalar arithmetic (including rounding and overflow), summation, and softmax. A Transformer family chooses one

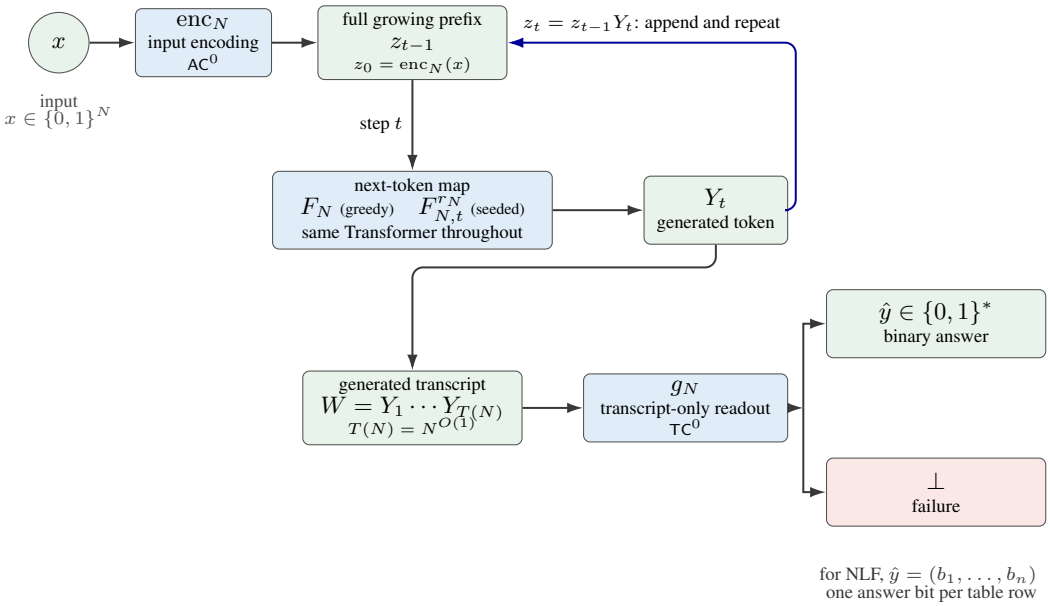


Figure 5: Autoregressive generation and transcript-only readout. Each token is appended before the next pass. g_N extracts the final answer.

convention for all input lengths, although its precision p_N may vary with N. For precision p, write $\mathbb{D}_p$ for the chosen datatype, encoded in $O(p)$ bits as in its source. Comparisons use numerical values.

Merrill–Sabharwal floating point. Use the p-precision floats and truncated scalar arithmetic of Merrill & Sabharwal (2024, Section 2.1). Their summation rule accumulates the scaled integer representations exactly and truncates only the final sum to a p-precision float, with the overflow convention of Merrill & Sabharwal (2023b, Appendix A, Definitions 13–15). Softmax truncates each exponential and normalizes using this sum and truncated division.

Li et al. fixed point. Use the format with no exponent bits in Li et al. (2024, Section 3.1). Round each elementary operation to the nearest representable value, breaking ties toward zero and saturating overflow at the endpoints. Sum from left to right, rounding after each addition. Round products before summing them. Softmax rounds the exponentials, sums them by this rule, and rounds each quotient.

Write mul_p, $\mathrm{sum}_{p,k}$, and $\mathrm{softmax}_{p,k}$ for the resulting operations. To make them total, return zero for a scalar operation that the source leaves undefined. A zero softmax denominator gives the zero vector. Decode unused bit patterns as zero. These defaults extend the source conventions. Definition A.3 separately requires the stated circuit bounds at precision p_N and the arities used on length-m prefixes. Finite precision alone does not supply those bounds or their uniformity.

Computing output bits. For maps $f_N : \{0,1\}^{a_N} \to \{0,1\}^{b_N}$ with $a_N, b_N = N^{O(1)}$, we use circuits C_N satisfying

$$C_N(x, j) = \big[f_N(x)\big]_j, \qquad x \in \{0,1\}^{a_N}, \quad j \in [b_N],$$

where j is encoded in binary. We say that all output bits of f are computable in TC^0 if the C_N have constant depth and size $N^{O(1)}$. Layer and head indices are treated as additional binary inputs when a family of component maps is considered jointly.

For Transformer computations on a prefix of length m, circuits are indexed by (N, m) and have size $(N + m)^{O(1)}$. Their depth and polynomial exponent are independent of both indices. A position $i \le m$ is supplied in binary to the embedding map. The same model parameters and positional rule are used for every m.

An encoding with polynomially bounded token length is represented by its binary length and a fixed-size array of token codes, padded with a fixed vocabulary symbol. Every length bit and token-code bit must satisfy the encoding's circuit bound. In particular, this applies to enc_N under the CoT encoding conventions above. Padding is only a circuit representation and is not part of the prefix passed to the Transformer.

For outputs in $\{0,1\}^{\le q(N)} \cup \{\bot\}$, the circuit produces a failure bit, a binary answer length, and a $q(N)$-bit payload. A successful answer y is represented by $(0, |y|, y0^{q(N)-|y|})$, and $\bot$ by $(1, 0, 0^{q(N)})$. The length uses $\lceil \log_2(q(N)+1) \rceil$ bits. Every bit of this representation, including the length and failure flag, must satisfy the same circuit bounds.

Causal state transition. For each N, the model $\mathcal{T}_N$ is specified by positive integers p_N, d_N, L_N and $H_{\ell,N}$ for $0 \le \ell < L_N$, together with maps

$$\begin{aligned} \phi_N &: \Sigma \times \mathbb{N}_{\ge 1} \to \mathbb{D}_{p_N}^{d_N}, \\ s_{\ell r,N} &: (\mathbb{D}_{p_N}^{d_N})^2 \to \mathbb{D}_{p_N}, & v_{\ell r,N} &: \mathbb{D}_{p_N}^{d_N} \to \mathbb{D}_{p_N}^{d_N}, \\ F_{\ell,N} &: (\mathbb{D}_{p_N}^{d_N})^{H_{\ell,N}+1} \to \mathbb{D}_{p_N}^{d_N}, & o_N &: \mathbb{D}_{p_N}^{d_N} \to \mathbb{D}_{p_N}^{|\Sigma|}, \end{aligned}$$

for $r \in [H_{\ell,N}]$. Given any finite prefix $z = z_1 \cdots z_m \in \Sigma^m$ with $m \ge 1$, define $h_i^0 = \phi_N(z_i, i)$ and, for $0 \le \ell < L_N$ and $r \in [H_{\ell,N}]$,

$$\begin{aligned} (\alpha_{ij}^{\ell r})_{j=1}^{i} &= \mathrm{softmax}_{p_N,i}\big((s_{\ell r,N}(h_i^\ell, h_j^\ell))_{j=1}^{i}\big), \\ a_i^{\ell r} &= \sum_{j=1}^{i} \alpha_{ij}^{\ell r} v_{\ell r,N}(h_j^\ell), \\ h_i^{\ell+1} &= F_{\ell,N}(h_i^\ell, a_i^{\ell 1}, \ldots, a_i^{\ell H_{\ell,N}}). \end{aligned}$$

The second line uses mul_{p_N} and $\mathrm{sum}_{p_N,i}$ coordinatewise. Restricting to $j \le i$ is the causal mask. The score vector in Definition A.3 is

$$\ell_{\mathcal{T}_N}(z) = o_N(h_m^{L_N}).$$

Standard scaled dot-product attention is obtained by taking $s(x,y)$ to be the rounded scaled dot product of Qx and Ky, taking $v(y) = Vy$, and allowing $F_{\ell,N}$ to contain the output projection, residual connection, normalization, and feed-forward block.

Resource bounds and uniformity. Let $H_N = \max_{0 \le \ell < L_N} H_{\ell,N}$. Every output bit of ϕ_N, $s_{\ell r,N}$, $v_{\ell r,N}$, $F_{\ell,N}$, and o_N must satisfy the circuit bounds above. For each N, these maps have a single finite description of total length $N^{O(1)}$. In particular, this description determines ϕ_N at every positive position, without supplying new parameters as the prefix grows. The family is DLOGTIME-uniform when one random-access machine, given size parameters (N, m), a tag identifying a dimension, component circuit, gate, connection, or hard-wired bit, and the relevant binary indices, returns the requested description bit in $O(\log(N+m))$ time. This requirement includes the numerical-operation circuits. No such machine is required in the non-uniform model.

A.3 Single-pass simulation

Lemma A.5 (Single-pass simulation, adapted from Merrill & Sabharwal, 2023b, Thm. 2)**.** *Let $\mathcal{T} = \{\mathcal{T}_N\}$ satisfy Definition A.3, with the selection rules of Appendix A.2. On a prefix of length $m \ge 1$, the output bits of a single forward pass and its fixed-seed next-token map $F_{N,t}^{r_N}$ have non-uniform constant-depth threshold circuits of size $(N+m)^{O(1)}$. The size and depth bounds are independent of the prescribed seed and generation step. The same bound holds after composition with any fixed non-uniform TC^0 readout with this size bound. For prefixes of polynomial length in N, including variable-length prefixes supplied with their length, these maps have non-uniform polynomial-size TC^0 circuits in N.*

Proof. Fix N, $m \ge 1$, a generation step t, and a seed word $r_{N,t}$. Compute the initial states at all m positions in parallel. Within each layer, compute the scores and values for all heads and all causal

position pairs in parallel, followed by the rounded softmax, weighted sums, and layer maps. Each stage has constant depth and size polynomial in $N+m$ by the component and arithmetic assumptions. There are only $L_N = O(1)$ layers, so their composition has constant depth and size $(N+m)^{O(1)}$. Compose the score-vector circuit with the assumed polynomial-size TC^0 circuit for $S_{N,t}(\cdot\,; r_{N,t})$. This computes $F^{r_N}_{N,t}$ with the same asymptotic bound, independently of t and $r_{N,t}$. Composing with a readout satisfying the stated bound preserves it.

