

## A Separation between exponential and sequential scaling based on expressivity

We prove Theorem 3, which is the formal version of the Theorem 1 stated in the main text.

### A.1 Preliminaries: expressivity of transformers

Before stating the theorem formally and proving it, let us first review known technical bounds on the expressivity of limited-precision, bounded-depth transformers [56, 36, 8]. To present these, we must first define the computational model of threshold circuits.

**Definition 5** ( $\text{TC}^0$  computational model). *A  $\text{TC}^0$  circuit is a boolean circuit with AND, OR, NOT, and MAJORITY gates of potentially unbounded fan-in. A  $\text{TC}^0$  circuit family is a collection of circuits indexed by the input size  $n$ , such that for each input size the circuit has polynomial width and bounded depth.*

It has recently been shown that constant-depth transformers can be well-approximated by the class of threshold circuits of constant depth.

**Proposition 1** (Transformers are in  $\text{TC}^0$ ; implied by Theorem 14 of [8]). *For any bounded-depth softmax-attention transformer  $T : \Sigma^* \rightarrow \mathbb{R}^{|\Sigma|}$  and any polynomial  $p(n)$ , there is a function  $\hat{T} : \Sigma^* \rightarrow \mathbb{R}^{|\Sigma|}$  in  $\text{TC}^0$  that approximates  $T$  to  $2^{-p(n)}$  additive error on inputs of length  $n$ .<sup>1</sup>*

This implies limitations on the expressive power of transformers, under standard computational complexity assumptions. In particular, it is a common conjecture that  $\text{TC}^0$  circuits are unable to determine  $s$ - $t$  connectivity in undirected graphs [3, 65], and this conjecture is normally stated as  $\text{L} \not\subseteq \text{TC}^0$ <sup>2</sup>, because  $\text{L}$  is a complete problem undirected graph connectivity [44, 47]. Therefore, Proposition 1 provides evidence that bounded-depth and poly-size transformers (without chain of thought) are not able to directly determine whether two nodes are connected in an inputted graph.

### A.2 Our result

Proposition 1 has not been shown to imply a tradeoff between parallel and sequential scaling in transformers, which is the new contribution in Theorem 3 proved in this section.

Given a function  $T : \Sigma^* \rightarrow \mathbb{R}^{|\Sigma|}$  operating on a polynomial-size alphabet of tokens  $\Sigma$ , and an input prompt  $x \in \Sigma^k$ , we inductively define the autoregressive distribution

$$D_{T,n}(x)$$

formed by sampling  $n$  tokens autoregressively from the transformer.  $D_{T,0}$  is the empty string with probability 1. For any  $n \geq 1$ , the distribution  $D_{T,n}$  is the distribution of  $[z_1, \dots, z_n]$  where  $[z_1, \dots, z_{n-1}] \sim D_{T,n-1}$ , and  $z_n \sim \text{softmax}(T([x; z_1, \dots, z_{n-1}]))$ .

We first prove that the distribution of outputs from a transformer is close in total variation to one generated by iteratively applying a  $\text{TC}^0$  circuit.

**Lemma 1** (Approximating the autoregressive distribution of a transformer). *Given a transformer  $T : \Sigma^* \rightarrow \mathbb{R}^{|\Sigma|}$  and polynomials  $p_1(n), p_2(n)$ , there is a function  $\hat{T}$  in  $\text{TC}^0$  such that for all  $x \in \Sigma^n$*

$$d_{TV}(D_{T,m}(x); D_{\hat{T},m}) \leq 2^{-p_1(n)},$$

for any  $m \leq p_2(n)$ , where  $d_{TV}$  denotes the total variation distance between distributions.

*Proof.* Let  $p(n)$  be a polynomial that we will fix later. Let  $\hat{T}$  be a  $\text{TC}^0$  circuit family such that  $\hat{T}$  approximates  $T$  up to  $2^{-p(n)}$  additive error on inputs of length  $n$ , as guaranteed by Proposition 1. For

<sup>1</sup>The  $\text{TC}^0$  circuit outputs in  $\mathbb{R}^{|\Sigma|}$  is returned up to some number of bits of precision.

<sup>2</sup>For directed graphs, which we will not use here, the relevant conjecture is  $\text{NL} \not\subseteq \text{TC}^0$

887  $m = 0$ , we have  $d_{TV}(D_{T,0}(x), D_{\hat{T},0}(x)) = 0$  by definition. For any string  $s$  of length  $\geq n$ , we have

$$\begin{aligned} d_{TV}(D_{T,1}(s), D_{\hat{T},1}(s)) &= \frac{1}{2} \sum_{i \in \Sigma} \left| \frac{\exp(T(s)_i)}{\sum_{j \in \Sigma} \exp(T(s)_j)} - \frac{\exp(\hat{T}(s)_i)}{\sum_{j \in \Sigma} \exp(\hat{T}(s)_j)} \right| \\ &\leq |\exp((|\Sigma| + 1)2^{-p(n)}) - \exp(-(|\Sigma| + 1)2^{-p(n)})| \\ &\leq 5(|\Sigma| + 1)2^{-p(n)}, \end{aligned}$$

