Tight Sample Complexity of Transformers

Chenxiao Yang CHENXIAO@TTIC.EDU
Nathan Srebro NATI@TTIC.EDU
Zhiyuan Li ZHIYUANLI@TTIC.EDU
Toyota Technological Institute at Chicago (TTIC)

Editors: Steve Hanneke and Tor Lattimore

Abstract

We tightly characterize the VC dimension of depth-L Transformers with a total of W parameters, mapping an input sequence of length T to a single output, establishing an upper bound of $O(LW\log(TW))$ and a nearly matching lower bound of $\Omega(LW\log(TW/L))$. We further tightly characterize the sample complexity of chain-of-thought learning using such a Transformer, showing teacher forcing (i.e. selecting a predictor consistent with the entire chain-of-thought on training data) learns with sample complexity $O\left(LW\log\left(\left(T+T'\right)W\right)\right)$ and that any learning rule that uses chain-of-thought data requires at least $\Omega\left(LW\log\left(\left(T+T'\right)W/L\right)\right)$ examples, where T is the input length and T' is the number of autoregressive steps.

Keywords: Transformers, VC-dimension, Learning Theory

1. Introduction

Transformers (Vaswani et al., 2017) are a class of models that operate on sequences of arbitrary length, where the architecture and the number of parameters are independent of the input sequence length T. A significant advantage one might then expect is that the sample complexity of learning is independent of the sequence length. However, we do not yet have a rigorous and precise understanding of this sample complexity, and its dependence on the input sequence length and other aspects of the architecture. Tightly characterizing the sample complexity is important both for understanding the length dependence (or lack thereof), as well as for translating results on the representational power of Transformers and their required size (e.g. Yun et al., 2019; Bhattamishra et al., 2020; Hahn, 2020; Pérez et al., 2021; Wei et al., 2021; Liu et al., 2023; Feng et al., 2023; Li et al., 2024; Merrill and Sabharwal, 2023) to actual learning guarantees and understanding the inductive bias of Transformers. We emphasize that in 'learning' here, we are referring to statistical learning by, e.g. empirical risk minimization, and are ignoring the important computational issue of finding such a risk minimizer.

Prior work analyzed the scale-sensitive sample complexity of Transformers, i.e. in terms of norm restrictions on the weights (and under margin assumptions or for scale sensitive loss). In particular, Trauger and Tewari (2023) obtained sample complexity that depends polynomially on the norms of weight matrices (and thus polynomially on the number of parameters under standard scaling), and exponentially on the depth, but as desired is independent of input sequence length[1]. Similarly, Zhang et al. (2022) also established sequence-length independent sample complexity, again by relying on norm bounds, but with only a logarithmic norm dependence, and a suboptimal quadratic dependence on the number of parameters (times a linear dependence on depth).

1. In this regard, these improve over a previous bound (Edelman et al., 2022), that has better polynomial dependency on the norms, but does depend logarithmically on sequence length

In this paper, instead of relying on the magnitudes of the weights and the scale of the outputs, we investigate the parametric capacity of Transformers, i.e. in terms of the number of weights, allowing the weights to be arbitrary real numbers. As the VC dimension tightly captures the sample complexity of PAC-learning Vapnik and Chervonenkis (1971); Blumer et al. (1989), this corresponds to characterizing the VC dimension (or its multiclass generalization) of the class of functions implementable by Transformers, with any real valued weights. This is a basic question about parametric classes and is fundamental in learning theory, with well-understood answers for many classical classes (e.g., linear predictors, polynomials). Even for understanding scale-sensitive or regularization-driven generalization, this serves as an important baseline.

The VC dimension of feed-forward networks has been extensively studied over a span of fifty years (e.g. Cover, 1965; Baum and Haussler, 1989; Maass, 1994; Koiran and Sontag, 1994; Karpinski and Macintyre, 1997; Bartlett et al., 2019), and is now well understood. In particular, we understand that for deep feed-forward networks, the VC dimension can be much larger than the number of parameters and that this greatly depends on the activation function: with hard-threshold activations there is no depth dependence, and the VC dimension is $\Theta(W \log W)$, where W is the total number of weights. With ReLU (or other piecewise linear) activations, the depth comes in linearly and the VC dimension is $\Omega(WL\log(W/L)) \leq \mathrm{VCdim} \leq \mathcal{O}(WL\log W)$. But with sigmoidal activations it significantly worsens (Koiran and Sontag, 1994; Karpinski and Macintyre, 1997). **How about the VC dimension of Transformers? How does it depend on the total number of parameters W, the depth L, the input sequence length T, and perhaps other parameters such as the number of attention heads per layer.**

Based on our experience with feed-forward networks, we would expect Transformer's VC dimension to also depend on the activation functions used. ReLU activations are common for feed-forward layers in practice, and we consider such units here. Attention layer is usually based on a softmax operation. But, a softmax operation hides a logistic activation and can be easily used to simulate it, and so if we allow softmax operations, we encounter the same difficulties and bad scaling as for sigmoidal feed-forward networks discussed above. Instead, we consider hard-attention, which is commonly studied in the theoretical literature on Transformers (e.g. Hahn, 2020; Merrill et al., 2022; Merrill and Sabharwal, 2023; Yang et al., 2025), and avoids sigmoidal, exponential and logarithmic functions.

Transformer VC-Dimension. Accordingly, what we study are Transformers that map T input tokens from some finite-size alphabet Σ to a single output token in Σ (or an embedding sequence to $\{0,1\}$ as classifiers), and consist of token embedding layer, ReLU feed-forward layers, hard-attention layers (average, leftmost or rightmost hard attention, and any number of attention heads), and greedy decoding. We allow arbitrary, but fixed (i.e. not learned) additive positional encoding, including no positional encoding (NOPE, Kazemnejad et al. (2023)). We allow skip connections, but no layerwise-normalization. See precise description in Section 2. We show (in Section 3) that for any such Transformers as classifiers, the VC dimension and thus the sample complexity of learning, is $\mathcal{O}(LW\log(WT))$, where W is the total number of weights, and L is the depth (number of feed-forward and attention layers). The dependence on $\mathcal{O}(LW\log W)$ is tight, even just due to the feed-forward sublayers. We further show that the multiplicative dependence on $\log T$ is also tight by providing a nearly matching lower bound of $\Omega(LW\log(WT/L))$. Edelman et al. (2022) previously establishes a lower bound of $\Omega(\log T)$ even when the number of parameters is fixed, but

this left open the possibility of an additive $\log T$ term—we improve this to show the necessity of the interaction.

Autoregressive CoT with Transformers. In Section 4, we turn to using a Transformer autoregressively to generate a final answer after a chain-of-thought of length T'. Following Joshi et al. (2025), we study the sample complexity of learning to map the input of length T to a single-token output obtained after T' steps, given training data consisting of m i.i.d. inputs and their entire T'-step chain-of-thought. Unlike standard supervised learning where a single measure of VC-dimension governs learnability, we show that two complexity measures govern sample complexity in CoT learning: TSdim (trace shattering dimension), measuring the capacity to shatter entire reasoning traces, governs the upper bound; ASdim (answer shattering dimension), measuring the capacity to shatter only the final answer, governs the CoT lower bound. While combining our single-step VC bound above with the generic result of Joshi et al. (2025) yields a sample complexity upper bound of $O(WL\log(W(T+T'))\log T'/\epsilon)$, directly bounding TSdim improves it to $\mathcal{O}(WL\log(W(T+T'))/\epsilon)$, avoiding the extra $\log T'$ factor. For lower bounds, answer shattering gives the CoT term $\Omega(WL\log(T+T')/\epsilon)$; combined with the ordinary supervised ReLU-network lower bound, this implies the worst-case lower bound $\Omega(WL\log((T+T')W/L)/\epsilon)$. Importantly, the upper bound is obtained by the obvious "teacher forcing" Williams and Zipser (1989) learning rule, i.e. selecting any Transformer in the class that is consistent with all the steps in CoT (as in also Joshi et al., 2025), and the lower bound holds for any learning rule, thus establishing the optimality of teacher forcing for training Transformers with CoT supervision.

Our upper bounds are based on combining proofs about general computation circuits (Bartlett et al., 1998), with more recent refinements for piecewise linear neural nets (Bartlett et al., 2019), and then carefully expressing the attention operation using a poly-sized *constant-depth* computation. The lower bounds use novel explicit constructions.

2. Preliminaries: Transformers and Autoregressive CoT

Throughout this paper, let Σ denote a vocabulary with $|\Sigma| = K$. We write token sequences as $X = (x_0, \ldots, x_{T-1}) \in \Sigma^T$ and embedding sequences as $H \in \mathbb{R}^{T\times d}$, where T is the sequence length and d is the embedding dimension. We adopt Python-style indexing: $X[:t]$ denotes the first t elements, and $X[-t:]$ the last t elements; $X \parallel Z$ denotes sequence concatenation.

We use Transformers in two roles. The fixed-length VC-dimension results use binary classifiers $f^{01} : \mathbb{R}^{T\times d} \to \{0,1\}$. The autoregressive CoT results use next-token generators with input length T, generation length T', and maximum context length $T_{\mathrm{ctx}} = T + T'$.

2.1. Transformers

We first define the Transformer architecture. Our formulation follows the standard decoder-only Transformer (Vaswani et al., 2017), but replaces softmax attention with *hard attention*, a discrete variant that has been central to recent theoretical studies of Transformers (Hahn, 2020; Merrill et al., 2022). As is standard in VC-dimension analysis, we assume exact arithmetic over the reals.