Now let $B(N) = (N+1)^c$ be any fixed polynomial dominating the prefix lengths under consideration. Represent the prefix by its binary length and $B(N)$ token slots. For every $1 \le m \le B(N)$, run the length-m circuit on the first m slots in parallel, and select the output whose index equals the supplied length. The number of copies and the size of each copy are polynomial in N, and the selector has constant depth. This proves the bound for variable-length prefixes as well. The padding length $B(N)$ belongs to the circuit representation of these computations and imposes no limit on the Transformer itself. □

A.4 Logarithmic-CoT simulation

This subsection reproves, under our model conventions, the known logarithmic-CoT simulation theorem of Li et al. (2024, Thm. 3.2): constant-depth, log-precision fixed-point Transformers with $O(\log N)$ CoT steps are simulable by polynomial-size, constant-depth threshold circuits. In our notation, the decision bound is $\mathsf{CoT}[O(\log N)] \subseteq \mathsf{TC}^0/\mathrm{poly}$ (Theorem A.7). The proof adapts their parallel trace enumeration using the preceding single-pass lemma, which covers both arithmetic conventions of Appendix A.2. Lemma A.6 additionally supplies circuits for complete fixed-seed transcripts and string-valued readouts.

Decoding circuits. Under Appendix A.2, a run uses $h(N)T(N) = O(T(N)\log N)$ prescribed or sampled seed bits. For example, sampling from a finite-precision distribution over the constant-size vocabulary can compare a random word with the cumulative probability thresholds, a constant-depth threshold computation. More generally, for each fixed step and seed word, the selection rule is a Boolean function of the $O(|\Sigma|\log N) = O(\log N)$ score bits. Its truth table gives a non-uniform depth-2 circuit of polynomial size, with a bound independent of the step and seed. This justifies the non-uniform scope of decoding strategies in Proposition C.3. The uniform refinements additionally require uniform selection circuitry and a uniform prescribed seed family whenever the rule uses the seed.

Lemma A.6 (Transcript and full-output simulation)**.** *For any generation process satisfying Definition A.4 with $T(N) = O(\log N)$, the binary codes of $W(x; r_N)$ and the decision bit $D_N(x; r_N)$ have non-uniform polynomial-size constant-depth threshold circuits for each fixed seed. Composing with a string readout from Appendix A.2 gives such circuits for the full answer, including its length and failure flag. The depth and polynomial size bounds are independent of the prescribed seeds.*

Proof. We adapt the trace-enumeration argument of Li et al. (2024, Thm. 3.2). Prescribe any seed family $r = \{r_N\}$ and let $T(N) \le c\log N$. Every candidate prefix has length at most $N^{c_{\mathrm{enc}}} + T(N) = N^{O(1)}$. The variable-length construction in Lemma A.5 therefore supplies polynomial-size constant-depth threshold circuits for every next-token map on these prefixes. The same seed word $r_{N,j}$ is used at step j in every candidate branch.

Let $w = \lceil \log_2 |\Sigma| \rceil$. Enumerate strings $\tau \in \{0,1\}^{wT(N)}$, interpreted as $T(N)$ token codes $(a_1, \ldots, a_{T(N)})$, and let $V(\tau)$ test that every code belongs to the vocabulary. There are $2^{wT(N)} = N^{O(1)}$ branches, of which $|\Sigma|^{T(N)}$ represent valid transcripts. Using the fixed-seed next-token maps, define the branch predicate

$$P_\tau(x) = V(\tau) \wedge \bigwedge_{j=1}^{T(N)} \left[F^{r_N}_{N,j}(\mathrm{enc}_N(x) a_{<j}) = a_j \right].$$

The encoding supplies the input prefix and its length. Appending the fixed candidate tokens $a_{<j}$ and updating the length use polynomial-size AC^0 circuits. The Transformer is defined on every valid candidate prefix, including those inconsistent with its actual run. The underlying Boolean circuits give arbitrary values on unused token codes, which V rejects. Every consistency test uses the input

encoding and fixed candidate tokens, so all steps are checked in parallel. Their conjunction adds constant depth.

Exactly one branch has $P_\tau(x) = 1$: its first token is forced by the initial prefix, and each subsequent token is forced by the earlier tokens and the fixed seed. For each transcript-code bit q, output

$$\bigvee_{\tau \in \{0,1\}^{wT(N)}} (P_\tau(x) \wedge \tau_q)\,.$$

This reconstructs the generated transcript. Testing its last token yields $D_N(x; r_N)$. Composing with the circuit for g_N yields the full extracted answer. There are polynomially many branches, $O(\log N)$ checks per branch, and polynomial-size component circuits. The final conjunctions, ORs, and readout add only constant depth, with bounds independent of the prescribed seeds. Thus the construction also applies to each realization of random-seed decoding. □

Theorem A.7 (Logarithmic-CoT simulation, adapted from Li et al., 2024, Thm. 3.2)**.**

$$\mathsf{CoT}[O(\log N)] \subseteq \mathsf{TC}^0/\mathrm{poly}.$$

The simulating circuit bounds are independent of the prescribed seed family.

Proof of Theorem A.7. For a language in $\mathsf{CoT}[O(\log N)]$, fix its prescribed seed family and apply Lemma A.6 to its decision bit. The resulting circuits decide the language, proving membership in $\mathsf{TC}^0/\mathrm{poly}$. □

A.5 Uniform versions of the simulations

The following refinements establish uniformity of the preceding circuit constructions. Uniformity uses the random-access circuit-description convention in Appendix A.2, including its requirement on numerical-operation circuits. Selection circuits with the seed word supplied as an input must be DLOGTIME-uniform jointly in N and the polynomially bounded step index t. For full-answer computations, an additional extractor must also be DLOGTIME-uniform in N.

Uniform decoding seeds. Use the seed-word length $h(N)$ from Appendix A.2. A prescribed seed family $r = \{r_N\}$, with $r_N = (r_{N,1}, \ldots, r_{N,T(N)})$ for a polynomially bounded horizon, is *uniform* if a single machine, given binary (N, t, k), returns $r_{N,t}[k]$ in $O(\log N)$ time for $1 \le t \le T(N)$ and $1 \le k \le h(N)$. This condition concerns the prescribed seed family, separately from uniformity of the selection circuits. Arbitrary fixed seeds, including those chosen in the amplification argument, need not satisfy it. When the selection rule ignores the seed, no seed condition is needed.

Theorem A.8 (Uniform forward-pass simulation)**.** *Let $\mathcal{T}$ be a* DLOGTIME*-uniform family satisfying Definition A.3, and let any additional readout have* DLOGTIME*-uniform* TC^0 *circuits of size* $(N + m)^{O(1)}$*, jointly in* (N, m)*. Every output bit of a forward pass and the readout is computable by a* DLOGTIME*-uniform constant-depth threshold circuit of size* $(N + m)^{O(1)}$*. If the selection circuits are uniform with the seed word supplied as an input, the same bound holds for next-token selection, jointly in* (N, m, t) *for polynomially bounded* t*. For fixed-seed decoding, prescribing a uniform seed family preserves this bound for* $F^{r_N}_{N,t}$*. No seed condition is needed when the selection rule ignores the seed. For every fixed integer* $b \ge 1$*, the same maps on prefixes of length at most*

$$B(N) = 2^{b\lceil \log_2(N+1)\rceil},$$

supplied by their binary length and $B(N)$ *padded token slots, have* DLOGTIME*-uniform* TC^0 *circuits of polynomial size in* N*. Composition with a* DLOGTIME*-uniform* AC^0 *prefix encoding preserves this conclusion.*

Proof. The size and depth bounds follow from the proof of Lemma A.5. For uniformity, use the circuit-block construction of Merrill & Sabharwal (2023a, Section 6.1, Cor. 3.2). Write $M = N + m$ and choose a fixed integer K large enough that

$$S(M) = 2^{K\lceil \log_2(M+1)\rceil}$$

bounds all component circuit sizes and index ranges. As in Merrill & Sabharwal (2023a, Lems. 3–4), pad circuit blocks to this power-of-two bound. Name a gate by a fixed tuple of fields for its stage, layer, positions, head, coordinate, and local gate index. These fields have fixed widths for each (N, m) and total length $O(\log M)$. The number of padded blocks and their sizes remain polynomial in M.

A description query within a block invokes the corresponding uniform component description. Connections between blocks identify the relevant position, head, coordinate, and bit fields. The causal mask is the comparison $j \leq i$. The block offsets use shifts, while field extraction, comparisons, addition, and subtraction take $O(\log M)$ time. Together with a constant number of calls to the component description machines, this gives $O(\log M)$ time per gate or connection query. Parameter bits are supplied by the uniform model description. Compose the uniform selection circuit $S_{N,t}$, initially leaving the seed-word bits as inputs, and the assumed uniform readout in the same way. Since t is polynomially bounded in N, its index has $O(\log M)$ bits. To prescribe a uniform seed, replace each seed input by the bit returned by its description machine in $O(\log N)$ time. If $S_{N,t}$ ignores the seed, no such query is needed.