888 whenever  $n$  is large enough and  $2^{-p(n)}(|\Sigma| + 1) \leq 1$ . So combining with the data-processing  
889 inequality, for any  $m \geq 1$ , we have

$$\begin{aligned} d_{TV}(D_{T,m}(x), D_{\hat{T},m}(x)) &\leq d_{TV}(D_{T,m-1}(x), D_{\hat{T},m-1}(x)) + \mathbb{E}_{z \sim D_{T,m-1}(x)} [d_{TV}(D_{T,1}([x; z]), D_{\hat{T},1}([x; z]))] \\ &\leq d_{TV}(D_{T,m-1}(x), D_{\hat{T},m-1}(x)) + 5(|\Sigma| + 1)2^{-p(n)}. \end{aligned}$$

890 Applying this inductively on  $m$  yields

$$\begin{aligned} d_{TV}(D_{T,m}(x), D_{\hat{T},m}(x)) &\leq 5m(|\Sigma| + 1)2^{-p(n)} \\ &\leq 5p_1(n)(|\Sigma| + 1)2^{-p(n)}. \end{aligned}$$

891 Choosing  $p(n)$  large enough so that the right-hand side is  $\leq 2^{-p_1(n)}$  concludes the proof.  $\square$

892 This allows us to consider the autoregressive distributions generated by  $\text{TC}^0$  circuits, which we will  
893 find easier to analyze than the autoregressive distributions generated by transformers.

894 We observe that, for constant-length chains of thought, the autoregressive distribution is also directly  
895 sampleable by a  $\text{TC}^0$  circuit with no chain of thought. This lemma was effectively claimed in Figure  
896 1 of [35], but without a proof.

897 **Lemma 2** (Constant-length CoT simulated by randomized  $\text{TC}^0$ ). *Let  $C$  be a constant number of*  
898 *steps, and let  $\hat{T} : \Sigma^* \rightarrow \mathbb{R}^{|\Sigma|}$  be a function in  $\text{TC}^0$ . Define the distribution of the last token  $\hat{P}(x)$  to*  
899 *be the law of  $z_C$  where  $z \sim D_{\hat{T},C}(x)$ .*

900 *Then for any polynomial  $p_1(n)$ , there is a polynomial  $p_2(n)$  and a function  $\tilde{T} : (\Sigma \cup \{0, 1\})^* \rightarrow \Sigma$*   
901 *in  $\text{TC}^0$  such that for all  $x \in \Sigma^n$  we have*

$$d_{TV}(\hat{P}(x); \tilde{P}(x)) \leq 2^{-p_1(n)}$$

902 *where  $\tilde{P}(x)$  is the law of  $\tilde{T}(x; r)$ , where  $r \sim \text{Unif}[\{0, 1\}^{p_2(n)}]$  are random input bits.*

903 *In other words, one step of  $\tilde{T}$  approximates  $C$  autoregressive steps of  $\hat{T}$ .*

904 *Proof.* For any polynomial  $p(n)$ , there is a  $\text{TC}^0$  circuit that (given a polynomial number of random  
905 bits), samples from a step of the autoregressive distribution with  $\hat{T}$  up to total variation error  $2^{-p(n)}$ .  
906 This is because first the circuit can compute  $\hat{T}$ , and then the softmax operation can be approximated  
907 by  $\text{TC}^0$  circuits, as proved in Theorem 14 of [8]. Concatenating this circuit  $C$  times, we obtain a  
908 randomized  $\text{TC}^0$  circuit  $\tilde{T}$  that satisfies the lemma, as long as we take  $p(n) \geq p_1(n) \log_2(1/C)$ .  $\square$

909 Now recall the folklore result that  $\text{TC}^0$  circuits can be derandomized.

910 **Lemma 3** (Derandomization of  $\text{TC}^0$ ; folklore). *Let  $p(n)$  and  $p'(n)$  be polynomials and  $\tilde{T} : (\Sigma \cup$   
911  $\{0, 1\})^* \rightarrow \Sigma$  be a  $\text{TC}^0$  function.*

912 *Then, there is a  $\text{TC}^0$  function  $\dot{T} : \Sigma^* \rightarrow \Sigma$  such that for any  $n$ , any  $x \in \Sigma^n$  and  $\sigma \in \Sigma$ , we have*

$$\dot{T}(x) = \sigma, \text{ if } \mathbb{P}_{r \sim \{0,1\}^{p(n)}} [\tilde{T}(x; r) = \sigma] \geq 1/2 + 1/p'(n).$$