Transformer Layers. A Transformer of depth L has L Transformer blocks. It transforms an input embedding sequence $H^{(0)} = H \in \mathbb{R}^{T\times d}$ through these blocks. We allow layer-dependent hidden dimensions $d^{(0)}, d^{(1)}, \ldots, d^{(L)}$ with $d^{(0)} = d$, so $H^{(\ell)} \in \mathbb{R}^{T\times d^{(\ell)}}$. Each row $H_i^{(\ell)} \in \mathbb{R}^{d^{(\ell)}}$ represents

the embedding of position i. Each block applies two sublayers: an attention mechanism followed by a feed-forward network.

$$\tilde{H}^{(\ell)} = \mathsf{Attn}^{(\ell)}(H^{(\ell-1)}), \quad H^{(\ell)} = \mathsf{FFN}^{(\ell)}(\tilde{H}^{(\ell)}). \tag{1}$$

We state attention-before-FFN blocks; residuals and other fixed orderings only change depth by constants.

Hard Attention. The key component distinguishing Transformers from feed-forward networks is the attention mechanism, which enables data-dependent routing of information across positions. For each attention head, queries, keys, and values are computed via affine transformations:

$$Q_i = W_Q H_i + b_Q, \quad K_i = W_K H_i + b_K, \quad V_i = W_V H_i + b_V, \tag{2}$$

where W_Q, W_K, W_V are projection matrices and b_Q, b_K, b_V are bias vectors (suppressing the layer index ℓ). We use *causal masking*: position i can only attend to positions $j \le i$. We analyze *Hard Attention* (Hahn, 2020; Merrill et al., 2022; Merrill and Sabharwal, 2023), the low-temperature limit of softmax attention where weights concentrate on score-maximizing positions. Let $M_i = \{j \le i : \langle Q_i, K_j\rangle = \max_{k\le i}\langle Q_i, K_k\rangle\}$ denote the set of causally-visible positions achieving the maximum score, and define attention weights $\alpha_{i,j} = \mathbf{1}[j \in M_i]/|M_i|$. The attention output is $\mathsf{Attn}(H)_i = \sum_j \alpha_{i,j} V_j$. Our analysis does not rely on the specific tie-breaking rule and generalizes to variants such as *leftmost-hard* or *rightmost-hard* attention (Hahn, 2020).

Multi-Head Attention. A *multi-head attention* layer at layer ℓ with $h^{(\ell)}$ heads computes $h^{(\ell)}$ attention outputs in parallel. Each head $i \in \{1, \dots, h^{(\ell)}\}$ computes queries, keys, and values via its own projection matrices $W_Q^{(\ell,i)}, W_K^{(\ell,i)}, W_V^{(\ell,i)}$, then applies hard attention as defined above to produce $\mathsf{Attn}^{(\ell,i)}(H)$, with value/output dimension matching the FFN input dimension. The outputs are summed: $\mathsf{Attn}^{(\ell)}(H) = \sum_{i=1}^{h^{(\ell)}} \mathsf{Attn}^{(\ell,i)}(H)$. For simplicity we sum head outputs; the usual concatenate-and-project formulation is covered by allowing head-specific output dimensions and projections.

Feed-Forward Network. The FFN sublayer at layer ℓ is a position-wise MLP with ReLU activations, mapping $\mathbb{R}^{d^{(\ell-1)}} \to \mathbb{R}^{d^{(\ell)}}$. It has at most D_0 internal affine/ReLU layers, for an absolute constant D_0, with arbitrary hidden widths collected in $\bar{d}^{(\ell)}$. Our results can be extended to any piecewise-linear activation, following Bartlett et al. (1998).

Input Layer. For next-token generators operating on token sequences, we first embed tokens into vectors. The *token embedding* $\mathsf{TE} : \Sigma \to \mathbb{R}^d$, parametrized by $W_{\mathsf{TE}} \in \mathbb{R}^{d\times K}$, maps each token to a d-dimensional vector. For a context-length-T architecture, we use a fixed, non-learned positional embedding $\mathsf{PE} : \{0, \dots, T-1\} \to \mathbb{R}^d$. In autoregressive results, $T = T_{\mathrm{ctx}}$, and the same parameters are run on each prefix using the corresponding initial segment of PE. The input embedding sequence is:

$$H = \big(\mathsf{TE}(x_0) + \mathsf{PE}(0), \dots, \mathsf{TE}(x_{T-1}) + \mathsf{PE}(T-1)\big) \in \mathbb{R}^{T\times d}. \tag{3}$$

The positional embedding may be arbitrary, but it is fixed and not counted as a trainable parameter.

Output Layer. For a length-T input, the output is extracted from the final position $H_{T-1}^{(L)} \in \mathbb{R}^{d^{(L)}}$ of the last layer. For an autoregressive prefix of length s, the same rule reads position $s-1$. For binary classifiers, a linear projection $w_{\text{out}} \in \mathbb{R}^{d^{(L)}}$ followed by a threshold yields the binary label. For next-token generators, a decoding matrix $W_{\mathsf{DE}} \in \mathbb{R}^{K \times d^{(L)}}$ maps the final representation to logits over the vocabulary. For a fixed architecture $\mathcal{A}$ and trainable parameter vector θ,

$$f_{\mathcal{A},\theta}^{01}(H) = \mathbf{1}\big[w_{\text{out}}^{\top} H_{T-1}^{(L)} > 0\big] \in \{0,1\}, \qquad f_{\mathcal{A},\theta}(X) = \arg\max_{x \in \Sigma} \big[W_{\mathsf{DE}} H_{T-1}^{(L)}\big]_x \in \Sigma. \tag{4}$$

The argmax uses a fixed deterministic tie-breaking rule. In practice, W_{DE} often shares weights with the token embedding W_{TE} (*weight tying*); our analysis permits both tied and untied configurations.

Hypothesis Classes. We distinguish a fixed architecture from a family of possible architectures. A generic Transformer architecture $\mathcal{A}$ is specified by the choices introduced above:

$$\mathcal{A} = \big\{K,\ L,\ \{d^{(i)}\}_{i=0}^{L},\ \{h^{(\ell)}\}_{\ell=1}^{L},\ \{\bar{d}^{(\ell)}\}_{\ell=1}^{L},\ \mathsf{PE},\ \mathsf{Res}\big\}, \tag{5}$$

where these entries record the architectural choices above. Here Res records a residual/no-residual flag for each attention and FFN sublayer. A residual adds the sublayer input to its output; if dimensions differ, the residual path may use an affine projection, counted in $W(\mathcal{A})$. Let $W(\mathcal{A})$ and $L(\mathcal{A})$ denote the total parameter count and depth. Varying the trainable parameters gives fixed-architecture hypothesis classes; collecting these over all architectures within a budget gives architecture families:

$$\begin{aligned} \mathcal{F}_{\mathcal{A}} &:= \{f_{\mathcal{A},\theta} : \theta \in \mathbb{R}^{W(\mathcal{A})}\}, & \mathcal{F}_{\mathcal{A}}^{01} &:= \{f_{\mathcal{A},\theta}^{01} : \theta \in \mathbb{R}^{W(\mathcal{A})}\}, \\ \mathfrak{F}_{L,W} &:= \{\mathcal{F}_{\mathcal{A}} : L(\mathcal{A}) \le L,\ W(\mathcal{A}) \le W\}, & \mathfrak{F}_{L,W}^{01} &:= \{\mathcal{F}_{\mathcal{A}}^{01} : L(\mathcal{A}) \le L,\ W(\mathcal{A}) \le W\}. \end{aligned} \tag{6}$$

When $\mathcal{A}$ is fixed, we often suppress it and write f_θ or f_θ^{01}. Thus $\mathfrak{F}_{L,W}$ and $\mathfrak{F}_{L,W}^{01}$ are families of fixed-architecture hypothesis classes, not single hypothesis classes.

2.2. Autoregressive Next-Token Generation

Next-Token Generator. A *next-token generator* maps a token context to the next token. We write $f : \Sigma^* \to \Sigma$ for convenience, but in our results f is only evaluated on contexts of length at most T_{ctx}. At each step, the model uses the same f to predict a new token and appends it to the current sequence; we denote this *apply-and-append* operation by $\bar{f} : \Sigma^* \to \Sigma^*$, where $\bar{f}(X) := X \,\|\, f(X)$. Starting from an initial prompt $X \in \Sigma^T$ of *input length* T, the model iteratively applies $\bar{f}$ to generate T' new tokens (the *generation length*):

$$f^{(T')}(X) := \underbrace{\bar{f} \circ \bar{f} \circ \cdots \circ \bar{f}}_{T' \text{ times}}(X) \in \Sigma^{T+T'}. \tag{7}$$

The generated suffix $f^{(T')}(X)[-T':]$ is the CoT trace, and its last token is the final answer.

Teacher Forced Training. In practice, autoregressive models are trained using *teacher forcing* (Williams and Zipser, 1989): at each step, the model predicts the next token conditioned on the *ground-truth* prefix rather than its own previous predictions. Concretely, consider a distribution $\mathcal{D}$ over input-output pairs (X, Z), where $X \in \Sigma^T$ is an input prompt and $Z = (z_0, \ldots, z_{T'-1}) \in \Sigma^{T'}$ is the target sequence. Given m i.i.d. samples $S = \{(X_i, Z_i)\}_{i=1}^{m}$ from $\mathcal{D}$ and a hypothesis class

$\mathcal{F}$ of next-token generators, the learning objective is to find $f \in \mathcal{F}$ minimizing the *empirical next-token prediction loss*:

$$\widehat{\mathcal{L}}(f; S) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{T'} \sum_{t=0}^{T'-1} \mathbf{1}\big[f(X_i \parallel Z_i[:t]) \neq Z_i[t]\big]. \tag{8}$$

End-to-End Evaluation. At test time, however, we ultimately care only about the correctness of the final token $Z[-1]$, treating the intermediate tokens $Z[:-1]$ as auxiliary computation. This is captured by the *population end-to-end loss*:

$$\mathcal{L}_{\text{e2e}}(f) = \mathbb{E}_{(X,Z)\sim\mathcal{D}} \left[\mathbf{1}\big[f^{(T')}(X)[-1] \neq Z[-1]\big]\right], \tag{9}$$

where $f^{(T')}(X)[-1]$ denotes the final token after T' steps of autoregressive generation.