For the variable-length circuit from Appendix A.3, pad all length-m copies, $1 \leq m \leq B(N)$, to one common power-of-two block bound polynomial in N. A gate name records m and its local index, so its description is obtained by a call to the preceding machine on (N, m) in $O(\log N)$ time. The length-equality tests and output selector have uniform connections determined by these fields. Computing $B(N)$ and block offsets uses shifts and multiplication by fixed constants. The same block construction composes the uniform prefix encoding with this circuit. The supplied prefix length remains circuit data throughout. □

Theorem A.9 (Uniform logarithmic-CoT simulation)**.** *Consider a decision system from Definition A.4 with $T(N) = O(\log N)$. Suppose its Transformer family and input encoder are* DLOGTIME*-uniform, $T(N)$ is computable from binary N in $O(\log N)$ time, and the selection circuits are uniform jointly in N and the step index with the seed word supplied as an input. If the prescribed seed family is uniform, then $D_N(x; r_N)$ has polynomial-size* DLOGTIME*-uniform TC^0 circuits. Thus every language decided by such a system belongs to uniform TC^0. The same construction computes the transcript and, with a uniform string readout, the full answer representation. No seed-family hypothesis is needed when selection ignores the seed.*

Proof. Use the trace-enumeration circuit of Lemma A.6. Its correctness, polynomial size, and constant depth are established there. It remains to verify uniformity.

Each candidate is already named by $wT(N) = O(\log N)$ bits, where $w = \lceil \log_2 |\Sigma| \rceil$ is fixed. Since $T(N)$ is computable in $O(\log N)$ time, the description machine can locate token a_j at offset $w(j-1)$ in the candidate name. The token-validity tests and the connections to the conjunctions and final selector are determined by these fields.

The prefix construction is uniform AC^0: if the encoded input length is ℓ, slot $i \leq \ell$ copies the input-encoding slot, and appended token a_s is selected by $i = \ell + s$ for $s < j$. The new length is $\ell + j - 1$. Theorem A.8 supplies the uniform variable-length simulation for each such prefix. Compose it with $S_{N,j}(\,\cdot\,; r_{N,j})$. Each substituted seed bit $r_{N,j}[k]$ is obtained in $O(\log N)$ time by the seed hypothesis. A seed-independent rule uses no seed bits, so the same construction applies without that hypothesis. Transcript reconstruction reads bits of the candidate name, and the final-token test uses a fixed code comparison. For a full answer, compose the reconstructed transcript with the assumed uniform extractor.

Pad branch and component blocks to powers of two as in the proof of Theorem A.8. Gate names record the candidate, step, component tag, and local gate index. A description query uses field extraction, comparisons, shifts, addition or subtraction, and a constant number of calls to the assumed uniform description machines. Each query therefore takes $O(\log N)$ time, proving the claimed uniformity. □

Fixed-seed lower bounds under uniform separation. Theorem A.9 implies the fixed-seed statement of Theorem 3.2, including Note C.1, for systems satisfying its uniformity hypotheses under the weaker assumption $\mathsf{NC}^1 \not\subseteq \mathsf{TC}^0$. Compose the uniform simulation with the always-valid uniform AC^0 reduction of Theorem 3.1. Correctness on its single-row Boolean instances would place every

$L \in \mathsf{NC}^1$ in uniform TC^0, a contradiction. The random-seed hardness arguments in the main text use the non-uniform separation because amplification supplies seed tuples that may depend arbitrarily on N.

A.6 Compatibility with the cited CoT models

Theorem A.10 (Li et al., 2024, Thms. 3.3, 3.5 and Merrill & Sabharwal, 2024, Thms. 2–3)**.** *In the cited deterministic models, constant-depth Transformers with $\Theta(N)$ CoT steps can decide an NC^1-complete language, such as the S_5 word problem. With polynomially many steps, the constant-precision model of Li et al. (2024) decides exactly the languages in $\mathsf{P/poly}$. The original uniform log-precision, saturated-attention, projected-pre-norm, strictly causal model of Merrill & Sabharwal (2024) decides exactly the languages in P with polynomially many steps.*

Generation, readout, and length. With greedy selection and an encoder that represents each input bit by its corresponding vocabulary symbol, Definition A.4 uses the autoregressive decision convention of Merrill & Sabharwal (2024) and Li et al. (2024, Section 2.1, Definition 3.4). Our $T(N)$ emissions include the answer, so they correspond to $T(N) - 1$ intermediate steps in Li et al. (2024). We retain Li et al.'s non-uniform family convention and additionally allow uniform AC^0 input encoders and seeded selection. The proofs above include their circuit costs. String-valued readout is a separate convention for full-answer computations, rather than part of the decision class $\mathsf{CoT}[T(N)]$.

Precision and architecture. Appendix A.2 retains both numerical conventions: Merrill and Sabharwal truncate after exact accumulation, while Li et al. round each addition. Each cited theorem keeps its original numerical and architectural assumptions. In particular, the precision in Merrill & Sabharwal (2024) is logarithmic in input plus generation length, hence $O(\log N)$ on polynomial runs. Theorem A.10 retains their saturated-attention and normalization assumptions. Our short-CoT theorem applies Li et al.'s trace enumeration to the one-step circuit bound in Definition A.3, under either qualifying convention.

Arithmetic in the circuit simulator. Proposition 3.3 chooses Li et al.'s constant-precision fixed-point construction. Scalar operations then have fixed finite tables, and iterated fixed-point addition has TC^0 circuits (Li et al., 2024, Appendix D.2). Parallel scalar operations and these sums give the required bounds.

The address-bit induction in Li et al. (2024, Lemma E.3) works at fixed precision for any address length. If B is the greatest representable value, each bit contributes $(B, -B)$ for agreement and $(-B, -B)$ for disagreement. Rounded accumulation yields 0 exactly for equal addresses and $-B$ otherwise. Their rounded exponentials are 1 and 0, respectively. Thus a required input read has softmax denominator 1 and retrieves its bit exactly. The dummy second input of a NOT gate has denominator zero. Our default supplies 0, which its Boolean gate map ignores. Reads at unused prompt positions affect only scratch coordinates. Residual connections preserve the token and address coordinates used later. Thus totalizing the arithmetic preserves every emitted gate value.

Finite prefixes. Every run in our CoT model visits positions at most $N^{c_{\mathrm{enc}}} + T(N)$, a polynomial in N. The construction of Li et al. (2024) specifies a finite positional table covering its prescribed computation. Extend that table by a fixed default vector at every later position. Its description remains polynomial in N, its lookup has constant-depth polynomial-size circuits, and every token in the prescribed run is unchanged. Thus the full finite-prefix formulation of Merrill & Sabharwal (2024), without a separate context-capacity parameter, preserves the polynomial CoT constructions used here.

Proof of Proposition 3.3. The polynomial-time algorithm for NLF gives a polynomial-size Boolean circuit for its failure flag, answer length, and padded output bits. Make each such bit the value of a non-input gate, adding two NOT gates where necessary. The constant-precision construction of Li et al. (2024, Thm. 3.3) emits the gate values in a fixed order using polynomially many tokens. Appendix A.6 verifies compatibility with our arithmetic and prefix conventions. A fixed TC^0 extractor projects the designated output-gate values. The generator uses greedy selection and ignores every seed, so the extracted answer is exact under either seed convention. □

B Completeness of Nested Logical Filtering

The membership proof evaluates Boolean sentences, and the hardness proof converts balanced sentences into table queries. We first state the formula-evaluation result used in both directions.

B.1 Boolean formula evaluation

Definition B.1 (BFVP)**.** The *Boolean Formula Value Problem* asks whether $\varphi(a) = 1$ for a fully parenthesized Boolean formula φ and an assignment $a \in \{0,1\}^m$. The formula uses literals $x_1, \ldots, x_m, \neg x_1, \ldots, \neg x_m$, constants $0, 1$, and fan-in-2 $\wedge$ and $\vee$ gates. Symbols have fixed-width codes and variable indices are delimited binary strings. Malformed encodings are rejected. A *Boolean sentence* has only constant leaves and needs no assignment. If N is the total encoding length, $\mathrm{BFVP}_{\mathrm{bal}}$ denotes the language of true, valid instances of depth at most $\lceil \log_2 N \rceil$, with leaves at depth zero. Instances exceeding this bound are rejected.

Theorem B.2 (Buss, 1987; Buss et al., 1992)**.** $\mathsf{BFVP} \in \mathsf{ALOGTIME} = \mathsf{DLOGTIME}$-*uniform* NC^1*, and* $\mathsf{BFVP}_{\mathrm{bal}}$ *is* NC^1*-complete under* $\leq_m^{\mathsf{AC}^0}$*. More precisely, every* $L \in \mathsf{NC}^1$ *has a* $\mathsf{DLOGTIME}$*-uniform* AC^0 *reduction* $x \mapsto \psi_x$ *whose output is always a valid, completely balanced Boolean sentence, has polynomial length depending only on* $|x|$*, and has value* 1 *exactly when* $x \in L$*.*

Buss (1987) gives uniform NC^1 formula evaluation and recognition of parenthesized syntax. Buss et al. (1992, Theorem 5.1 and Lemma 5.2) construct the balanced output and compute each output symbol in deterministic logarithmic time. The reduction below uses this construction directly, without requiring a uniform implementation of the formula-balancing theorem of Spira (1971).

B.2 Encoding conventions

Use a fixed constant-width binary code for all symbols, including a field delimiter and a blank. A decision record lists the fields (n, m, b, T, Q, i), separated by delimiters. A function record omits the last field. Write n, m, b and all indices in binary without leading zeros. Require $n, m, b \geq 1$ and indices in range. The table field contains exactly nmb digits in row-major order, using b bits per entry, most significant bit first. Query constants also use b bits. Entries and constants may have leading zeros.