913 *Proof.* Let  $p_1(n)$  be a polynomial that we will fix later. Consider the circuit  $T'$  that upon  
914 input  $[x; r_1, \dots, r_{p_1(n)}]$  where  $x \in \Sigma^n$  and  $r_i \in \{0, 1\}^{p(n)}$ , takes a majority vote over

915  $\tilde{T}(x; r_1), \dots, \tilde{T}(x; r_{p_1(n)})$ . By a Chernoff bound, and a large enough polynomial  $p_1(n)$ , we have  
 916 that for any  $x \in \Sigma^n$  and  $r \in \{0, 1\}^{p(n)}$ , we have

$$\mathbb{P}_{r_1, \dots, r_{p_1(n)}}[T'(x; r_1, \dots, r_{p_1(n)}) = \sigma] \geq 1 - |\Sigma|^{-n-1} \text{ if } \mathbb{P}_{r \sim \{0,1\}^{p(n)}}[\tilde{T}(x; r) = \sigma] \geq 1/2 + 1/p'(n).$$

917 By a union bound over all inputs  $x \in |\Sigma|^n$ , for any  $n$  there is a random seed  $[r_1^*, \dots, r_{p_1(n)}^*]$  such that

$$T'(x; r_1^*, \dots, r_{p_1(n)}^*) = \sigma, \text{ if } \mathbb{P}_{r \sim \{0,1\}^{p(n)}}[\tilde{T}(x; r) = \sigma] \geq 1/2 + 1/p'(n).$$

918 For any  $x \in \Sigma^n$ , let  $\dot{T}(x) = T'(x; r_1^*, \dots, r_{p_1(n)}^*)$ , which is in  $\text{TC}^0$  since the seed can be hardcoded  
 919 into the circuit and is of polynomial length.  $\square$

920 With these preliminaries, we arrive at Theorem 3, which is the formal statement of Theorem 1, which  
 921 was in the main text. We assume that there are two output tokens yes, no  $\in \Sigma$ , and the transformer's  
 922 final token in the chain of thought is its response – either yes or no.

923 **Theorem 3.** *We have the following results for  $(s, t)$ -connectivity problems of size  $n$  and transformers.*

- 924 • **Sequential scaling succeeds:** *There is a constant  $c > 0$  such that a log-precision transformer*  
 925 *with a CoT of length  $\leq n^c$  solves any  $(s, t)$ -connectivity problem.*
- 926 • **Parallel scaling fails:** *Assume that  $\text{L} \not\subseteq \text{TC}^0$ . Let  $C_1, C_2 > 0$  be constants, and let*  
 927  *$T : \Sigma^* \rightarrow \mathbb{R}^{|\Sigma|}$  be a polynomial-precision transformer. Let  $m(n) := n^{C_2}$  be the number of*  
 928 *chains of thought over which we take majority vote (breaking ties arbitrarily). Then there*  
 929 *are infinitely-many  $n$  such that there is a size- $n$   $(s, t)$ -connectivity graph problem  $(G, s, t)$*   
 930 *with answer  $\text{ans} \in \{\text{yes}, \text{no}\}$ , such that*

$$\mathbb{P}_{z_1, \dots, z_{m(n)} \sim D_{T, C_1}(G, s, t)}[\text{Majority}(z_1, C_1, \dots, z_{m(n)}, C_1) \neq \text{ans}] < 1/2 + 1/n.$$

931 *I.e., majority vote over  $m(n)$  parallel chains of thought with length  $C_1$  is correct with*  
 932 *probability at most  $1/2 + o(1)$ .*

933 *Proof.* For the positive result that sequential scaling succeeds, it is sufficient to use Corollary 2.1 of  
 934 [35], which implies that log-precision transformers with  $t(n)$ -length chain of thought can simulate  
 935 Turing machines that run in time  $t(n)$ . Since  $(s, t)$ -connectivity is solvable in polynomial time (e.g.  
 936 with breadth-first search), the first part of the theorem follows.

937 For the negative result that parallel scaling fails, we use the lemmas that we have developed above.  
 938 Suppose by contradiction that for large enough  $n$ , we have for all size- $n$  problems  $(G, s, t, \text{ans})$  that

$$\mathbb{P}_{z_1, \dots, z_{m(n)} \sim D_{T, C_1}(G, s, t)}[\text{Majority}(z_1, C_1, \dots, z_{m(n)}, C_1) = \text{ans}] \geq 1/2 + 1/n.$$

939 Then by Lemma 1 there is a  $\text{TC}^0$  function  $\hat{T}$  such that for all large enough  $n$  and all size- $n$  problems  
 940  $(G, s, t, \text{ans})$ , we have

$$\mathbb{P}_{z_1, \dots, z_{m(n)} \sim D_{\hat{T}, C_1}(G, s, t)}[\text{Majority}(z_1, C_1, \dots, z_{m(n)}, C_1) = \text{ans}] \geq 1/2 + 2/n.$$