Thus training checks predictions on ground-truth prefixes, while end-to-end evaluation rolls out the model on its own prefixes and keeps only the final token.

3. VC-Dimension of Transformers

For VC-dimension analysis, we focus on the binary classifier family $\mathfrak{F}^{01}_{L,W}$ with context length T, depth L, and W parameters. The upper bound is uniform over every fixed class in this family, while the lower bound is witnessed by one fixed class. The extension to multi-class settings is standard.

3.1. Upper Bound

We first establish an upper bound. The main idea is that hard attention acts as a *discrete switch*: once the attention rankings and ReLU activation patterns are fixed, the network output is polynomial in the parameters. We can then extend the piecewise-polynomial framework for ReLU networks (Bartlett et al., 1998) to account for these additional switches.

Theorem 1 (VC Dimension Upper Bound) *For any depth $L \geq 1$, parameter budget $W \geq L$, and context length $T \geq 1$, let $\mathfrak{F}^{01}_{L,W}$ be the binary Transformer family defined in Section 2.1. Then, for every $\mathcal{F}^{01} \in \mathfrak{F}^{01}_{L,W}$,*

$$\mathrm{VCdim}(\mathcal{F}^{01}) = \mathcal{O}(WL\log(TW)). \tag{10}$$

Proof Sketch. We proceed in two steps: first partition the parameter space by discrete branching patterns, then bound the number of output sign patterns inside each region.

Step 1: Piecewise-Polynomial Framework. We bound the *growth function* $\Pi_{\mathcal{F}^{01}}(H_{1:N})$, which counts the number of distinct output patterns achievable over N embedding sequences; if these N sequences are shattered, this number is 2^N. We partition the parameter space $\mathbb{R}^W$ into regions where all discrete branching decisions are fixed. Within each region, the network output is polynomial in θ, so $\Pi_{\mathcal{F}^{01}}(H_{1:N}) \leq |\mathcal{P}| \cdot \max_C \Pi_C$, where $|\mathcal{P}|$ is the number of regions and Π_C is the number of output patterns within region C. Warren's theorem bounds Π_C: for m polynomials of degree D in W variables, the number of distinct sign vectors is at most $(\mathcal{O}(mD/W))^W$.

Step 2: Counting Branching Polynomials. At each layer, we identify the *branching polynomials* whose signs determine all discrete choices. For the FFN, these are all internal ReLU pre-activations ($\mathcal{O}(NTW)$ polynomials). For attention, these are the pairwise score differences $g^{(\ell)}_{n,i,p,q}(\theta) := \langle q_{n,i}(\theta), k_{n,p}(\theta) - k_{n,q}(\theta)\rangle$, whose signs determine the relative ranking of positions p and q. A W-parameter architecture has at most $\mathcal{O}(W)$ nontrivial heads; with T positions, each head and each query requires $\binom{T}{2} = \mathcal{O}(T^2)$ comparisons, yielding $\mathcal{O}(NWT^3)$ additional branching polynomials per layer. Applying Warren's bound and taking logarithms, the T^3 term contributes $WL\log T$ to the exponent, yielding the stated bound. The complete derivation appears in Appendix B.

The appendix proves a slightly sharper fixed-architecture statement with logarithmic factor $\log(TW(\mathcal{A})L(\mathcal{A}))$; using $L(\mathcal{A}) \le L \le W$ and $W(\mathcal{A}) \le W$ gives the displayed form. ■

3.2. Lower Bound

We now construct a Transformer that simulates a Recursive Retrieval Machine (Figure 1). The context stores all 2^B possible B-bit label configurations, where $B = \lfloor \log_2(T-1) \rfloor$. For each group-layer pair, the parameters specify which configuration to retrieve. One attention lookup then selects one of T positions, realizing $B = \Theta(\log T)$ labels across the corresponding bit-indexed samples. Chaining this lookup across L layers and $n = \Theta(W)$ groups gives the $\Omega(WL\log T)$ term. Our bound improves the $\Omega(\log T)$ bound from Edelman et al. (2022).

Theorem 2 (VC Dimension Lower Bound) *There exists an absolute constant $C_0 > 0$ such that for any depth $L \ge 1$, context length $T \ge 3$, and parameter budget $W \ge C_0 L$, let $\mathfrak{F}^{01}_{L,W}$ be the binary Transformer family defined in Section 2.1. Then there exists $\mathcal{F}^{01} \in \mathfrak{F}^{01}_{L,W}$ such that*

$$\mathrm{VCdim}(\mathcal{F}^{01}) \ge c \cdot WL\left(\log T + \log(W/L)\right), \tag{11}$$

where $c > 0$ is an absolute constant.

Proof Sketch. We construct a set of $N = nLB$ samples that can be shattered, where $n = \Theta(W)$ is the number of sample groups, L is the depth, and $B = \lfloor \log_2(T-1) \rfloor$. Each sample is indexed by a triple (j, ℓ, i) with $j \in [n]$, $\ell \in [L]$, $i \in [B]$. The construction below gives the $WL\log T$ term; the $WL\log(W/L)$ term comes from the standard ReLU-network lower bound, since Transformers contain ordinary position-wise ReLU networks as a subclass. We describe the retrieval construction in three steps: (1) how the context encodes labels, (2) how parameters specify retrieval targets, and (3) how attention and FFN perform recursive retrieval.

Step 1: Context as Label Repository. For each fixed pair (j, ℓ), the $B = \lfloor \log_2(T-1) \rfloor$ labels indexed by $i \in [B]$ are encoded by one retrieved context position. Position $t \in \{0, \ldots, T-2\}$ stores the t-th labeling by encoding $\mathrm{bit}_i(t)$, so the context enumerates all $2^B \le T-1$ label configurations. For target labels $Y_{j,\ell,1}, \ldots, Y_{j,\ell,B}$, the matching position is $t^* = \sum_i Y_{j,\ell,i} 2^{i-1}$.

Step 2: Parameters as Target Addresses. For each group-layer pair, the parameters store the lookup position. The labels $\{Y_{j,\ell,i}\}_{i=1}^{B}$ are aggregated into $s_{j,\ell} = \sum_k Y_{j,\ell,k} 2^{k-1}$. A trainable affine compression maps each group identifier j to an encoded pointer containing all L addresses $(s_{j,1}, \ldots, s_{j,L})$.

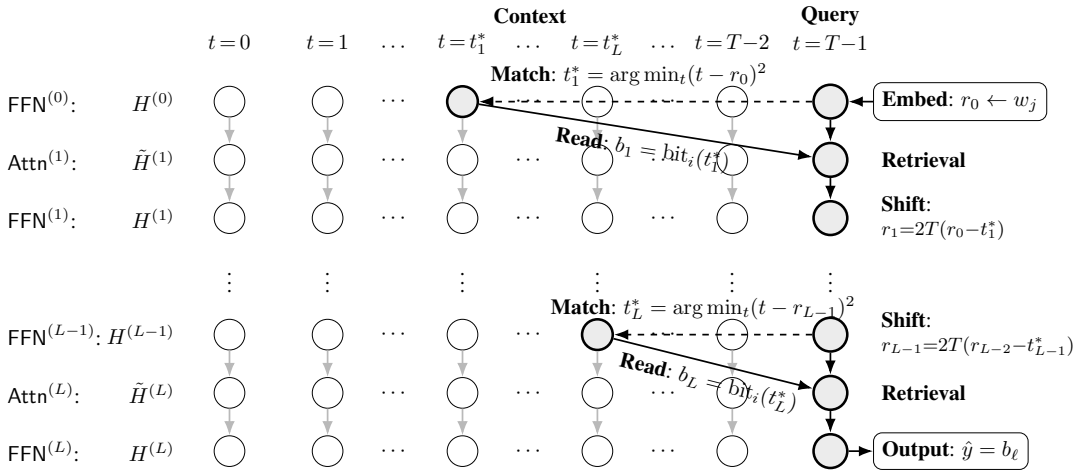

Figure 1: Recursive retrieval gadget for sample (j, ℓ, i). The context positions enumerate all B-bit label configurations, while the query register r_0 stores an encoded list of target addresses $(s_{j,1}, \ldots, s_{j,L})$. At layer k, hard attention selects the context position t_k^* nearest to the current register, the value head reads its i-th bit, and the FFN shifts the register to expose the next address. A gate outputs the bit read at the target layer ℓ.

Step 3: Recursive Retrieval via Attention + FFN. Given a register r encoding target address s, attention selects the context position $t^* = s$: keys extract $(2t, -t^2)$, the query extracts $(r, 1)$, and the score $2rt - t^2$ is maximized at the integer nearest to r. The value projection reads $\mathrm{bit}_i(t^*)$. The base-$2T$ encoding keeps $|r - s| < 1/2$, so s is the unique nearest integer.

To chain L retrievals, set $r_0 := \sum_{m=1}^{L} s_m/(2T)^{m-1}$. After retrieval step k, the FFN performs the left-shift $r_k = 2T(r_{k-1} - t^*)$, exposing the next address. A gate records the retrieved bit only when $k = \ell$.