Write atoms as $[j \bowtie v]$, with comparisons interpreted as in Definition 2.1. Build queries from these atoms using binary $\wedge$ and $\vee$, with every operation fully parenthesized. Blanks may occur only in the query and are ignored when parsing.

Let N be the total bit length. The decision language rejects malformed records, and the full function returns $\perp$ on them. Valid records satisfy $n, m, b \leq N$.

We write NLF_d and $\mathrm{NLF}_{\mathrm{dec},d}$ for the full-answer and decision variants restricted to queries of depth at most $\lceil \log_2 N \rceil$, counting atoms as depth zero. The decision variant rejects deeper queries, and the full-answer variant returns $\perp$.

B.3 Proof of Theorem 3.1

Proof. All circuit families below are $\mathsf{DLOGTIME}$-uniform.

Membership. Figure 6(a) shows the two stages. We first check the encoding and evaluate all atoms in parallel. We then evaluate the resulting Boolean sentence.

Uniform TC^0 arithmetic and counting circuits (Hesse et al., 2002) check the field separators, binary syntax, index ranges, and required table length nmb. First reject dimensions or field widths above N, so the remaining arithmetic uses $O(\log N)$ bits. Remove query blanks by routing each retained symbol to its rank among retained symbols. Test all candidate atom spans in parallel, checking the column index and the b-bit constant. Require disjoint atoms. Outside them, allow only parentheses and connectives.

The circuit computes each referenced cell's position from the input fields and selects its bits with multiplexers. It places the cell and constant in fixed N-bit arrays, padding on the left with zeros.

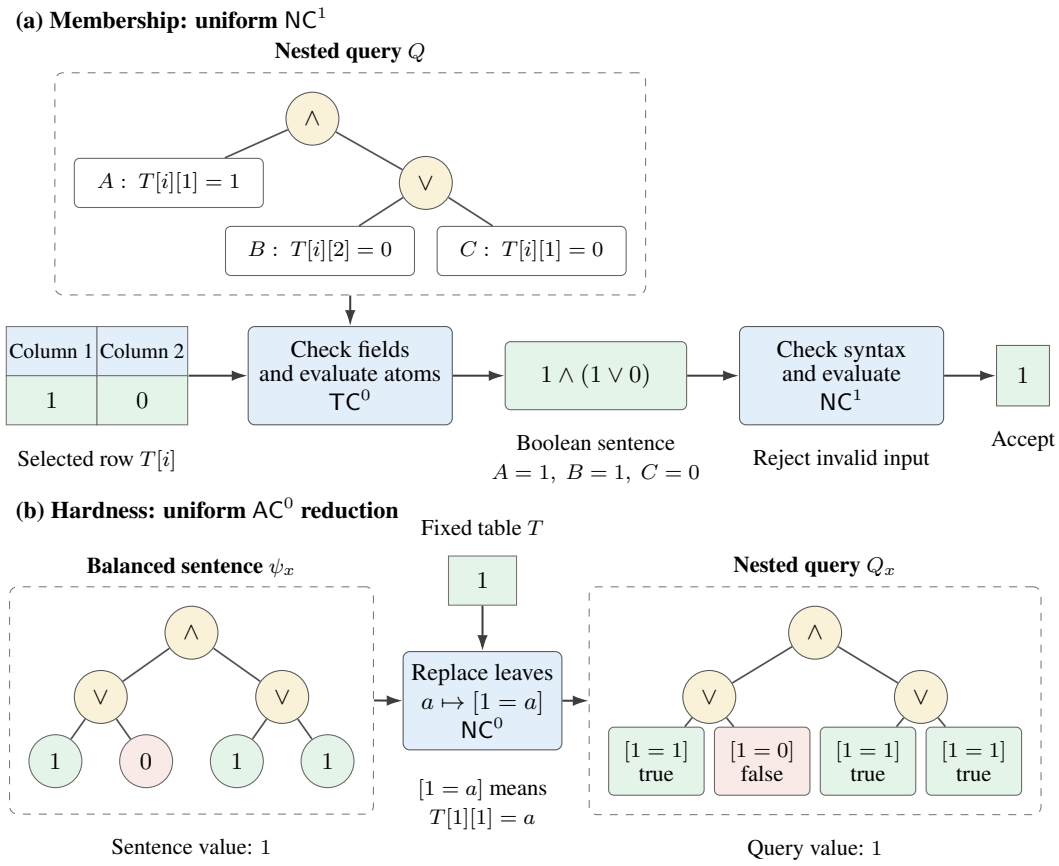


Figure 6: **The two directions of the completeness proof.** (a) Evaluate the comparison atoms on the selected row, then check and evaluate the resulting Boolean sentence. (b) The balanced sentence supplied by Theorem B.2 becomes a query on $T = [1]$ by replacing each constant a with $[1 = a]$. The tree structure and leaf values stay unchanged. Both examples yield the answer 1.

Equality and unsigned order have AC^0 circuits: compare corresponding bits or locate the first differing bit.

Replace each atom by its truth digit, keep parentheses and connectives, and compact by prefix ranks into N padded symbol slots, recording the actual length. These operations use polynomially many parallel tests and have constant depth. Buss (1987) gives uniform NC^1 algorithms for recognizing fully parenthesized Boolean sentences and evaluating them (Theorem B.2). Handle the compacted length by running these circuits for every length up to N in parallel and selecting the matching outputs. Accept exactly when all encoding and syntax checks pass and the sentence has value one. Since $\mathsf{TC}^0 \subseteq \mathsf{NC}^1$ and uniform NC^1 is closed under composition (Buss et al., 1992, Thm. 2.7), this proves membership, with no restriction on query depth.

Hardness. Fix $L \in \mathsf{NC}^1$ and $x \in \{0,1\}^s$. The construction of Buss et al. (1992, Thm. 5.1 and Lem. 5.2) gives a valid, completely balanced Boolean sentence ψ_x with value $\mathbf{1}\{x \in L\}$. Its polynomially bounded length and tree shape depend only on s. Each output symbol is computable in deterministic logarithmic time, so $x \mapsto \psi_x$ is a uniform AC^0 map (Buss et al., 1992, Thm. 2.4).

NLF query leaves must be table comparisons. Use $T = [1]$ and replace each constant leaf a of ψ_x by $[1 = a]$, which means $T[1][1] = a$. Keep the parentheses and connectives, and call the resulting query Q_x. Figure 6(b) illustrates the substitution: $[1 = 0]$ is false and $[1 = 1]$ is true. Each leaf keeps its value, so

$$Q_x([1]) = \mathrm{val}(\psi_x) = \mathbf{1}\{x \in L\}.$$

Pad every symbol replacement with query blanks to the same constant width. Add the fixed fields for the one-cell Boolean table and row 1, and call the resulting record $R_s(x)$. Each output bit depends on at most one source symbol, so this step is a uniform NC^0 map. Composition gives a uniform AC^0 reduction (Buss et al., 1992, Thm. 2.6). All outputs on s-bit inputs have one length $N_s = s^{O(1)}$, because the source sentences have one length for each s. Every output is valid, and $x \in L$ exactly when $R_s(x) \in \mathrm{NLF}_{\mathrm{dec}}$. Since L was arbitrary, this proves hardness. □

Restricted instances. The reduction uses only equality atoms and the table $T = [1]$. Its query tree is completely balanced, so its depth equals the logarithm of its number of leaves. Each leaf occupies at least one encoded bit, giving depth at most $\log_2 N_s$. Hardness therefore holds for $\mathrm{NLF}_{\mathrm{dec},d}$ as defined in Appendix B.2. Membership also holds: on a valid query, depth equals the largest excess of opening over closing parentheses in any prefix. Uniform TC^0 prefix counts check this bound, and parallel tests check that the table has one row and Boolean entries.

Full answers. For a function record, apply the same validity test and evaluate the query for row indices $1, \ldots, N$ in parallel. Return length n and the row values, padded with zeros after position n. Invalid inputs return the failure representation of Appendix A.2. This uses polynomial size and depth $O(\log N)$, so the full-answer representation is computable in uniform NC^1.

For hardness, omit the final row-index field and its delimiter from $R_s(x)$. The resulting function record $R_s^{\mathrm{fun}}(x)$ again has polynomial length depending only on s, satisfies the logarithmic depth bound, and has the one-bit answer $\mathbf{1}\{x \in L\}$. This always-valid uniform AC^0 reduction supplies the function hardness used in Section 3.2.

C Short-CoT Barriers

This appendix turns the circuit simulations into a lower bound for short CoT. Appendix A shows that a fixed-seed system with $O(\log N)$ emissions is simulated by polynomial-size, constant-depth threshold circuits, while Theorem 3.1 shows that $\mathrm{NLF}_{\mathrm{dec}}$ is NC^1-complete, even on one-row instances with logarithmic-depth queries. Under Assumption 2.2, an exact short-CoT solver on all valid inputs would therefore imply the forbidden containment $\mathsf{NC}^1 \subseteq \mathsf{TC}^0/\mathrm{poly}$.

Random-seed decoding needs an additional argument: the seed that succeeds on one input may fail on another. We amplify a per-input success probability of at least $2/3$ and then fix one tuple of seeds that works for every input of a given length. This yields the same worst-case barrier for random-seed solvers and extends to full answers and the single-pass case. Finally, Proposition C.3 gives an average-case ceiling under a separate distributional circuit-hardness hypothesis; the worst-case separation alone does not bound average-case accuracy.