941 By Lemma 2, there is a  $\text{TC}^0$  function  $\tilde{T}$  that approximates the autoregressively-applied  $\hat{T}$ , in the  
 942 sense that there is a polynomial  $\tilde{p}$  such that for any size- $n$  problem  $(G, s, t, \text{ans})$

$$\mathbb{P}_{r_1, \dots, r_{m(n)} \sim \{0,1\}^{\tilde{p}(n)}}[\text{Majority}(\tilde{T}(x; r_1), \dots, \tilde{T}(x; r_{m(n)})) = \text{ans}] \geq 1/2 + 3/n.$$

943 Since Majority is gate, the circuit  $\tilde{T}(x; r_1), \dots, \tilde{T}(x; r_{m(n)})$  is a  $\text{TC}^0$  function and so it can be  
 944 derandomized by Lemma 3. Using this lemma, yields a  $\text{TC}^0$  function  $\dot{T}$  such that for any size- $n$   
 945 problem  $(G, s, t, \text{ans})$ ,

$$\dot{T}(G, s, t) = \text{ans}.$$

946 Recall that  $(s, t)$ -connectivity is complete for the class  $\text{L}$  under  $\text{TC}^0$  reductions (see e.g., [3, 65]), and  
 947 we have constructed a  $\text{TC}^0$  circuit for the problem. This implies  $\text{L} \subseteq \text{TC}^0$ , which contradicts our  
 948 assumption that  $\text{L} \not\subseteq \text{TC}^0$ .  $\square$

949

## B Evidence from Vertex Query Model for sequential vs. parallel scaling separation

### B.1 Proof of Theorem 2

*Proof.* First we will show the lower bound.

Notice that due to the structure of the graph, there is a symmetry for  $s, t_1, t_2$  such that it doesn't matter where an algorithm explore from, and there is no advantage to exploring from more than one of them. So, without loss of generality, assume that the model will explore from  $s$ , and stop when it reaches  $t_1$  or  $t_2$  (note that because the vertex labels are uniformly random, there is no other way of getting a higher than 50% success rate than finding  $t_1$  or  $t_2$  when starting from  $s$ ).

To get from  $s$  to  $t_b$ , the algorithm must explore each intersection (those vertices with degree greater than two). To get from the current intersection to the next one, the algorithm has no way to distinguish between the long and short path until it explores at least  $l$  vertices, and so there is at most a  $1/2$  chance the model takes  $l$  oracle calls to get to the next intersection, and at least a  $1/2$  chance it takes  $2l$  oracle calls (if it takes the long path for  $l$  vertices, then any node it has discovered is still  $l$  vertices away from the next intersection, so it must make at least  $l$  more calls). Since there are  $d$  intersections<sup>3</sup>, a standard Chernoff bound for iid Bernoulli random variables shows that the probability of finding  $t_b$  in at most  $(1 - \delta) \frac{3}{2}ld$  oracle calls is at most  $\exp(-\frac{1}{2}\delta^2 \frac{3}{2}d)$ , and if we don't find  $t_b$ , then the best the algorithm can do is guess, and get a  $1/2$  probability of being correct, yielding the desired result.

For the upper bound, we will consider this algorithm: each time we reach a new intersection (including the start), choose an unexplored neighbor, and explore down that path for  $l$  vertices, and if the next intersection is not found, try one of the other unexplored paths from before.

At a new intersection, the algorithm has three unexplored paths:

1. The short path to the next intersection
2. The long path to the next intersection
3. The path to the previous intersection it didn't take

So, notice that the algorithm we defined has a  $1/3$  chance of taking  $l$  oracle calls to reach the next intersection, a  $1/3$  chance of taking  $2l$ , and  $1/3$  chance of taking  $3l$ . Using Hoeffdings inequality, the probability the algorithm takes more than  $(1 + \delta)2ld$  oracle calls is at most  $\exp(-2d\delta^2)$ , so the algorithm succeeds with at least one minus this probability.

□

### B.2 Vertex Query Model with Random Access

Consider the following, less restrictive model than the Vertex Query Model (VQM) in the main text. We call this model the Vertex Query Model with Random Access (VQMRA).

**Definition 6** (Vertex Query Model with Random Access). *An algorithm for  $(s, t_1, t_2)$ -connectivity is in the VQMRA if it takes as input  $s_1, t_1, t_2$ , and can only access the graph  $G$  through “neighborhood queries”  $N_G$ , which given a vertex  $v$  output the set of neighbors  $N_G(v) = \{u : \exists(v, u) \in E\}$ . In the VQMRA, the algorithm is allowed to query any vertex by its identifier, even if it is not in the initial set or has not been visited before.*

In this less restrictive model of computation, we can also prove the necessity of a minimum number of queries (corresponding to a minimum length for a chain of thought by our Ansatz that the VQM models the capabilities of transformers with bounded chain-of-thought).