Putting It Together. This affine compression uses $\Theta(n)$ parameters to store n address sequences; the backbone requires only $\mathcal{O}(L)$ parameters. For a sufficiently large absolute constant C_0, the regime $W \geq C_0 L$ lets us choose $n = \Theta(W)$. We therefore shatter nLB samples, yielding $\Omega(WL \log T)$. Together with the ReLU-subclass lower bound $\Omega(WL \log(W/L))$, this proves Theorem 2. The complete construction appears in Appendix C. ■

4. Auto-Regressive CoT with Transformers

We now turn to the *autoregressive* setting, where next-token generators $f : \Sigma^* \to \Sigma$ iteratively produce variable-length token sequences. This raises a natural question: *how does the availability of full CoT traces during training affect learnability?* We first formalize CoT learnability, then introduce the two dimensions governing the generic upper and lower bounds, and finally instantiate both for Transformers.

4.1. Chain-of-Thought Learnability

We begin by formalizing the CoT learning problem. Fix an input domain $\mathcal{X} \subseteq \Sigma^T$. The base class $\mathcal{F}$ of next-token generators induces the trace and end-to-end hypothesis classes:

$$\mathcal{F}^{(T')} = \left\{ X \mapsto f^{(T')}(X)[-T':] : f \in \mathcal{F} \right\}, \quad \mathcal{F}_{\text{e2e}}^{(T')} = \left\{ X \mapsto f^{(T')}(X)[-1] : f \in \mathcal{F} \right\}. \tag{12}$$

Our goal is to learn the end-to-end class $\mathcal{F}_{\text{e2e}}^{(T')}$. We consider the *realizable* setting: there exists a ground-truth generator $f_* \in \mathcal{F}$. Given an input distribution $\mathcal{D}$ over $\mathcal{X}$, we label each input $X \sim \mathcal{D}$ with the full CoT trace $Z = {f_*}^{(T')}(X)[-T':]$, and seek to output a final-token predictor with small end-to-end error. Realizability means f_* predicts every token in the trace correctly: $f_*(X \parallel Z[:t]) = Z[t]$ for all $t \in \{0, \ldots, T' - 1\}$. Formally, adopting the definition from Joshi et al. (2025):

Definition 3 (Realizable Chain-of-Thought Learnability) *We say $\mathcal{F}_{\text{e2e}}^{(T')}$ is* CoT-learnable *with sample complexity $m(\varepsilon, \delta)$ if there exists a learning rule $A : (\mathcal{X} \times \Sigma^{T'})^* \to \Sigma^{\mathcal{X}}$ such that for every distribution $\mathcal{D}$ over $\mathcal{X}$ and $f_* \in \mathcal{F}$, given $m \geq m(\varepsilon, \delta)$ samples $S = \{(X_i, Z_i)\}_{i=1}^m$ with $X_i \sim \mathcal{D}$ and $Z_i = {f_*}^{(T')}(X_i)[-T':]$, with probability at least $1 - \delta$,*

$$\Pr_{X \sim \mathcal{D}} \Big[A(S)(X) \neq {f_*}^{(T')}(X)[-1] \Big] \leq \varepsilon. \tag{13}$$

Learning Algorithm. When full CoT traces are available, a natural approach is to train the model to match them step by step. This leads to teacher forced training (Section 2.2) as introduced in Section 2. In the realizable setting, it reduces to finding any generator consistent with all observed CoT traces. Formally, we define the learner as

$$\begin{aligned} \text{Cons}_{\text{CoT}}(S): \quad & \text{Return } \hat{f} \in \mathcal{F} \text{ such that} \\ & \hat{f}(X_i \parallel Z_i[:t]) = Z_i[t] \quad \text{for all } (X_i, Z_i) \in S,\ t \in \{0, \ldots, T' - 1\}. \end{aligned} \tag{14}$$

This is a proper learner: it returns a generator $\hat{f} \in \mathcal{F}$, whose induced final-token predictor is $X \mapsto \hat{f}^{(T')}(X)[-1]$. In standard PAC learning, training and test objectives coincide, so ERM is universal, achieving optimal distribution-free sample complexity. For CoT learning, however, there is a fundamental asymmetry: the learner is trained on full traces but evaluated only on the final token. The intermediate tokens carry information that could guide learning, but a learning algorithm might exploit them without matching every step, and at test time, a different reasoning path could still lead to the correct final answer. This raises a natural question: is teacher forcing the universal approach for CoT learning?

A direct generic bound would lose an extra logarithmic factor in T'. The next two subsections remove this loss for Transformers and show that the resulting rate is optimal.

4.2. Dual Complexity Measures

To characterize CoT sample complexity, we introduce two measures that capture the two sides of the train-test asymmetry: the *trace shattering dimension* TSdim for the upper bound, and the *answer shattering dimension* ASdim for the CoT lower bound.

The first measure captures the complexity of predicting the *entire trace* correctly, which requires that every token along the reasoning path match the target. Equivalently, it is the VC dimension of the binary loss class that asks whether a generated trace matches a reference trace.

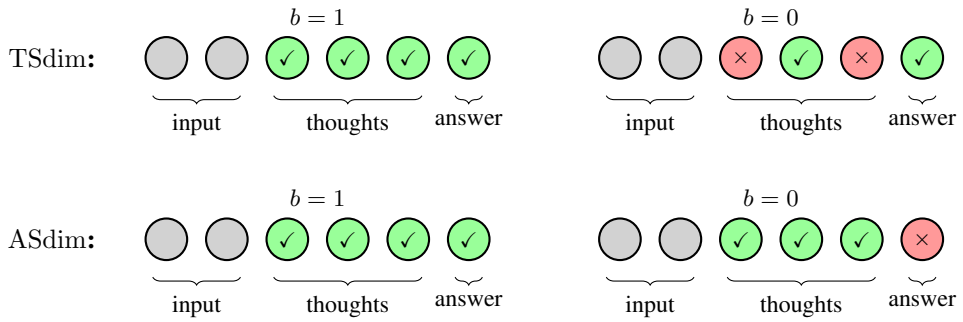

Figure 2: **Top** (Trace Shattering Dimension TSdim): for $b = 1$ the entire trace matches the ground-truth; otherwise some tokens differ. **Bottom** (Answer Shattering Dimension ASdim): for each input, all generators share the same thought prefix and differ only in the final answer.

Definition 4 (Trace Shattering Dimension) *A set $S = \{X_1, \dots, X_n\} \subseteq \mathcal{X}$ is* trace shattered *by $\mathcal{F}$ if there exists a trace assignment $R_S : S \to \Sigma^{T'}$ such that for every $b \in \{0,1\}^n$, some $f_b \in \mathcal{F}$ satisfies for all $i \in [n]$:*

$$b_i = 1 \Longrightarrow f_b^{(T')}(X_i)[-T':] = R_S(X_i), \qquad b_i = 0 \Longrightarrow f_b^{(T')}(X_i)[-T':] \neq R_S(X_i). \tag{15}$$

The trace shattering dimension $\mathrm{TSdim}(\mathcal{F}; \mathcal{X}, T')$ *is the size of the largest such set.*

The second measure captures the complexity of predicting only the *final answer*, namely the last token of the trace. The key constraint is that all generators share the same trace prefix, so the intermediate tokens provide no information about the final answer.

Definition 5 (Answer Shattering Dimension) *Assume $0, 1 \in \Sigma$. A set $S = \{X_1, \dots, X_n\} \subseteq \mathcal{X}$ is* answer shattered *if there exists a prefix assignment $R_S : S \to \Sigma^{T'-1}$ such that for every $b \in \{0,1\}^n$, some $f_b \in \mathcal{F}$ makes the first $T' - 1$ generated tokens equal to this prefix and sets the final token to b_i:*

$$f_b{}^{(T')}(X_i)[-T': -1] = R_S(X_i), \qquad f_b{}^{(T')}(X_i)[-1] = b_i. \tag{16}$$

The answer shattering dimension $\mathrm{ASdim}(\mathcal{F}; \mathcal{X}, T')$ *is the size of the largest such set.*

Main Result. Since answer shattering is a stronger requirement (varying the final token while keeping the prefix fixed necessarily varies the full trace), we have $\mathrm{ASdim} \leq \mathrm{TSdim}$. Unlike standard PAC learning, where the VC-dimension governs both bounds, here the upper bound scales with TSdim while the lower bound scales with ASdim; the gap between them determines how tight this generic characterization is.

Theorem 6 (Generic Sample Complexity Bounds for CoT) *Fix any $(\mathcal{F}, \mathcal{X}, T')$. There exist universal constants $C, c, \delta_0 > 0$ such that:*

(i) For all $\varepsilon, \delta \in (0,1)$, $\mathcal{F}_{\mathrm{e2e}}^{(T')}$ is CoT-learnable (using the learning rule $\mathrm{Cons}_{\mathrm{CoT}}$) with sample complexity

$$m(\varepsilon, \delta) \leq C \cdot \frac{\mathrm{TSdim}(\mathcal{F}; \mathcal{X}, T') \cdot \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}. \tag{17}$$

(ii) Let $d_{\mathrm{AS}} := \mathrm{ASdim}(\mathcal{F}; \mathcal{X}, T')$. *If* $d_{\mathrm{AS}} \geq 2$, *then for all* $\varepsilon \in (0, 1/8)$ *and all* $\delta \leq \delta_0$, *any* CoT-*learnability guarantee requires* $m(\varepsilon, \delta) \geq c \cdot d_{\mathrm{AS}}/\varepsilon$.

The proof is deferred to Appendix D. For the upper bound, we reduce TSdim to the CoT trace loss class and apply standard VC theory. For the lower bound, we construct a hard distribution on an answer shattered set; since the trace prefix is fixed, each observed trace reveals only its own final token, leaving unseen points unpredictable.

4.3. Universality of Teacher Forcing for Transformers

We now instantiate Theorem 6 for Transformer-based next-token generators. The proof follows the same partition-based analysis as Section 3, applied to each intermediate step of the CoT trace. See Appendix E for details.