Note C.1 (Single-pass Transformers). Taking $T(N) = 1$ in Theorem 3.2 rules out a single-pass solver under either success requirement.

C.1 Amplification and seed fixing

Lemma C.2 (Adleman's argument for threshold circuits)**.** *Consider a random-seed procedure whose seed is sampled independently of its input. Suppose there are constants a, d such that, for every input length N and every seed r, the fixed-seed output is computed by a non-uniform Boolean threshold circuit $C_{N,r}$ of size at most N^a and depth at most d. If the procedure decides a language L with per-input success probability $\geq 2/3$ over its seed, then $L \in \mathsf{TC}^0/\mathrm{poly}$.*

Proof. Run $k = cN$ independent copies of the procedure with independent seeds $r_1, \ldots, r_k$ and output the majority of their answers. For each fixed input x, a Chernoff bound gives failure probability $\leq e^{-2k(1/6)^2} = e^{-cN/18} < 2^{-N}$ for a suitable constant c. Hence

$$\Pr_{r_1,\ldots,r_k}[\exists x \in \{0,1\}^N : \mathrm{MAJ}(C_{N,r_1}(x), \ldots, C_{N,r_k}(x)) \neq L(x)] < 2^N \cdot 2^{-N} = 1,$$

so some seed tuple $(r_1^*, \ldots, r_k^*)$ makes the majority answer correct on *all* inputs of length N. Fix this tuple, independently of which length-N input is supplied, and choose the corresponding fixed-seed

circuits. Their parallel composition with one majority gate has size $O(kN^a) = O(N^{a+1})$ and depth at most $d + 1$. The tuple may vary with N and is hard-wired as non-uniform advice. Hence $L \in \mathsf{TC}^0/\mathrm{poly}$. This is the circuit-level Adleman argument. See Hajnal et al. (1993, Lem. 4.1 and Prop. 4.2) and Arora & Barak (2009). □

C.2 Worst-case NLF lower bounds

Proof of Theorem 3.2. Fix any $L \in \mathsf{NC}^1$ and the reduction R_s from Theorem 3.1. All its outputs on s-bit strings are valid single-row Boolean instances of one length $N_s = s^{O(1)}$, satisfying the logarithmic query-depth bound.

Fixed-seed decoding. Suppose a decision system is correct on these valid instances using $O(\log N)$ emissions. Lemma A.6, which proves Theorem A.7, supplies a threshold circuit C_{N_s} for its decision bit D_{N_s} at the prescribed seed, with size $N_s^{O(1)}$ and constant depth. Composing with R_s gives a non-uniform TC^0 circuit deciding L on s-bit inputs. Its size is polynomial in s, and the reduction adds only constant depth. Applying this argument to every $L \in \mathsf{NC}^1$ would give $\mathsf{NC}^1 \subseteq \mathsf{TC}^0/\mathrm{poly}$, contrary to Assumption 2.2.

Random-seed decoding. For each target-length seed r, the composed map $x \mapsto C_{N_s,r}(R_s(x))$ has the same polynomial size and constant depth bounds. The seed is independent of the source input, because N_s depends only on s. Per-instance correctness on the valid reduction outputs gives per-input success at least $2/3$ for this source-language procedure. Apply Lemma C.2 at source length s: $O(s)$ independent runs and a union bound over the 2^s source strings yield a fixed seed tuple deciding L. This again places every $L \in \mathsf{NC}^1$ in $\mathsf{TC}^0/\mathrm{poly}$. No success guarantee on malformed or out-of-restriction target strings is used.

For full NLF, use R_s^{fun} and its corresponding target length instead. Apply the full-output part of Lemma A.6 to the string readout of Appendix A.2, and test whether the extracted answer is the one-bit string 1, treating every other output as rejection. On the reduction outputs, the correct vector has one bit, equal to $\mathbf{1}\{x \in L\}$, so this test preserves exact correctness and cannot decrease a per-input success guarantee. The same amplification applies without giving the extractor an input or a row index. Taking $T(N) = 1$ gives Note C.1. All emission bounds are measured in the binary input length. The hard instances have one table row and a growing query. □

C.3 Average-case NLF bound

Proposition C.3 (‡, asymptotic average-case ceiling)**.** *Let $\mathcal{I}$ be the set of bit lengths admitting a valid* NLF *function record. Suppose distributions $\mathcal{E}_N$ on valid length-N records satisfy the following: for every non-uniform family $\{C_N\}$ of polynomial-size, constant-depth threshold circuits with the full-answer representation of Appendix A.2, there is a threshold N_C such that*

$$\Pr_{I \sim \mathcal{E}_N}\big[C_N(I) = \mathrm{NLF}(I)\big] \le \frac{2}{3} \qquad \text{for every } N \in \mathcal{I},\ N \ge N_C.$$

Then every generation process from Definition A.4 using $O(\log N)$ emissions and a string readout as in Appendix A.2, under either seed convention, has a threshold N_0 such that

$$\Pr_{I \sim \mathcal{E}_N,\ r_N}\big[g_N(W(I; r_N)) = \mathrm{NLF}(I)\big] \le \frac{2}{3} \qquad \text{for every } N \in \mathcal{I},\ N \ge N_0.$$

Here r_N is constant at its prescribed value in fixed-seed mode and is uniformly sampled in random-seed mode.

Proof of Proposition C.3. Fix a Transformer generation process from Definition A.4 with $O(\log N)$ emissions and a string readout g_N as in Appendix A.2. For each N, let μ_N be the point mass at its prescribed seed in fixed-seed mode, or the uniform distribution on its seed words in random-seed mode. For each admissible length $N \in \mathcal{I}$, set

$$p_N := \Pr_{I \sim \mathcal{E}_N,\, r_N \sim \mu_N}\big[g_N(W(I; r_N)) = \mathrm{NLF}(I)\big]$$

and, for a fixed seed r, let

$$q_N(r) := \Pr_{I \sim \mathcal{E}_N}\big[g_N(W(I; r)) = \mathrm{NLF}(I)\big].$$

Property	Single pass	CoT $O(\log N)$	Polynomial CoT
Fixed-seed decision upper bound	$\mathsf{TC}^0/\mathrm{poly}$	$\mathsf{TC}^0/\mathrm{poly}$ (Thm. A.7)	P/poly / P
Exact NLF computation	No	No	**Yes** (Prop. 3.3)
Eventual success on $\mathcal{E}_N$	$\leq 2/3^{\ddagger}$	$\leq 2/3^{\ddagger}$	1 attainable
Transcript bound (Thm. D.2)	$K = \Omega(n)$	$K = \Omega(n)$	$K = \Omega(n)$
Noisy direct-readout success (Thm. 4.4(a))	$\leq (1-\varepsilon)^{\Omega(n)}$	$\leq (1-\varepsilon)^{\Omega(n)}$	$\leq (1-\varepsilon)^{\Omega(n)}$

Table 4: Supplementary computational, information, and reliability guarantees, with fixed vocabulary. The negative exact-computation results are conditional on Assumption 2.2. ‡: Proposition C.3. P/poly, P: respective models of Theorem A.10. The transcript bound assumes high-accuracy transcript-only recovery on $\mathcal{D}_n$. Noise follows Definition 4.1.

Then $p_N = \mathbb{E}_{r\sim\mu_N}[q_N(r)]$, so there exists a seed r_N^* in the support of μ_N, chosen separately for each $N \in \mathcal{I}$, such that $q_N(r_N^*) \geq p_N$. In fixed-seed mode this is just the prescribed seed. At other lengths choose an arbitrary seed from the corresponding support. By Lemma A.6, the fixed-seed map

$$C_N(I) := g_N(W(I; r_N^*))$$

is computed by a non-uniform polynomial-size constant-depth threshold circuit with the full-answer representation. A single polynomial bounds the size at every input length. The Transformer description and chosen seed may be hard-wired. Thus $\{C_N\}$ is one circuit family covered by the hypothesis, giving $q_N(r_N^*) \leq 2/3$ at every sufficiently large admissible length. Since $p_N \leq q_N(r_N^*)$, the claimed bound follows. □

D Transcript Length and Errors Without Injected Noise

This appendix separates the output needed to communicate an answer from errors made while generating it. On the simple distribution $\mathcal{D}_n$, the answer is a uniform n-bit string even though computing it is trivial. We prove that high-accuracy recovery from the transcript alone therefore requires $K = \Omega(n)$ fixed-vocabulary tokens (Theorem D.2). This bound explains why exact computation can still require a long textual output. We then identify a condition under which generation errors make that output unreliable without the injected channel of Section 4: positive mean conditional log-loss of decoded row bits directly yields exponentially small exact-answer success (Proposition D.3). This error premise must be established for the generator: exact polynomial-length CoT constructions (Proposition 3.3) show that transcript length alone does not force errors.

D.1 Transcript-length lower bound

An output-only transcript must convey the answer even when the generator can compute it exactly. This requirement already appears on easy queries.

Definition D.1 (Uniform-answer distribution)**.** For any fixed $m = m(n) \geq 1$, $\mathcal{D}_n$ draws independent fair bits for all cells of $T \in \{0,1\}^{n\times m}$, uses field width $b = 1$, and fixes the query $Q = [1 = 1]$ without blanks. Then $Y^* = \mathrm{NLF}(T, Q)$ is uniform on $\{0,1\}^n$, and the encoded length depends only on $n, m(n)$.