**Theorem 4** (Minimum number of VQMRA queries needed for graph connectivity). *Consider the graph  $G$  given by two disjoint paths of length  $L \geq 3$  with randomly permuted vertex IDs. Suppose  $s, t_1, t_2$  are distinct endpoints of these paths such that  $s$  and  $t_i$  are on the same path for exactly one  $i \in \{1, 2\}$ . Then*

- $\Omega(L)$  **queries needed:** For any VQMRA algorithm that executes  $q \leq (L - 2)/2$  queries, the probability of correctness of the algorithm on  $(s, t_1, t_2)$ -connectivity is exactly  $1/2$ .

<sup>3</sup>Including  $s$ , for which the same logic applies when getting from  $s$  to the next intersection.

997 •  $O(L)$  **queries sufficient:** There is a VQMRA algorithm that executes  $L - 1$  queries and  
 998 solves the  $(s, t_1, t_2)$ -connectivity problem with probability 1.

999 *Proof.* For the positive result, consider the algorithm that queries  $s$ , then the neighbor of  $s$ , and so on,  
 1000 until it reaches the other end of the path. This takes at most  $L - 1$  queries, and reaches either  $t_1$  or  $t_2$ ,  
 1001 at which point the algorithm has enough information to return the correct answer.

1002 For the analysis of the negative result, let  $u_1, \dots, u_L$  denote the ordered vertices of the first path  
 1003 and let  $v_1, \dots, v_L$  denote the ordered vertices of the second path. Let the algorithm run and make  
 1004  $q \leq (L - 2)/2$  queries. By the pigeonhole principle there must be an  $i \in \{1, \dots, L - 1\}$  such  
 1005 that the algorithm has not queried  $u_i, v_i, u_{i+1}$  and  $v_{i+1}$ . Now note that if we additionally reveal  
 1006 the neighborhoods of  $u_1, \dots, u_{i-1}, u_{i+2}, \dots, u_L$  and  $v_1, \dots, v_{i-1}, v_{i+2}, \dots, v_L$  with vertex queries  
 1007 then the algorithm still has probability of success  $1/2$ , since it is equally likely given its information  
 1008 that  $u_i$  is connected to  $u_{i+1}$  as it is for  $v_i$  to be connected to  $v_{i+1}$ .  $\square$

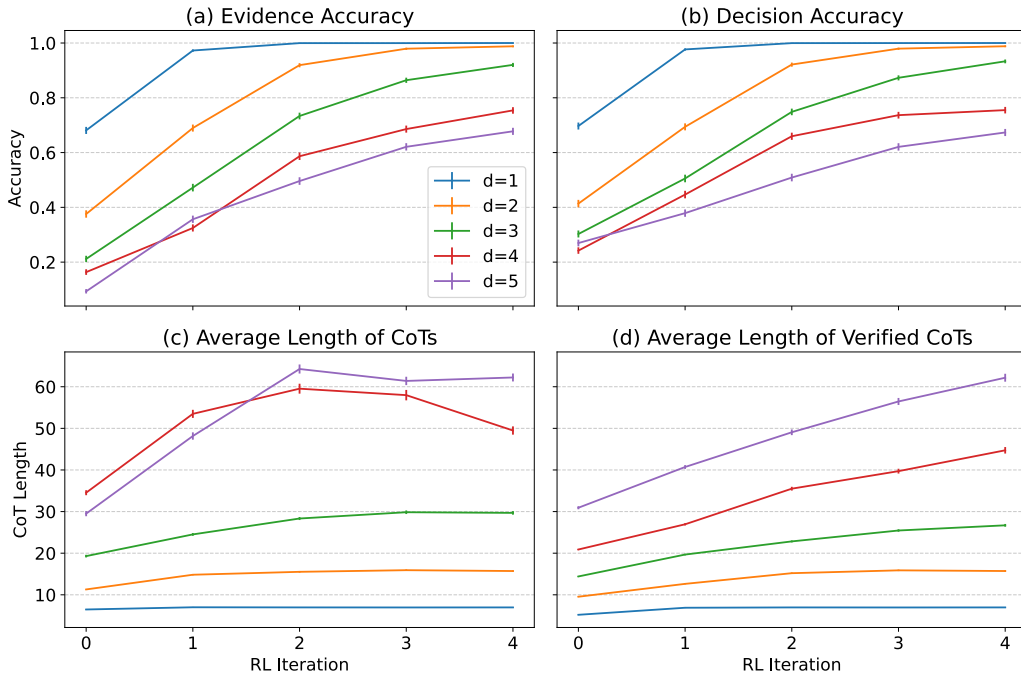


Figure 6: Path model’s (a) Evidence accuracy, (b) Decision accuracy, and average length of (c) CoTs that follow the format, and (d) CoTs that are verified on Bridge tasks of various depths after RL iterations. Error bars represent 95% binomial confidence intervals for accuracies, and 95% normal confidence intervals for CoT lengths.