Theorem 7 (CoT Trace-Shattering Upper Bound) *For any* $L \geq 1$, $W \geq L$, *finite alphabet* $|\Sigma| = K$, *and* $\mathcal{F} \in \mathfrak{F}_{L,W}$, *fix* $\mathcal{X} \subseteq \Sigma^T$, *lengths* T, T', *and* $T_{\mathrm{ctx}} := T + T'$. *Then*

$$\mathrm{TSdim}(\mathcal{F}; \mathcal{X}, T') = O\left(WL \log(T_{\mathrm{ctx}} WLK)\right). \tag{18}$$

In particular, this is $O(WL \log(T_{\mathrm{ctx}} W))$ *whenever* $K \leq \mathrm{poly}(W)$.

A Scratchpad Lower Bound. The upper bound shows that $\mathrm{Cons}_{\mathrm{CoT}}$ *suffices*. But is it *necessary*? Could a smarter algorithm achieve lower sample complexity by exploiting the structure of CoT traces? We show that the answer is *no*: any algorithm requires $\Omega(WL \log T_{\mathrm{ctx}}/\varepsilon)$ samples for Transformers.

The key idea is that CoT traces can act as a label-invariant scratchpad. The generator writes intermediate tokens that depend on the input but not on the labeling, so observing the trace prefix reveals no information about the final token. Nevertheless, the model can use this scratchpad internally, for example as a lookup table, to shatter $\Omega(\log T_{\mathrm{ctx}})$ final answers. We formalize this intuition:

Theorem 8 (CoT Answer-Shattering Lower Bound via Scratchpad) *For any* $N \geq 2$, *there exists a constant-size alphabet* Σ, *an input length* $T = \mathcal{O}(\log \log N)$, *a generation length* $T' = \Theta(N \log N)$, *and a class* $\mathcal{F}$ *of constant-depth, constant-width hard-attention Transformers such that, with* $T_{\mathrm{ctx}} := T + T'$,

$$\mathrm{ASdim}(\mathcal{F}; \Sigma^T, T') = \Omega(\log T_{\mathrm{ctx}}). \tag{19}$$

Proof Sketch. We shatter $m := \lfloor \log_2 N \rfloor$ inputs. The generator writes an input-dependent but label-independent scratchpad, then uses one final readout to recover the requested label bit from a universal table. Let $M := 2^m = \Theta(N)$ and $\ell := \lceil \log_2 m \rceil$. For readability, write the scratchpad using the symbols $\{0, 1, \#, \mathrm{END0}, \mathrm{END1}\}$, where $\#$ is a row separator and $\mathrm{END0}, \mathrm{END1}$ are anchor tokens. The formal proof uses constantly many decorated copies of these tokens to store finite-state carry/borrow information. As in Theorem 2, we use positional features containing τ and τ^2.

Concretely, each **INPUT** $X_i := \mathrm{bin}_\ell(i)\# \in \Sigma^{\ell+1}$ encodes the index i in LSB-first binary. The generated scratchpad has two segments:

- **DECODE**: $\mathrm{bin}_\ell(i-1)\# \to \cdots \to \mathrm{bin}_\ell(0)\# \to \mathrm{END0}$. Counts down from $i-1$ to 0. The anchor token END0 is placed at position $p_0(i) = (i+1)(\ell+1)$; the integer i is encoded into END0's positional encoding.
- **TABLE**: $\mathrm{bin}_m(0)\# \to \mathrm{bin}_m(1)\# \to \cdots \to \mathrm{bin}_m(M-1)\# \to \mathrm{END1}$. Enumerates all $M = 2^m$ binary strings in order, serving as a universal lookup table *identical for all inputs and all labelings*.

Final readout. The labeling $y \in \{0,1\}^m$ is encoded into parameter $s := \sum_j y_j 2^j$, but s is gated to have no effect until the final step, so the scratchpad is label-invariant. Let $L_{\mathrm{in}} := \ell + 1$ and $L_{\mathrm{out}} := m + 1$ be the row lengths in the input/decode and table segments. At the final step, Attn retrieves anchor position $p_0(i)$ from END0; since $p_0(i) = (i+1)L_{\mathrm{in}}$, the FFN recovers i and computes $r = p_0(i) + 1 + sL_{\mathrm{out}} + i$, the position of the i-th bit in the s-th table row. A final attention lookup fetches that bit. Since the s-th row stores $\mathrm{bin}_m(s)$ and $y = \mathrm{bin}_m(s)$, this bit equals y_i.

Since different labelings produce identical traces except for the final token, this achieves answer shattering of m samples. The total generation length is $T' = |\mathrm{DECODE}| + |\mathrm{TABLE}| + c = \mathcal{O}(m \cdot \ell) + \Theta(M \cdot m) + c = \Theta(N \log N)$. Thus $m = \Omega(\log T')$. Since $T = \mathcal{O}(\log \log N)$, we have $T_{\mathrm{ctx}} = T + T' = \Theta(T')$ and hence $m = \Omega(\log T_{\mathrm{ctx}})$, giving $\mathrm{ASdim} \geq \Omega(\log T_{\mathrm{ctx}})$. ■

Recursive Amplification. Amplifying the scratchpad construction by the recursive retrieval idea of Theorem 2 gives the following CoT lower bound. The amplification uses $\Theta(W)$ independent prompt groups and $\Theta(L)$ retrieval rounds; each round contributes one scratchpad-style answer-shattering copy.

Corollary 9 (CoT Lower Bounds for Transformers) *There exist constants $C_0, L_0, c, \delta_0 > 0$ such that for every $L \geq L_0$, $W \geq C_0 L$, and sufficiently large T', there exist a finite alphabet Σ with $K := |\Sigma| \leq \mathrm{poly}(W)$, an input length $T = O(\log \log T')$, a prompt domain $\mathcal{X} \subseteq \Sigma^T$, and a class $\mathcal{F} \in \mathfrak{F}_{L,W}$ such that, with $T_{\mathrm{ctx}} = T + T'$,*

$$\mathrm{ASdim}(\mathcal{F}; \mathcal{X}, T') \geq c\, W L \log T_{\mathrm{ctx}}. \tag{20}$$

Moreover, taking the worst case over token-prompt Transformer classes in $\mathfrak{F}_{L,W}$, for all $\varepsilon \in (0, 1/8)$, the CoT*-learning sample complexity satisfies $m(\varepsilon, \delta_0) \geq c\, W L \log(T_{\mathrm{ctx}} W/L)/\varepsilon$. Here "sufficiently large" includes $T' \geq CWL\log(W/L)$, which is needed only for the token-prompt ReLU witness.*

Universality of Teacher-Forcing / $\mathrm{Cons}_{\mathrm{CoT}}$. Theorems 6 and 7 show that $\mathrm{Cons}_{\mathrm{CoT}}$ learns every $\mathcal{F} \in \mathfrak{F}_{L,W}$ with sample complexity on the order of $WL\log(T_{\mathrm{ctx}} WLK)/\varepsilon$, up to the standard $\log(1/\varepsilon)$ and confidence terms. Conversely, the lower bound has two independent token-prompt sources. The scratchpad construction in Corollary 9, together with Theorem 6, gives the CoT term $WL \log T_{\mathrm{ctx}}/\varepsilon$. The remaining $WL\log(W/L)/\varepsilon$ term is an ordinary supervised obstruction, implemented with token prompts: a marker token selects one fixed positional encoding, after which the Transformer simulates a ReLU MLP. Taking the larger of the two witnesses gives the lower bound in Corollary 9. Thus teacher forcing is optimal in sample complexity for Transformers.

In other words, despite the train-test asymmetry, *matching every intermediate token is the right approach*; exploiting the CoT traces in any other way cannot reduce sample complexity.

5. Discussion

Other Activations. Our main result Theorem 1 extends to piecewise-polynomial activations, e.g. gated ReLU ($(x, y) \mapsto x \cdot [y]_+$) (Shazeer, 2020). The same Warren-based analysis applies, except that the depth dependence becomes $\tilde{O}(WL^2 \log T)$ instead of $\tilde{O}(WL \log T)$: the final output can have degree exponential in L as a piecewise polynomial of the parameters, and taking its logarithm yields an extra factor of L.

Softmax Attention. Our approach for $O(\log T)$ VC dimension does not extend to softmax attention because of the exponential function. To our knowledge, the best upper bound for VC dimension of softmax Transformers is $\mathsf{poly}(T)$, by applying results such as Theorem 8.14 in (Anthony and Bartlett, 2009).

Open Problems. Our results show that teacher forcing is optimal in sample complexity for learning Transformers with CoT supervision. Two questions remain open: (1) Is teacher forcing universally optimal for general hypothesis classes? In other words, can the gap between the trace shattering dimension and the answer shattering dimension be closed? (2) What is the sample complexity when learning the end-to-end mapping without access to intermediate reasoning traces? This setting is discussed by Joshi et al. (2025) and may exhibit fundamentally different scaling.

References

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Peter L Bartlett, Vitaly Maiorov, and Ron Meir. Almost linear vc dimension bounds for piecewise polynomial networks. *Neural computation*, 10(8):2159–2173, 1998.

Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

Eric B Baum and David Haussler. What size net gives valid generalization? *Neural computation*, 1 (1):151–160, 1989.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the ability and limitations of transformers to recognize formal languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Anselm Blumer, André Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik–Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36 (4):929–965, 1989. doi: 10.1145/76359.76371.

Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, EC-14(3):326–334, 1965.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pages 5793–5831. PMLR, 2022.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36, 2023.

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

Nirmit Joshi, Gal Vardi, Adam Block, Surbhi Goel, Zhiyuan Li, Theodor Misiakiewicz, and Nathan Srebro. A theory of learning with autoregressive chain of thought, 2025.