Theorem D.2 (Transcript information bound)**.** *Let $|\Sigma| \geq 2$. Let an arbitrary randomized generator emit $W \in \Sigma^{\leq K}$ on input drawn from $\mathcal{D}_n$, and let $\hat{Y} = g(W)$ for a fixed transcript-only extractor. Any stopping rule with maximum length K is allowed. Then*

$$\Pr[\hat{Y} \neq Y^*] \geq 1 - \frac{(K+1)\log_2|\Sigma| + 1}{n}.$$

In particular, success $1-\delta$ requires $K \geq ((1-\delta)n - 1)/\log_2|\Sigma| - 1$.

For fixed vocabulary, high-accuracy recovery on $\mathcal{D}_n$ therefore requires a maximum transcript budget $\Omega(n)$. This is a distributional information bound, not a claim about the compressibility of every individual answer. It also permits arbitrary decoders and does not by itself imply an exponential noise penalty. The latter follows from the direct-readout restriction below.

This bound isolates output information from computation: under $\mathcal{D}_n$, computing the answer merely copies the first column, but the answer still ranges over all n-bit strings. A short program that performs this copying does not contradict the bound. Its interpreter receives the original table, whereas the extractor in Theorem D.2 receives only the transcript. For the same reason, coding several answer bits into a token is already accounted for by $\log_2 |\Sigma|$, and access to a clean input at decoding changes the information available to the receiver.

Proof of Theorem D.2. Let $I \sim \mathcal{D}_n$, $Y^* = \mathrm{NLF}(I)$ uniform on $\{0,1\}^n$ (Definition D.1), and W the transcript. Replace every output of $g(W)$ outside $\{0,1\}^n$ by 0^n, and denote the resulting estimate by $\hat{Y}$. This cannot increase the error probability, so a lower bound on its error also bounds the original extractor's error. Since W takes values in $\Sigma^{\le K}$,

$$H(W) \;\le\; \log_2\Big(\textstyle\sum_{j=0}^{K} |\Sigma|^j\Big) \;\le\; \log_2\big(|\Sigma|^{K+1}\big) \;=\; (K+1)\log_2|\Sigma| \qquad (|\Sigma| \ge 2),$$

and by data processing $H(\hat{Y}) \le H(W)$. Fano's inequality (Cover & Thomas, 2006, Thm. 2.10.1) applied to estimating Y^* from $\hat{Y}$ over the alphabet $\{0,1\}^n$ gives, with $P_e = \Pr[\hat{Y} \ne Y^*]$,

$$H(Y^* \mid \hat{Y}) \;\le\; h_2(P_e) + P_e \log_2\big(2^n - 1\big) \;\le\; 1 + nP_e.$$

On the other hand, $H(Y^* \mid \hat{Y}) = H(Y^*) - I(Y^*;\hat{Y}) \ge H(Y^*) - H(\hat{Y}) \ge n - (K+1)\log_2|\Sigma|$. Combining, $n - (K+1)\log_2|\Sigma| \le 1 + nP_e$, i.e., $P_e \ge 1 - \big((K+1)\log_2|\Sigma| + 1\big)/n$. Solving for K at $P_e \le \delta$ gives $K \ge \big((1-\delta)n - 1\big)/\log_2|\Sigma| - 1$. In fixed-seed mode the transcript is a deterministic function of the random input. In random-seed mode its law additionally averages over the seed. The entropy argument applies to both joint laws. □

D.2 Errors without an injected channel

To relate this separation to ordinary generation, fix an instance I and let A_j be the event that the decoded answer is valid and its j-th row bit is correct. Write $q_j = \Pr[A_j \mid A_1, \dots, A_{j-1}]$. Proposition D.3 gives

$$\Pr[\text{exact answer}] = \exp\left(-\sum_{j=1}^{n} \lambda_j\right), \qquad \lambda_j = -\log q_j,$$

with the usual zero-probability conventions. A positive lower bound on mean conditional log-loss yields an exponential success upper bound, but establishing that premise for a learned model is a separate question. The quantities are conditional on correct earlier decoded bits. Unconditional per-row accuracy cannot generally replace them when errors are dependent.

D.3 Exact log-loss form of answer accuracy

Proposition D.3 (Exact log-loss form, assumption-light)**.** *Fix I and the transcript law under either seed convention. For $1 \le j \le n$, let $A_j = \{g(W) \in \{0,1\}^n,\ [g(W)]_j = \mathrm{NLF}(I)_j\}$ be the event that the decoded answer is valid and its j-th row bit is correct. Set $\lambda_j = -\ln \Pr[A_j \mid A_1, \dots, A_{j-1}]$ whenever the conditioning event has positive probability, and set $\lambda_j = +\infty$ otherwise, with $-\ln 0 = +\infty$. All CoT tokens are marginalized. Then the following identity is exact:*

$$\Pr\big[\textit{all decoded row bits correct}\big] \;=\; \exp\Big(-\sum_{j=1}^{n} \lambda_j\Big).$$

Consequently, if the model's per-row conditional log-loss *satisfies $\frac{1}{n}\sum_j \lambda_j \ge \lambda$ for a constant $\lambda > 0$, then $\Pr[\textit{correct}] \le e^{-\lambda n}$. Conversely, $\Pr[\textit{correct}] \ge 1 - \gamma$ for $0 \le \gamma < 1$ forces $\frac{1}{n}\sum_j \lambda_j \le \frac{1}{n}\ln\frac{1}{1-\gamma}$, i.e., vanishing mean conditional log-loss and error. A mean conditional per-row error bounded away from 0 is a sufficient condition for the exponential upper bound.*

These events are well-defined for any extraction map, including one that decodes several row bits from a token. Their intersection is exactly the event of a correct full answer.

Proof. If a conditioning event has probability 0, both the full success probability and $\exp(-\sum_j \lambda_j)$ are 0 by convention. Otherwise, the chain rule gives $\Pr[\bigcap_j A_j] = \prod_j \Pr[A_j \mid A_{<j}] = \prod_j e^{-\lambda_j} = \exp(-\sum_j \lambda_j)$. This includes point-mass transcript laws in fixed-seed mode. The full transcript is marginalized inside each conditional probability, subject only to the decoded-row correctness events. The two consequences follow directly by taking logarithms when success is positive. For conditional error $e_j = 1 - \Pr[A_j \mid A_{<j}]$ on a positive-probability prefix, $\lambda_j = -\ln(1 - e_j) \geq e_j$, so a mean conditional error $\geq \bar{e}$ implies $\Pr[\text{correct}] \leq e^{-\bar{e}n}$. If a prefix has probability 0, the success bound is already trivial. □

Proposition D.3 converts an *empirically measurable* quantity, mean conditional log-loss along correct answer prefixes, into an exact accuracy expression. Accuracy on n-row instances has an exponential upper bound precisely when this mean log-loss stays bounded below by a positive constant. A nonvanishing mean conditional error is sufficient. The converse requires additional control of the individual conditional errors.

E Noise Separation and Readout Extensions

The ordinary and redundant readout bounds share one conditional per-slot estimate and one argument over predictable commitment times. We then prove the PoT bounds and the external-aggregation construction.

E.1 Readout conventions and the common per-slot bound

Formal direct-readout model. For each answer length n in bits, fix the number $A = A(n)$ of answer tokens and positive block sizes $\beta_1, \ldots, \beta_A$ satisfying $\sum_{i=1}^{A} \beta_i = n$. Block i is decoded by a bijection $\varphi_i : S_i \to \{0,1\}^{\beta_i}$ with $S_i \subseteq \Sigma$. Each answer token is a *commitment*, and its designated position is a *slot*. The commitment times $\tau_i \in \mathbb{N}_{\geq 1} \cup \{\infty\}$ are ordered, strictly increasing whenever the later time is finite. Each time is *predictable*: whether $\tau_i = t$ is determined by $(x, \rho, R_{<t})$ before R_t is observed. The readout returns

$$\hat{b} = \varphi_1(R_{\tau_1}) \cdots \varphi_A(R_{\tau_A}).$$

An unreached slot or a token outside its coding set S_i returns $\perp$. This formalizes Definition 4.2. Since $2^{\beta_i} = |S_i| \leq |\Sigma|$, summing $\beta_i \leq \log_2 |\Sigma|$ over the A blocks gives $n \leq A \log_2 |\Sigma|$.

Definition E.1 (r-redundant readout)**.** Fix $r \geq 1$. An *r-redundant* readout system is defined exactly as in Definition 4.2, except that each per-slot decoder is a surjection $\varphi_i : S_i \to \{0,1\}^{\beta_i}$ whose fibers have size at most r: $\max_y |\varphi_i^{-1}(y)| \leq r$. Bijective readout is the case $r = 1$.

Formal setup. For the per-instance bounds, fix $I = (T, Q)$ with n rows. Realize Definition 4.1 using mutually independent variables $U_t \sim \text{Bernoulli}(\varepsilon)$ and V_t uniform on $[|\Sigma| - 1]$, all independent of ρ. The intended token is $W_t = f_t(I, \rho, R_{<t})$. The realized token is W_t if $U_t = 0$, and otherwise the V_t-th symbol of $\Sigma \setminus \{W_t\}$ in a fixed order. With $\mathcal{F}_t = \sigma(\rho, R_1, \ldots, R_t)$, the token W_t is $\mathcal{F}_{t-1}$-measurable and (U_t, V_t) is independent of $\mathcal{F}_{t-1}$. In fixed-seed mode ρ is constant. In random-seed mode it is included in $\mathcal{F}_0$. Thus the same conditional estimate applies under both conventions, with fresh channel randomness in each case.