## 1009 C Experimental Details and Further Experiments

### 1010 C.1 Training

1011 For each CoT strategy and task, we train a Mistral causal language model [17, 66] with 4 hidden  
 1012 layers, 4 attention heads, and intermediate size 128 with a context length of 400 for 200 epochs on  
 1013 NVIDIA A100 GPU with 40GB memory. We sweep through the learning rate values in  $\{1e-4, 3e-4,$   
 1014  $1e-3, 3e-3\}$  and train the model for 200 epochs with a batch size of 1000. We have also experimented  
 1015 with different weight decay values and learning rate schedules, but we found no significant difference  
 1016 in the results and used 0.05 weight decay and a cosine learning rate schedule, with a 0.1 warm-up  
 1017 ratio. We use the same hyperparameters for RL iterations, except that we fine-tune the model for 20  
 1018 epochs at each iteration. Each pretraining experiment takes under 12 GPU hours, while fine-tuning

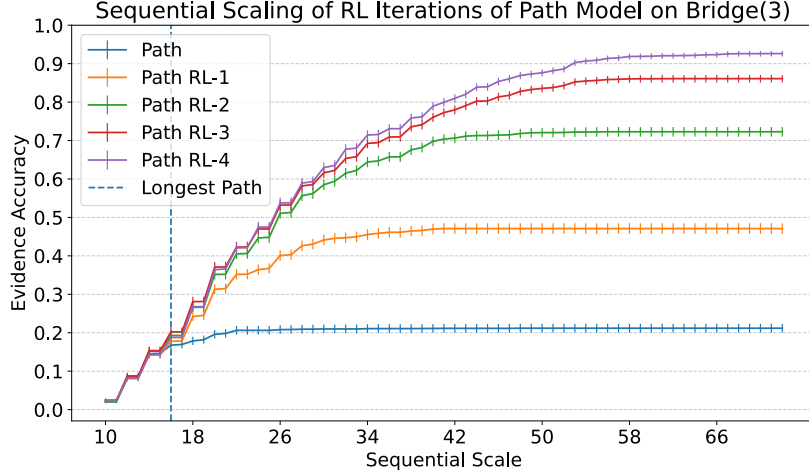


Figure 7: Evidence accuracy of Path model before and after RL iterations with different sequential CoT budgets on the Bridge(3) task. Error bars represent 95% binomial confidence intervals.

for RL takes under 3 GPU hours. Additionally, debugging and hyperparameter tuning for each experiment took under 72 GPU hours.

## C.2 Sequential Scaling of Walk Strategies

**Experiment setup** To study sequential scaling of CoTs in a controlled setting, we also ran experiments with a CoT strategy with tunable scale. A **Walk-L** CoT is generated by sampling a random walk that starts at the source node, conditioned on visiting the target node within at most  $L$  steps. Hence, models trained with Walk-L strategies at different scales  $L$  are exposed to successful traces of random walk on the same task, but with different number of steps the walk is allowed to take to reach the target. As  $L$  increases, the CoTs become longer, less optimal, and more exploratory.

**Results** After training models for the Bridge(5) task with Walk-L CoT strategies, we find that the accuracy of the models consistently increases with  $L$ , which shows that the models trained on more exploratory and longer walks perform better (See Figure 13).

## C.3 Experiment with a Smaller Transformer

**Experiment setup** We also ran experiments using smaller transformer models with 2 hidden layers, and a variant of DFS strategy called **DFS-BT**. In a CoT of DFS-BT strategy, we include the whole DFS trace, which is a walk in the DFS tree including the backtracking steps.

**Results** We find that small models trained on DFS-BT CoTs solve the task consistently, while small models trained on DFS CoTs fail to solve the Bridge tasks of larger depths (See Figure 9), which can be explained by the smaller model’s more limited expressivity.

## C.4 LLM Experimental Details

We used vllm and 2 A100 GPUs (80 GB of memory each) for inference. The experiments to make each plot took less than four hours each. Debugging and hyperparameter tuning took under 120 GPU hours. We constructed 32 random labelings of the bridge graph, and then, using prompts of the form Figure 10, create CoTs of 4096 tokens. Depending the model, we added the appropriate special tokens to make the input prompt from the user, and to make the model use thinking mode during the CoT. Each model recommended using temperature 0.6 for thinking, which we did. We used a custom logit processor to make the model substitute the end thinking token and the eos token with the token for "wait", inspired by [39]. Then we truncate the CoT at intervals evenly spaced by tokens, and append the end of thinking token, and “Answer: Node [start node label] is in the same