Marek Karpinski and Angus Macintyre. Polynomial bounds for VC dimension of sigmoidal and general pfaffian neural networks. *Journal of Computer and System Sciences*, 54(1):169–176, 1997. doi: 10.1006/jcss.1997.1477.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in Transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Pascal Koiran and Eduardo D Sontag. Vc dimension in circuit complexity. In *Proceedings of the 7th Annual Conference on Computational Learning Theory*, pages 1–10, 1994.

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

Binbing Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Dipendra Misra. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Wolfgang Maass. Neural nets with superlinear vc-dimension. *Neural Computation*, 6(5):877–884, 1994.

William Merrill and Ashish Sabharwal. The expresssive power of transformers with chain of thought. *arXiv preprint arXiv:2310.07923*, 2023.

William Merrill, Ashish Sabharwal, and Noah A Smith. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10:843–856, 2022.

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

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Jacob Trauger and Ambuj Tewari. Sequence length independent norm-based generalization bounds for transformers. *arXiv preprint arXiv:2310.13088*, 2023.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: A case study on Approximating Transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Chenxiao Yang, Nathan Srebro, David McAllester, and Zhiyuan Li. Pencil: Long thoughts with short memory. *arXiv preprint arXiv:2503.14337*, 2025.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Yufeng Zhang, Boyi Liu, Qi Cai, Lingxiao Wang, and Zhaoran Wang. An analysis of attention via the lens of exchangeability and latent variable models. *arXiv preprint arXiv:2212.14852*, 2022.

Contents

Appendix A. Technical Preliminaries

This appendix provides the formal definitions of PAC learnability and VC-dimension used throughout the paper.

A.1. PAC Learning Framework

We begin with the standard framework of *probably approximately correct* (PAC) learning (Valiant, 1984).

Binary Classification. Let $\mathcal{X}$ be a domain and $\mathcal{F}^{01} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class of binary classifiers. A learning algorithm A receives m labeled samples $S = \{(x_i, y_i)\}_{i=1}^m$ drawn i.i.d. from a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, and outputs a hypothesis $\widehat{f} = A(S)$. The *population risk* of a hypothesis f under distribution $\mathcal{D}$ is

$$\mathcal{L}_{\mathcal{D}}(f) := \mathbb{P}_{(x,y)\sim\mathcal{D}}[f(x) \neq y]. \tag{21}$$

A distribution $\mathcal{D}$ is *realizable* by $\mathcal{F}^{01}$ if there exists $f_* \in \mathcal{F}^{01}$ such that $\mathcal{L}_{\mathcal{D}}(f_*) = 0$.

Definition 10 (PAC Learnability) *A hypothesis class $\mathcal{F}^{01}$ is* PAC learnable *if there exists an algorithm A and a function $m : (0,1) \times (0,1) \to \mathbb{N}$ such that for every $\varepsilon, \delta \in (0,1)$ and every distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ that is realizable by $\mathcal{F}^{01}$, given $m(\varepsilon,\delta)$ i.i.d. samples $S \sim \mathcal{D}^{m(\varepsilon,\delta)}$, the algorithm outputs $\widehat{f} = A(S)$ satisfying*

$$\mathbb{P}_{S\sim\mathcal{D}^{m(\varepsilon,\delta)}}\left[\mathcal{L}_{\mathcal{D}}(\widehat{f}) \leq \varepsilon\right] \geq 1 - \delta, \tag{22}$$

where the probability is over the random samples and the algorithm's internal randomness (if any). The function $m(\varepsilon,\delta)$ is called the sample complexity.

A.2. VC-Dimension

The VC-dimension characterizes the sample complexity of PAC learning binary classifiers in the realizable setting.

Definition 11 (Shattering) *A hypothesis class $\mathcal{F}^{01}$* shatters *a finite set $S = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}$ if for every labeling $b \in \{0,1\}^n$, there exists $f \in \mathcal{F}^{01}$ such that $f(x_i) = b_i$ for all $i \in [n]$.*

Definition 12 (VC-Dimension) *The* VC dimension *of $\mathcal{F}^{01}$, denoted $\mathrm{VCdim}(\mathcal{F}^{01})$, is the size of the largest set that can be shattered by $\mathcal{F}^{01}$. If arbitrarily large sets can be shattered, we set $\mathrm{VCdim}(\mathcal{F}^{01}) = \infty$.*

The fundamental theorem of statistical learning (Vapnik and Chervonenkis, 1971; Blumer et al., 1989) establishes that a class is PAC learnable if and only if it has finite VC-dimension. Moreover, the sample complexity is characterized by the VC-dimension:

Theorem 13 (Fundamental Theorem of Statistical Learning) *A hypothesis class $\mathcal{F}^{01}$ is PAC learnable over all distributions if and only if $\mathrm{VCdim}(\mathcal{F}^{01}) < \infty$. Furthermore, the sample complexity satisfies:*

$$\Omega\left(\frac{\mathrm{VCdim}(\mathcal{F}^{01}) + \log(1/\delta)}{\varepsilon}\right) \leq m(\varepsilon,\delta) \leq O\left(\frac{\mathrm{VCdim}(\mathcal{F}^{01})\log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}\right). \tag{23}$$

Appendix B. Proof of VC Dimension Upper Bound (Theorem 1)

We establish the upper bound using the recursive partitioning framework for piecewise-polynomial networks (Bartlett et al., 1998), extended to handle the data-dependent routing of Average-Hard Attention.

The proof proceeds in two steps. First, in Section B.1, we construct a partition of parameter space $\mathbb{R}^W$ into regions where all discrete branching decisions (ReLU activations and attention routing) are fixed. Second, in Section B.2, we count the total number of sign patterns by combining Warren's bound on the number of regions with a local polynomial argument.

B.1. Recursive Parameter Partitioning

Fix any Hard-Attention Transformer architecture $\mathcal{A}$ (as defined in Section 2.1) with depth $L := L(\mathcal{A})$ and parameter count $W := W(\mathcal{A})$. Let $\mathcal{F}$ denote the corresponding binary classifier class

$$\mathcal{F} := \mathcal{F}^{01}_{\mathcal{A}} = \{H \mapsto f^{01}_\theta(H) : \theta \in \mathbb{R}^W\}.$$

For any inputs $H_{1:N} := (H_1, \dots, H_N)$, define

$$\Pi_{\mathcal{F}}(H_{1:N}) \;:=\; \left|\{(f^{01}_\theta(H_1), \dots, f^{01}_\theta(H_N)) : \theta \in \mathbb{R}^W\}\right|. \tag{24}$$

The growth function is $\Pi_{\mathcal{F}}(N) := \max_{H_{1:N}} \Pi_{\mathcal{F}}(H_{1:N})$.

Assume $\mathrm{VCdim}(\mathcal{F}) \geq N$. Then there are N embedding sequences $H_1, \dots, H_N \in \mathbb{R}^{T\times d}$ with $\Pi_{\mathcal{F}}(H_{1:N}) = 2^N$. We will upper bound $\Pi_{\mathcal{F}}(H_{1:N})$ as a function of N, W, L, T; requiring $2^N \leq \Pi_{\mathcal{F}}(H_{1:N})$ then yields an upper bound on $\mathrm{VCdim}(\mathcal{F})$.

Partition Decomposition. Let $\mathcal{P}$ be any partition of $\mathbb{R}^W$ into regions. For each region $C \in \mathcal{P}$, define

$$\Pi_C := \left|\{(f^{01}_\theta(H_1), \dots, f^{01}_\theta(H_N)) : \theta \in C\}\right| \tag{25}$$

as the number of distinct sign patterns realized by parameters in C. Then trivially

$$\Pi_{\mathcal{F}}(H_{1:N}) \leq \sum_{C\in\mathcal{P}} \Pi_C. \tag{26}$$

We construct a partition $\mathcal{P}_L$ (built layer-by-layer) such that: (i) on each $C \in \mathcal{P}_L$, all discrete branching decisions are fixed for $H_{1:N}$, so the pre-threshold output becomes a polynomial in θ; (ii) $|\mathcal{P}_L|$ is controlled by Warren's bound.

Lemma 14 (Warren's Bound, (Bartlett et al., 1998, Lemma 2.1)) *Let $g_1, \dots, g_m$ be polynomials in W variables, each of degree at most D. If $m \geq W$, the number of distinct sign vectors $(\mathrm{sgn}(g_1(\theta)), \dots, \mathrm{sgn}(g_m(\theta))) \in \{-1, 0, 1\}^m$ as θ varies in $\mathbb{R}^W$ is at most $2(2emD/W)^W$.*

Layerwise Refinement. Let $H^{(\ell)}_{n,t}(\theta)$ denote the layer-ℓ hidden representation for input H_n at token position t, viewed as a function of the parameters θ. We define partitions $\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_L$ of $\mathbb{R}^W$, where $\mathcal{P}_\ell$ freezes all branching choices in the first ℓ layers. For $\ell \in \{0, \dots, L\}$, we maintain that for every region $C \in \mathcal{P}_\ell$, every sample $n \in [N]$, and every token $t \in \{0, \dots, T-1\}$, $H^{(\ell)}_{n,t}(\theta)$ coincides on C with a polynomial of degree at most $\deg_\ell$.

Base case. Set $\mathcal{P}_0 := \{\mathbb{R}^W\}$. In our VC analysis, the inputs $H_n \in \mathbb{R}^{T\times d}$ are fixed embedding sequences, so $H_{n,t}^{(0)}(\theta) = H_{n,t}$ is constant in θ (degree 0). (The same argument also covers learnable embeddings, in which case $H^{(0)}$ is affine in θ.) Thus the inductive property holds with $\deg_0 \le 1$.