Lemma E.2 (Per-slot success bound)**.** *Let φ map a coding subset $S \subseteq \Sigma$ surjectively onto $\{0,1\}^{\beta}$ with all fibers of size at most r (i.e. $\max_y |\varphi^{-1}(y)| \leq r$), extend φ by $\perp$ off S, let $y \in \{0,1\}^{\beta}$ be any target block, and let $\varepsilon \in (0, \frac{1}{2}]$, $|\Sigma| \geq 2$, $1 \leq r < |\Sigma|$. Then for every t,*

$$\Pr\big[\, \varphi(R_t) = y \;\big|\; \mathcal{F}_{t-1} \,\big] \;\leq\; q_r \;:=\; (1 - \varepsilon) + \varepsilon \, \frac{r - 1}{|\Sigma| - 1} \;<\; 1 \qquad \textit{almost surely.}$$

Proof. Condition on $\mathcal{F}_{t-1}$, so that W_t is determined, and write $F = \varphi^{-1}(y)$, $|F| \leq r$.

Case 1: $\varphi(W_t) = y$, i.e. $W_t \in F$. The slot decodes to y if the token escapes corruption (probability $1 - \varepsilon$) or if it is corrupted into one of the remaining $|F| - 1 \leq r - 1$ elements of F, each realized with probability $\varepsilon/(|\Sigma| - 1)$. Hence the conditional probability is exactly $(1 - \varepsilon) + \varepsilon \frac{|F| - 1}{|\Sigma| - 1} \leq q_r$.

Case 2: $\varphi(W_t) \neq y$, i.e. $W_t \notin F$. The slot decodes to y only if the token is corrupted (probability ε) into one of the at most r elements of F, giving conditional probability at most $\varepsilon \frac{r}{|\Sigma|-1}$. This is at most q_r, since

$$q_r - \varepsilon \frac{r}{|\Sigma|-1} \;=\; (1-\varepsilon) - \frac{\varepsilon}{|\Sigma|-1} \;\geq\; (1-\varepsilon) - \varepsilon \;\geq\; 0$$

using $|\Sigma| \geq 2$ and $\varepsilon \leq \frac{1}{2}$.

In both cases the bound is q_r, and $q_r < 1$ precisely because $r < |\Sigma|$ forces $\varepsilon \frac{r-1}{|\Sigma|-1} < \varepsilon$. Independence $(U_t, V_t) \perp \mathcal{F}_{t-1}$ justifies the conditional computation. For $r = 1$, this gives the bijective-readout bound $q_1 = 1 - \varepsilon$. □

E.2 Direct and redundant readout bounds

Theorem E.3 (Separation under redundant readout, unconditional, per-instance)**.** *Fix $\varepsilon \in (0, \frac{1}{2}]$, $|\Sigma| \geq 2$, and $1 \leq r < |\Sigma|$. Under $\mathcal{N}_\varepsilon$, with either internal-seed convention, every r-redundant readout system, on every* NLF *instance with n rows, outputs the exact answer with probability at most*

$$q_r^{\,A} \leq q_r^{\,n/\log_2|\Sigma|}, \qquad \textit{where} \qquad q_r \;=\; (1-\varepsilon) \;+\; \varepsilon\,\frac{r-1}{|\Sigma|-1} \;<\; 1.$$

In particular the success probability still decays exponentially in n, at rate $\ln(1/q_r)/\log_2|\Sigma|$ per row, and part (b) of Theorem 4.4 is unchanged, so the separation persists for every fixed $r < |\Sigma|$.

Proof of Theorem 4.4(a) and Theorem E.3. We prove the bound for r-redundant readout. Direct readout is the case $r = 1$. First prescribe the internal seed ρ, so all expectations below are over channel draws. Let $y_i \in \{0,1\}^{\beta_i}$ be the ith block of the fixed answer NLF(I) and put

$$E_i = \{\tau_i < \infty \text{ and } \varphi_i(R_{\tau_i}) = y_i\}.$$

Exact correctness is the event $\bigcap_{i \leq A} E_i$. A missing slot produces a malformed answer. Decomposing over the last commitment time gives

$$\mathbb{E}\Big[\prod_{i \leq A} \mathbf{1}_{E_i}\Big] = \sum_t \mathbb{E}\Big[\mathbf{1}\{\tau_A = t\} \prod_{i<A} \mathbf{1}_{E_i} \cdot \mathbf{1}\{\varphi_A(R_t) = y_A\}\Big].$$

Predictability gives $\{\tau_A = t\} \in \mathcal{F}_{t-1}$. On this event all earlier slots occur before t, so the product of the first two factors is $\mathcal{F}_{t-1}$-measurable. Conditioning each summand on $\mathcal{F}_{t-1}$ and applying Lemma E.2 yields

$$\Pr\Big[\bigcap_{i \leq A} E_i\Big] \leq q_r \,\Pr\Big[\bigcap_{i<A} E_i\Big].$$

Induction therefore gives $\Pr[\bigcap_{i \leq A} E_i] \leq q_r^A$. Surjectivity onto $\{0,1\}^{\beta_i}$ forces $2^{\beta_i} \leq |S_i| \leq |\Sigma|$. Since $\sum_i \beta_i = n$, we have $A \geq n/\log_2|\Sigma|$, and hence success is at most $q_r^{\,n/\log_2|\Sigma|}$. The bound is independent of the prescribed internal seed, so averaging over ρ gives the same result in random-seed mode. Taking $r = 1$ gives Theorem 4.4(a). For every fixed $r < |\Sigma|$, $q_r < 1$ gives the exponential decay in Theorem E.3. The PoT construction in part (b), proved below, has no per-slot decoder and applies for every r. □

Dependence on redundancy. For a fixed number A of answer slots, the ratio of the redundant-readout upper bound to the bijective one is

$$\frac{q_r^A}{(1-\varepsilon)^A} = \left(1 + \frac{\varepsilon(r-1)}{(1-\varepsilon)(|\Sigma|-1)}\right)^A.$$

Thus redundancy changes the exponential rate, while every fixed $r < |\Sigma|$ still gives $q_r < 1$.

E.3 PROGRAM-OF-THOUGHT BOUNDS

Interpreter convention. Token strings serve directly as programs. One concrete implementation uses two fixed vocabulary symbols for binary digits, rejects other symbols, and simulates a fixed deterministic universal Turing machine with polynomial overhead. This absorbs binary encoding and parsing into $\mathcal{U}$. Their cost is included in the execution budget.

Proof of Theorem 4.4(b). Choose a fixed program P_0 implementing a polynomial-time NLF algorithm and a polynomial execution budget p for $\mathcal{U}$. Such a budget exists because $\mathcal{U}$ has polynomial simulation overhead. Set $L = L_0 = |P_0|$ and intend the t-th token of P_0 at step $t \le L_0$, regardless of the internal seed or realized prefix. Thus the construction has the same channel success probability under either seed convention. Then

$$\Pr[R_{1:L_0} = P_0] = \Pr\left[\bigcap_{t \le L_0} \{U_t = 0\}\right] = (1-\varepsilon)^{L_0}.$$

On this event, the realized program is P_0 and the interpreter halts with $\mathrm{NLF}(I)$ within budget, so Definition 4.3 returns the exact answer. Corrupted transcripts may also succeed. Hence success is at least $(1-\varepsilon)^{L_0} = p_0 > 0$ on every instance. □

From a fixed evaluator to generated code. The program P_0 is a generic evaluator: the table and query are supplied as inputs rather than copied into its source. In practice, a model may instead emit query-specific code. If a correct intended program has length at most $c_0 + c_1|Q|$, Corollary E.4 gives the lower bound

$$\Pr[\text{exact answer}] \ge (1-\varepsilon)^{c_0 + c_1|Q|}.$$

This remains constant for bounded encoded query length and is polynomially small when $|Q| = \Theta(\log n)$. Bounded query depth alone does not ensure bounded encoded length, because constants and column indices can grow. Thus extending the construction to learned PoT requires measuring both intended-program correctness and program length as the workload changes. The positive constant in Theorem 4.4(b) need not be large: long programs or high substitution rates can make it small even when it is independent of table size.

Corollary E.4 (Query-inlining variant)**.** *There exists a query-specific PoT construction that intends a correct program P_Q of length at most $c_0 + c_1|Q|$ and executes it within one polynomial budget independent of the instance. Under $\mathcal{N}_\varepsilon$, its exact-answer success is at least $(1-\varepsilon)^{c_0+c_1|Q|}$ under either internal-seed convention. For fixed $\varepsilon > 0$ and vocabulary, this lower bound is constant for queries of bounded encoded length and is $n^{-O(\varepsilon)}$ when $|Q| = \Theta(\log n)$, superpolynomially larger than the $e^{-\Omega(\varepsilon n)}$ ceiling of Theorem 4.4(a).*

Proof of Corollary E.4. Construct P_Q by embedding a self-delimiting encoding of Q as data in a fixed evaluator that applies it to the clean table. A fixed encoding gives $L = |P_Q| \le c_0 + c_1|Q|$ for constants c_0, c_1. Evaluation is polynomial in the encoded input length, so the interpreter's polynomial simulation overhead supplies one polynomial execution budget for all instances. Choose L before generation and intend the tokens of P_Q regardless of the internal seed or realized prefix. The event of no substitution has probability $(1-\varepsilon)^L$ and guarantees the correct answer within budget. Since $L \le c_0 + c_1|Q|$ and $0 < 1-\varepsilon < 1$, success is at least $(1-\varepsilon)^{c_0+c_1|Q|}$. For $|Q| = \Theta(\log n)$, the exponent is $O(\log n)$. The inequalities $\varepsilon \le -\ln(1-\varepsilon) \le 2\varepsilon$ for $0 < \varepsilon \le 1/2$ give the claimed $n^{-O(\varepsilon)}$ lower bound. For fixed positive ε, its ratio to $e^{-\Omega(\varepsilon n)}$ grows faster than every polynomial in n. □

E.4 EXTERNAL AGGREGATION AND THE READOUT BOUNDARY

For example, a parser can wait until it sees the first `ANSWER:` marker and designate the next n binary tokens as the answer. Before each such token arrives, its role is already determined by the observed prefix. A parser that chooses the last marker after reading the entire response has a different power: later text can change which earlier candidate is accepted. The theorem covers the former commitment rule, even when the generator reasons adaptively before committing. It does not cover unrestricted retrospective selection or error-correcting aggregation.