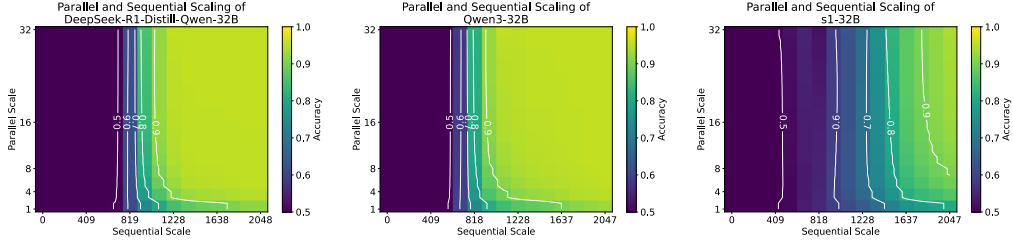


Figure 8: A comparison of parallel and sequential scaling for three LLMs tested on the  $(s, t_1, t_2)$ -connectivity problem for a graph that is the composition of two paths. Note the similar trend to Figure 1.

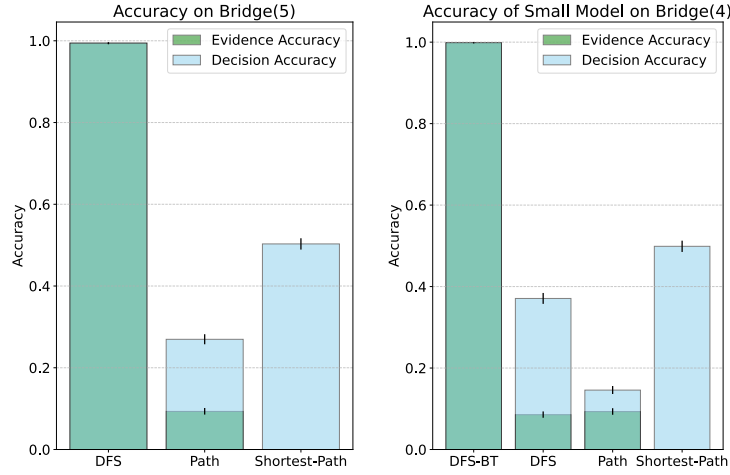


Figure 9: Decision and evidence accuracy of (left) models trained on CoT strategies for Bridge(5) task, and (right) models with 2 hidden layers trained on CoT strategies, including DFS-BT and DFS, for Bridge(5) task. Error bars represent 95% binomial confidence intervals.

connected component as node ” before using the model to find the logits for the next token. The model is considered correct if the logit for the correct node is higher than the logit for the incorrect node introduced in the initial prompt<sup>4</sup>. For parallel scaling, we generated up to 64 distinct CoTs for each graph, and analytically calculated the probability that a random subsample would vote for the correct or incorrect solution (or tie). All of the results have a standard deviation of at most 0.08.

#### C.4.1 LLM Additional Experiments

In Figure 8 also tested the LLMs on a setting closer to the setting of Theorem 4, where the graph to be explored is two disjoint paths, and we once again confirm the theory, and see similar trends to those in Figure 1.

## D Further related work

**Expressivity of Transformers with CoT** The representational power of transformers has been studied in several works [50, 70, 34, 45, 57]. Recent work also highlights the expressivity and sample efficiency gains of reasoning with chain-of-thoughts [64, 21, 9, 33, 35, 29, 41]. In particular, many studies use graph-based tasks as a testbed for studying multi-step reasoning with CoTs [1, 22, 49].

<sup>4</sup>With some tie breaking when the logits are within  $1e-8$  of each other. We found that techniques weighting the confidence by the magnitude of the logits or their difference did not significantly change any results.



### Prompt format

Given the following list of undirected edges in a graph (with nodes labeled 0 through 33), is node 0 in the same component as 10 or as 27? (it is connected to exactly one of the two) Think step by step.  
 (11, 12), (23, 24), (6, 7), (25, 17), (4, 5), (27, 28), (9, 10), (2, 16), (13, 14), (2, 3), (5, 6),  
 (18, 17), (10, 11), (3, 4), (31, 32), (18, 19), (19, 33), (30, 31), (20, 21), (2, 9), (24, 25),  
 (15, 16), (12, 13), (7, 8), (19, 20), (1, 2), (32, 33), (29, 30), (14, 15), (28, 29), (1, 0), (8,  
 0), (21, 22), (19, 26), (22, 23), (26, 27)

Figure 10: Example prompt from the LLM experiments. The prompt includes basic instructions for the task, along with the recommendation to think step by step (to avoid the model responding immediately with a guess, and then spending the rest of the chain of thought trying to justify it).