Inductive step. Fix a layer $\ell \in [L]$ and consider a parent region $C \in \mathcal{P}_{\ell-1}$. By the inductive hypothesis, the inputs to layer ℓ (denoted $H^{(\ell-1)}(\theta)$) are fixed polynomials on C. We refine C into sub-regions by fixing the outcomes of all discrete branching operations at layer ℓ. Since the layer applies attention *before* the FFN (Section 2.1), we do this in two stages: first freeze the discrete routing/tie structure of attention to make $\tilde{H}^{(\ell)} := \mathsf{Attn}^{(\ell)}(H^{(\ell-1)})$ polynomial, then freeze the ReLU activation pattern of the $\mathsf{FFN}^{(\ell)}$ on $\tilde{H}^{(\ell)}$. If residual connections are present, they only add previous polynomial representations, possibly after an affine projection, to sublayer outputs; this preserves the polynomial invariant and introduces no additional branching polynomials, so the same partition argument applies.

1. Attn *routing ($\mathcal{B}_{\mathsf{Attn}}$).* The Average-Hard Attention mechanism routes information based on the *relative order* of attention scores. To determine the set of maximizers, it suffices to determine the sign of the difference between any pair of scores. For each head h, sample n, query position $i \in \{0,\dots,T-1\}$, and ordered key pair $0 \le p < q \le i$ (due to causal masking), we define the pairwise difference polynomial:

$$g_{n,i,p,q}^{(\ell,h)}(\theta) := \left\langle q_{n,i}^{(\ell,h)}(\theta),\, k_{n,p}^{(\ell,h)}(\theta) - k_{n,q}^{(\ell,h)}(\theta) \right\rangle. \tag{27}$$

Since queries q and keys k are affine transformations of the previous layer's polynomial representations, g is a polynomial on C.

We partition C based on the **ternary signs** ($\sigma \in \{+1, 0, -1\}$) of these difference polynomials. This fine-grained partition is crucial for handling the Average-Hard mechanism: fixing the ternary signs uniquely determines the strict ordering and equality relations among all allowed keys, which freezes the set of maximizers $M_{n,i} := \{j \le i \mid \text{score}_j = \max_{k\le i} \text{score}_k\}$ for every query. Once $M_{n,i}$ is fixed, its cardinality becomes a constant integer, and the attention weights $\alpha_{i,j} = \mathbf{1}[j \in M_{n,i}]/|M_{n,i}|$ become fixed rational constants. Thus, within any refined sub-region, the attention output $\mathsf{Attn}(H)_{n,i} = \sum_{j\in M_{n,i}} \frac{1}{|M_{n,i}|} V_{n,j}^{(\ell,h)}(\theta)$ collapses to a fixed linear combination of the polynomial value vectors. Since the values V are affine transformations of $H^{(\ell-1)}$, this preserves the polynomial property. The set $\mathcal{B}_{\mathsf{Attn}}$ collects all such pairwise differences. Accounting for causal masking, for each query position i there are $\binom{i+1}{2}$ key pairs, so each head contributes $\sum_{i=0}^{T-1}\binom{i+1}{2} = \mathcal{O}(T^3)$ polynomials. Under a W-parameter budget, there are at most $\mathcal{O}(W)$ heads, hence $|\mathcal{B}_{\mathsf{Attn}}| = \mathcal{O}(NWT^3)$.

2. ReLU activations in the FFN *($\mathcal{B}_{\mathsf{FFN}}$).* Fix any region where the attention routing decisions above are frozen. On such a region, every intermediate representation $\tilde{H}_{n,t}^{(\ell)}(\theta)$ is polynomial in θ (as argued above). We refine sequentially through the internal affine/ReLU layers of the layer-ℓ FFN. For each sample n, token t, internal FFN layer r, and ReLU hidden unit u in that internal layer, let $a_{n,t,r,u}^{(\ell)}(\theta)$ denote the corresponding pre-activation, computed after the previous internal ReLU masks have been fixed. Since the input to each internal layer is polynomial on the current region, each $a_{n,t,r,u}^{(\ell)}$ is polynomial there. We refine by the signs of all these pre-activations; after all internal masks are fixed, the FFN is a mask-fixed affine network, so its outputs remain polynomial in θ. The set $\mathcal{B}_{\mathsf{FFN}}$ collects all such pre-activations across all ReLU units in the FFN. Since $D^{(\ell)} \le D_0 = \mathcal{O}(1)$, under a W-parameter budget, there are at most $\mathcal{O}(W)$ ReLU units, so $|\mathcal{B}_{\mathsf{FFN}}| = \mathcal{O}(NTW)$.

Remark (sum vs. product). The quantities $|\mathcal{B}_{\mathsf{Attn}}|$ and $|\mathcal{B}_{\mathsf{FFN}}|$ count polynomials, so they add. The sequential composition (Attn then FFN) manifests instead in a *product* bound on the number of refined regions, since we refine first by $\mathcal{B}_{\mathsf{Attn}}$ and then by $\mathcal{B}_{\mathsf{FFN}}$.

Refinement Procedure. Let $\Delta_\ell^{\mathsf{Attn}}$ be the maximum number of child regions obtained by refining a parent region in $\mathcal{P}_{\ell-1}$ using the ternary sign patterns of $\mathcal{B}_{\mathsf{Attn}}$ (padding with constant polynomials if $|\mathcal{B}_{\mathsf{Attn}}| < W$). By Lemma 14,

$$\Delta_\ell^{\mathsf{Attn}} \le 2\Big(\frac{2e|\mathcal{B}_{\mathsf{Attn}}|D_\ell}{W}\Big)^W. \tag{28}$$

Next, within any attention-frozen region, let $\Delta_\ell^{\mathsf{FFN}}$ be the maximum number of child regions obtained by refining using the (binary) sign patterns of $\mathcal{B}_{\mathsf{FFN}}$ (again padding with constants if needed). The constant number of sequential internal refinements is absorbed into constants. Applying Lemma 14 gives

$$\Delta_\ell^{\mathsf{FFN}} \le 2\Big(\frac{2e|\mathcal{B}_{\mathsf{FFN}}|D_\ell}{W}\Big)^W. \tag{29}$$

Let $\Delta_\ell := \max_{C\in\mathcal{P}_{\ell-1}} |\{C' \in \mathcal{P}_\ell : C' \subseteq C\}|$ denote the maximum number of child regions produced by refining any parent region. Since we refine sequentially, each parent region produces at most $\Delta_\ell \le \Delta_\ell^{\mathsf{Attn}} \cdot \Delta_\ell^{\mathsf{FFN}}$ children in $\mathcal{P}_\ell$, hence $|\mathcal{P}_L| \le \prod_{\ell=1}^L \Delta_\ell$.

Degree Bounds. We have the crude parameter-counting bounds

$$|\mathcal{B}_{\mathsf{Attn}}| = \mathcal{O}(NWT^3), \qquad |\mathcal{B}_{\mathsf{FFN}}| = \mathcal{O}(NTW).$$

For degrees: within any region where routing decisions and ReLU masks are fixed, a Transformer layer applies an attention sublayer and an FFN with at most $D_0 = \mathcal{O}(1)$ internal affine/ReLU layers, hence only a constant number of affine maps (degree 1 in θ) to polynomial inputs (degree $\deg_{\ell-1}$). Hence $\deg_\ell \le \deg_{\ell-1} + \mathcal{O}(1) = \mathcal{O}(\ell)$ for representations. For branching polynomials, attention score differences are inner products of queries and keys (each degree $\le \deg_{\ell-1} + \mathcal{O}(1)$), yielding $D_\ell \le 2\deg_{\ell-1} + \mathcal{O}(1) = \mathcal{O}(\ell)$.

B.2. Final Bound Derivation

Total Number of Regions. Combining (28) and (29) across layers:

$$|\mathcal{P}_L| \le \prod_{\ell=1}^L \Delta_\ell \le \prod_{\ell=1}^L \Delta_\ell^{\mathsf{Attn}} \cdot \Delta_\ell^{\mathsf{FFN}} \le \prod_{\ell=1}^L 4\Big(\frac{2e|\mathcal{B}_{\mathsf{Attn}}|D_\ell}{W}\Big)^W \Big(\frac{2e|\mathcal{B}_{\mathsf{FFN}}|D_\ell}{W}\Big)^W. \tag{30}$$

Patterns within a Final Region. Fix a final region $C \in \mathcal{P}_L$. By construction, all branching decisions are fixed, so each output logit $z_n(\theta) := \langle w_{\text{out}}, H_{n,T-1}^{(L)}(\theta)\rangle$ is a polynomial in θ. Since w_{out} is included in θ, $\deg_{\text{out}} \le \deg_L + 1 = \mathcal{O}(L)$. If $N < W$, the desired VC upper bound is immediate after increasing constants, so assume $N \ge W$. Applying Lemma 14 to the N polynomials $\{z_n\}_{n=1}^N$ yields

$$\Pi_C \le 2\Big(\frac{2eN\deg_{\text{out}}}{W}\Big)^W. \tag{31}$$

Putting it Together. From (26), $\Pi_{\mathcal{F}}(H_{1:N}) \le \sum_{C\in\mathcal{P}_L} \Pi_C \le |\mathcal{P}_L| \cdot \max_{C\in\mathcal{P}_L} \Pi_C$. Combining with (30) and (31):

$$\Pi_{\mathcal{F}}(H_{1:N}) \le |\mathcal{P}_L| \cdot \max_C \Pi_C \le \left(\prod_{\ell=1}^{L} 4\Big(\frac{2e|\mathcal{B}_{\mathsf{Attn}}|D_\ell}{W}\Big)^W \Big(\frac{2e|\mathcal{B}_{\mathsf{FFN}}|D_\ell}{W}\Big)^W\right) \cdot 2\Big(\frac{2eN\deg_{\text{out}}}{W}\Big)^W. \tag{32}$$