The direct-readout restriction is essential. For $\varepsilon < 1/2$, repeating each intended answer bit $O(\log n)$ times and applying an external majority decoder achieves success at least $1 - 1/(2n)$ with $O(n \log n)$ generated tokens (Proposition E.5). This establishes a sufficient escape through aggregation. It does not characterize all robust protocols. Similarly, a last-marker parser need not have predictable commitments. Our benchmark parser audit found that selecting the first marker changed answers on 1.44% of 26,873 generations, which motivates the model without proving that production parsers satisfy it.

Finally, independent symmetric substitution is an explicit modeling assumption. Positive-temperature sampling alone need not imply a fixed error floor. Appendix D studies endogenous errors separately using conditional log-loss and witness-faithfulness assumptions. Operationally, the PoT guarantee requires returning execution output directly: making a model transcribe the answer exposes those answer tokens to the channel again.

Proposition E.5 (The interface hypothesis is necessary)**.** *Fix $\varepsilon \in (0, \frac{1}{2})$. There is a redundant-encoding CoT protocol whose generator ignores the internal seed (each answer bit emitted $k = O(\log n/(\frac{1}{2} - \varepsilon)^2)$ times, recomputed from the clean input) together with a* deterministic external per-slot majority decoder *whose success under $\mathcal{N}_\varepsilon$ is $\geq 1 - \frac{1}{2n} \to 1$, with total generated length $O\big((\frac{1}{2} - \varepsilon)^{-2}\, n \log n\big)$.*

Proof of Proposition E.5. Let $\delta = \frac{1}{2} - \varepsilon > 0$ and assume, WLOG, one bit per block with designated symbols s_0, s_1. The protocol emits, for each block $i \in [A]$ ($A \leq n$), $k = \lceil \ln(2n^2)/(2\delta^2) \rceil$ consecutive tokens whose intended value is s_{b_i}, recomputed from the clean input each time (the policy ignores the internal seed and realized context, so it works under either seed convention). Conditionally on the input, the events $\{R = s_{b_i}\}$ across the k copies are i.i.d. with probability exactly $1 - \varepsilon \geq \frac{1}{2} + \delta$ (a corrupted copy never equals the intended symbol). The external decoder takes, per block, the majority vote of $\varphi(R)$ over the k realized tokens (non-designated symbols count as abstentions, which cannot outvote a strict majority of correct copies). By Hoeffding, $\Pr[\text{at most } k/2 \text{ correct copies}] \leq e^{-2k\delta^2} \leq \frac{1}{2n^2}$. A union bound over $A \leq n$ blocks gives success $\geq 1 - \frac{1}{2n}$. Total generated length is $Ak = O(\delta^{-2} n \log n)$. The decoder is a deterministic external device: its computation is not autoregressive sampling and does not pass $\mathcal{N}_\varepsilon$. (This protocol is, of course, *not* direct-readout: each bit is decoded by aggregation over k positions, which is exactly the clause of Definition 4.2 it violates, consistent with Theorem 4.4(a).) Related interactive-coding results are discussed by Schulman (1996). The guarantee used here is the repetition construction above. □

F Additional Experimental Details

The synthetic sweep uses 20 instances at each of five row counts. Transcripts are generated once at temperature zero and reused across post-generation character-noise levels. Table 5 records the full grid. The character channel is a serialization-level experiment and does not reproduce realized-token feedback during model generation.

		exact match at row count n				
ε	mode	50	100	200	400	800
0	TCoT	0.300	0.050	0.000	0.000	0.000
0	PoT	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
0.005	TCoT	0.222	0.038	0.000	0.000	0.000
0.005	PoT	0.275	0.269	0.212	0.256	0.281
0.01	TCoT	0.182	0.017	0.000	0.000	0.000
0.01	PoT	0.081	0.087	0.069	0.075	0.087
0.02	TCoT	0.105	0.007	0.000	0.000	0.000
0.02	PoT	0.000	0.000	0.006	0.006	0.013

Table 5: Full synthetic exact-match grid under post-generation character substitutions.

Slope and first-error estimators. For each ε, the reported slope $\hat{s}(\varepsilon)$ is the ordinary-least-squares coefficient of n in $\log \mathrm{EM}(n, \varepsilon)$, retaining only row counts with positive exact match. The first-error hazard estimate divides observed first errors by row-slots reached with a correct prefix. The reported count is $37/1053 = 0.0351$. The corresponding small-hazard log-loss approximation is $-\log(1 - 0.0351) \approx 0.0357$. This is numerically close to $|\hat{s}(0)| = 0.0358$, but an empirical hazard is not identical to the per-instance conditional log-loss in Proposition D.3. Aggregating over queries and row positions requires additional homogeneity assumptions for a common-rate interpretation.

quantity	TCoT	PoT	prediction tested
fitted $d(\ln \mathrm{acc})/dn$ at $\varepsilon = 0$	-0.0358	0.0000	scaling
conditional per-row hazard $\hat{\lambda}$ (first-error estimate)	0.0351	n/a	small-hazard comparison
slope at $\varepsilon = 0.01$ (predicted $\hat{s}(0) + \ln(1-\varepsilon) = -0.046$)	-0.0478	$+0.0001$	slope comparison
slope at $\varepsilon = 0.02$ (predicted -0.056)	-0.0551	$+0.0012$	additivity
correct-program rate (synthetic NLF)	n/a	1.000	program correctness

Table 6: Reported slope and hazard diagnostics for the synthetic sweep.

Only $n = 50, 100$ have positive TCoT exact match in the reported grid. Thus its fitted lines have two points. Zero observations at larger sizes cannot establish a population success probability of zero. The comparisons $\hat{s}(\varepsilon) \approx \hat{s}(0) + \log(1 - \varepsilon)$ are descriptive, and substitutions can in principle repair an initially incorrect symbol. These diagnostics do not prove an additive error model or an asymptotic scaling law for the learned generator.

Benchmark comparability. Public-benchmark PoT uses up to eight execution-feedback rounds, whereas synthetic PoT is single-shot. Successful execution does not imply a correct answer. The context cap is 50K characters, and both modes receive the same selected instances. Selection by table size or answer type does not hold query difficulty fixed. Strict and judge scores measure different notions of correctness. In particular, the DataBench numeric-list comparison reverses under strict scoring, so the judge result cannot be treated as an exact-match improvement.

Scope of the reported protocol. The available experimental record specifies a DeepSeek backbone, prompts for textual answers or pandas programs, and Monte-Carlo channel resampling. It does not identify the precise model checkpoint, channel-resampling count, or averaging procedure underlying every reported benchmark score. We preserve the reported measurements without reconstructing unavailable runs or attaching confidence intervals unsupported by the record. Releasing those details and the underlying generations is necessary for independent reproduction.

G Discussion, Limitations, and Open Problems

The computational and reliability results concern different resources. Encoded input size N controls the conditional short-CoT barrier, while answer length n controls the direct-readout noise bound. Polynomial CoT can compute NLF exactly, so a failure observed at long reasoning lengths cannot be explained solely by the short-CoT circuit upper bound. Under the stated noise channel, however, exact-answer delivery remains constrained by how the answer is read out.

The separation permits unrestricted generators and therefore attributes no failure specifically to attention, depth, or training. Its PoT side assumes clean access to the table and query, reliable execution, and an intended correct program. Practical code generation can fail before execution, and alternative decoders can protect textual answers. The empirical evidence is limited to one model family, small strata, and post-generation character corruption. The public benchmarks also use a repair loop. The judge shares a provider with the evaluated model. These comparisons are descriptive and do not establish statistical significance or universal exponential scaling in deployment.

The most useful next steps are an end-to-end analysis of correct-program generation, experiments with feedback-coupled noise, and controlled comparisons of execution against redundant textual answers with external decoding. A general characterization of robust readout protocols must specify stopping and verification conventions. The majority construction proves sufficiency, with broader necessity unresolved. The open problems below state these and the remaining complexity questions.

The bounds concern exposure during generation. Execution and external decoding consume additional resources, so comparisons should report generated length, executed work, repair attempts, and accuracy together.

G.1 Open problems

Open Problem G.1. Prove $\mathsf{NC}^1 \not\subseteq \mathsf{TC}^0/\mathrm{poly}$ (Assumption 2.2), a central open problem of circuit complexity, or the average-case hypothesis of Proposition C.3.

Open Problem G.2. Extend architecture-specific CoT-length lower bounds (Amiri et al., 2025) from hard to soft attention, and beyond output entropy for multi-bit outputs.

Open Problem G.3. Prove an end-to-end PoT reliability theorem: correct-program probability as a function of query complexity, composed with verifier-guided resampling.

Open Problem G.4. Under explicitly specified termination, budget, and verification conventions, prove or refute that every $\mathcal{N}_\varepsilon$-robust exact-answer protocol for NLF factors through a deterministic aggregation or verification component external to token sampling (Proposition E.5 proves sufficiency, while necessity is open).

The complementary question of endogenous error floors is Open Problem **??**.