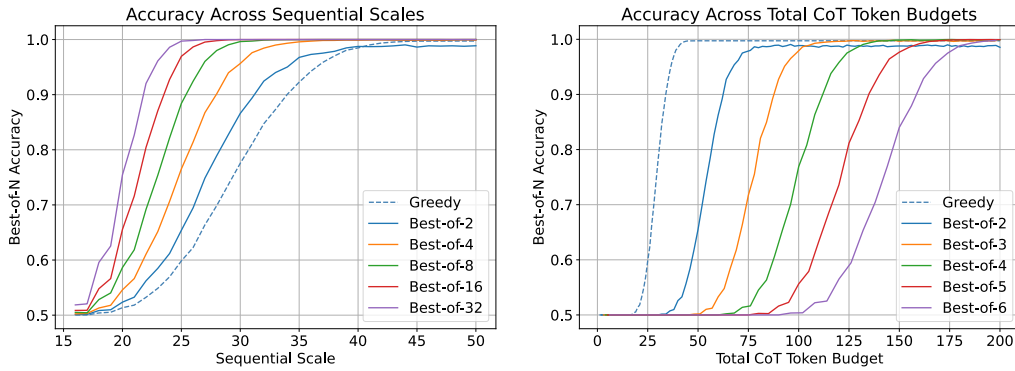


Figure 11: Best-of-N accuracy for parallel scaling of the model trained with DFS CoT strategy on Bridge(5) task (left) across sequential scales (maximum CoT length) and (right) total CoT token budget. Outputs are sampled with temperature 1.0 for parallel scaling.

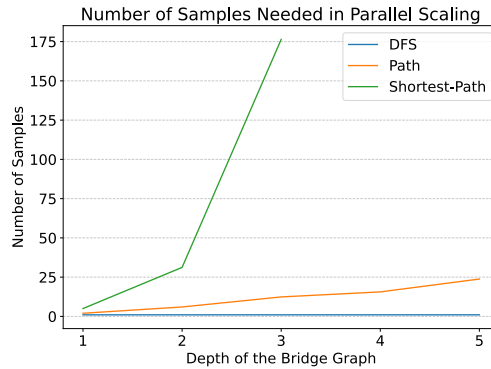


Figure 12: Number of samples to get larger than 95% best-of-N accuracy by parallel scaling models trained with different CoT strategies with temperature 1.0. The average over 5 runs is reported.



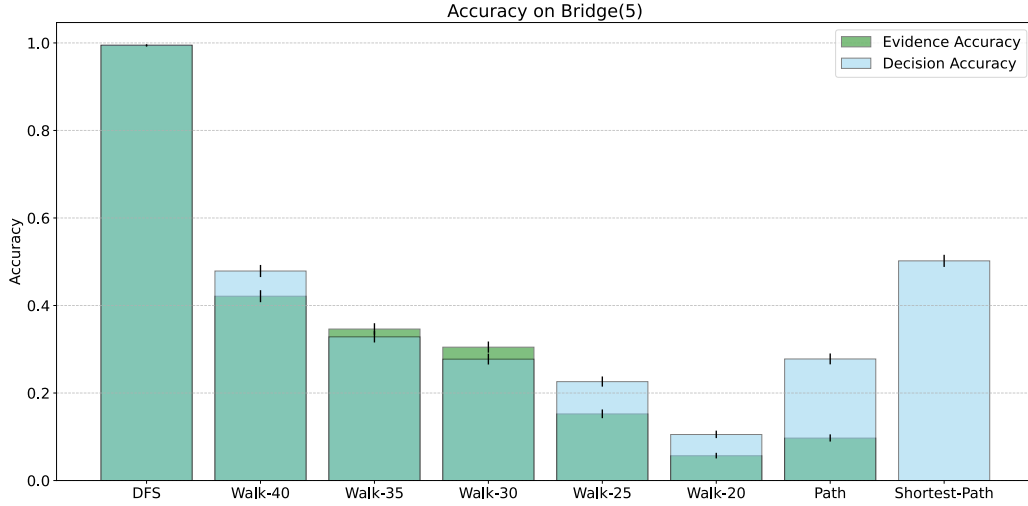


Figure 13: Decision and evidence accuracy of models trained on Walk CoT strategies for Bridge(5) task. Error bars represent 95% binomial confidence intervals.

1062 **Test-Time Scaling** Extensive work focused on scaling inference-time compute optimally [63, 55,  
1063 53, 2], in search of inference-time scaling laws [67, 27, 31, 19]. A line of work has focused on studying  
1064 optimal sequential scaling [39, 74, 25, 7] by examining the role of CoT length [40, 10, 18, 68]. The  
1065 benefits of learning to search [12, 26, 38, 51] and problem-solving strategies like backtracking and  
1066 self-correction [54, 11, 24] by scaling the CoT length have also been demonstrated [37, 73, 69], as  
1067 well as the limits of these approaches [30, 32]. Another line of work has studied parallel scaling [61, 5]  
1068 by examining the behavior of majority voting or a best-of- $n$  method over a diverse set of responses  
1069 generated in parallel [15, 6]. Finally, the role of reinforcement learning [52, 14, 16] in advancing  
1070 reasoning by improving the CoT quality and scaling it naturally [13, 75, 28] has been explored.