If $H_{1:N}$ is shattered, then $\Pi_{\mathcal{F}}(H_{1:N}) = 2^N$. Taking logarithms:

$$N \le \mathcal{O}(WL) + \sum_{\ell=1}^{L} W\log\Big(\frac{C\,|\mathcal{B}_{\mathsf{Attn}}|D_\ell}{W}\Big) + \sum_{\ell=1}^{L} W\log\Big(\frac{C\,|\mathcal{B}_{\mathsf{FFN}}|D_\ell}{W}\Big) + W\log\Big(\frac{C\,N\deg_{\text{out}}}{W}\Big). \tag{33}$$

Substituting $|\mathcal{B}_{\mathsf{Attn}}| \le c_1 NWT^3$, $|\mathcal{B}_{\mathsf{FFN}}| \le c_2 NTW$, and $D_\ell = \mathcal{O}(\ell)$, the factor W cancels with the $/W$:

$$\log\Big(\frac{C\,|\mathcal{B}_{\mathsf{Attn}}|D_\ell}{W}\Big) \le \log(C'\,NT^3\ell), \qquad \log\Big(\frac{C\,|\mathcal{B}_{\mathsf{FFN}}|D_\ell}{W}\Big) \le \log(C''\,NT\ell). \tag{34}$$

Also $\deg_{\text{out}} = \mathcal{O}(L)$, so (33) yields

$$N \le \mathcal{O}(WL\log N + WL\log T + WL\log L). \tag{35}$$

Inverting the Inequality. We use the standard implication: if $A \ge 2$, $B \ge 0$, and $N \le A\log N + B$, then $N = \mathcal{O}(A\log A + B)$. Applying this to (35) with $A = \mathcal{O}(WL)$ and $B = \mathcal{O}(WL(\log T + \log L))$ gives

$$N \le \mathcal{O}(WL\log(WL) + WL\log T + WL\log L) = \mathcal{O}(WL\log(TWL)). \tag{36}$$

Hence $\mathrm{VCdim}(\mathcal{F}) = \mathcal{O}(WL\log(TWL))$. When $L \le W$, this simplifies to $\mathrm{VCdim}(\mathcal{F}) = \mathcal{O}(WL\log(TW))$. ∎

Appendix C. Proof of VC Dimension Lower Bound (Theorem 2)

We provide a constructive proof. Throughout this section, W denotes the *total* number of trainable parameters. We work at a fixed sequence length T (the same T in all parts). We index positions by $t \in \{0,\dots,T-1\}$, where the last token ($t = T-1$) plays the role of the query/output token. For the Recursive Retrieval Machine construction, we use residual connections, which are allowed in our architecture class.

The proof has two components. First, in Section C.1, we present a Recursive Retrieval Machine construction that yields a class $\mathcal{F}^{01}_{\text{RRM}} \in \mathfrak{F}^{01}_{L,W}$ with VC-dimension $\Omega(WL\log T)$. Second, in Section C.2, we demonstrate that the same Transformer setup contains a standard deep ReLU-network subclass $\mathcal{F}^{01}_{\text{ReLU}} \in \mathfrak{F}^{01}_{L,W}$ with VC-dimension $\Omega(WL\log(W/L))$. One of these two classes witnesses the final lower bound.

C.1. Recursive Retrieval Machine Lower Bound $\Omega(WL\log T)$

Goal and Setup. Let $B = \lfloor \log_2(T-1) \rfloor$, so that $2^B \le T-1$. Note that $B = \Theta(\log T)$ for $T \ge 3$, so we freely write $\log T$ in asymptotic bounds. Define a sample set S of size $N := n \cdot L \cdot B$, indexed by (j,ℓ,i) with $j \in [n]$, $\ell \in [L]$, $i \in [B]$. Fix an arbitrary labeling $Y \in \{0,1\}^N$, written as $Y_{j,\ell,i}$. Our goal is to construct weights θ so that $f^{01}_\theta(H^{(j,\ell,i)}) = Y_{j,\ell,i}$ for all points in S.

Input Construction. Since our classifiers take embedding sequences as input, we directly specify inputs $H^{(j,\ell,i)} \in \mathbb{R}^{T\times d^{(0)}}$ of length T, consisting of T embedding vectors:

$$H^{(j,\ell,i)} = \left[h_0^{(j,\ell,i)},\, h_1^{(j,\ell,i)},\, \ldots,\, h_{T-2}^{(j,\ell,i)},\, h_{T-1}^{(j,\ell,i)}\right]^\top. \tag{37}$$

We append two scalar coordinates to every embedding: (i) a constant coordinate $c \equiv 1$ (present in all positions), and (ii) an indicator coordinate η (equal to 1 on context positions and 0 on the query position).

(i) Context sequence. For each $t \in \{0,\ldots,T-2\}$, define the context embedding

$$h_t^{(j,\ell,i)} \;=\; \left[t,\; -t^2,\; \mathrm{bit}_i(t),\; c=1,\; \eta=1,\; \mathbf{0}\right]^\top. \tag{38}$$

The component $\mathrm{bit}_i(t) \in \{0,1\}$ is the i-th bit of integer t (in any fixed convention). Thus position $t \in \{0,\ldots,T-2\}$ stores the bit pattern of integer $t \in \{0,\ldots,T-2\}$.

(ii) Query token. Let $\mathbf{e}_j \in \mathbb{R}^n$ and $\mathbf{e}_\ell \in \mathbb{R}^L$ be one-hot vectors. The query token activates the memory address j and the target layer ℓ:

$$h_{T-1}^{(j,\ell,i)} \;=\; \left[0,\; 0,\; 0,\; c=1,\; \eta=0,\; \mathbf{e}_j^\top,\; \mathbf{e}_\ell^\top,\; \mathbf{0}\right]^\top. \tag{39}$$

Parameter Encoding. For each address j and layer ℓ, aggregate the B labels into an integer address $s_{j,\ell} := \sum_{k=0}^{B-1} Y_{j,\ell,k+1}\, 2^k \in \{0,\ldots,2^B-1\} \subseteq \{0,\ldots,T-2\}$, which corresponds to context position $s_{j,\ell} \in \{0,\ldots,T-2\}$. Then compress $(s_{j,1},\ldots,s_{j,L})$ into a single real number via base-$2T$ encoding:

$$a_j \;:=\; \sum_{m=1}^{L} \frac{s_{j,m}}{(2T)^m} \;\in\; [0, 0.5). \tag{40}$$

We designate a specific scalar dimension in the hidden state to act as an *address register* r. The value a_j is not part of the input; it is stored in the trainable weights. The one-hot vector $\mathbf{e}_j$ only tells the first block which stored number to use.

Reading the identifiers once. The one-hot coordinates $\mathbf{e}_j$ and $\mathbf{e}_\ell$ are used only in the first Transformer block. In that block, trainable linear maps read them once: the query map uses $\mathbf{e}_j$ to produce the scalar address $r_0 = 2Ta_j$ in the first attention score, and the FFN/residual part records the few scalars needed later, such as the updated address register, the target-layer number $g = \ell$, and the accumulator. The context features $(t, -t^2, \mathrm{bit}_i(t), c, \eta)$ already have constant dimension and are carried forward by residual connections. After the first block, the network no longer carries $\mathbf{e}_j$ or $\mathbf{e}_\ell$; all later layers operate on a constant number of coordinates. This first read costs $\Theta(n)$ parameters for choosing a_j from $\mathbf{e}_j$ and $\mathcal{O}(L)$ parameters for choosing g from $\mathbf{e}_\ell$.

Layer Dynamics. We analyze execution through an inductive invariant. We reserve disjoint coordinate subspaces: (1) a *context subspace* that stores (embedded versions of) the context features $t, -t^2, \mathrm{bit}_i(t), c, \eta$ and is never overwritten; (2) a *query work subspace* that stores the scalar register r_k, the selected index t^*, the extracted bit b_k, and an accumulator z. The residual path carries the context coordinates unchanged from layer to layer. The attention and FFN updates may read these coordinates, but they write only to the query work coordinates. In particular, the computations that update the query work subspace never overwrite the context features needed for later retrievals.

Lemma 15 (Inductive Extraction) *Suppose that at retrieval step k ($1 \le k \le L$) the scalar address available to the query projection equals $r_{k-1} = \sum_{m=k}^{L} \frac{s_{j,m}}{(2T)^{m-k}}$. For $k = 1$, this value is produced directly from $\mathbf{e}_j$ by the first block's query map; for $k \ge 2$, it is the register value left by the previous block. Then:*

1. *The attention mechanism uniquely attends to the context token $t^* = s_{j,k}$.*
2. *The* FFN *updates the register to $r_k = \sum_{m=k+1}^{L} \frac{s_{j,m}}{(2T)^{m-(k+1)}}$.*

Proof Attention step (Nearest-Neighbor Retrieval). We configure the query/key projections so that the attention score at the query position for a context token t equals

$$\langle q, k_t \rangle \;=\; 2r_{k-1} \cdot t \;-\; t^2 \;+\; M\eta_t, \tag{41}$$

where $M > 0$ is a sufficiently large constant. Concretely, this is achieved by choosing $k_t = [\, t,\, -t^2,\, \eta_t \,]$ and $q = [\, 2r_{k-1},\, 1,\, M \,]$. Observe that maximizing the term $2r_{k-1}t - t^2$ is equivalent to minimizing the squared Euclidean distance between the register and t:

$$\operatorname*{argmax}_t (2r_{k-1}t - t^2) = \operatorname*{argmin}_t |t - r_{k-1}|^2. \tag{42}$$

The term $M\eta_t$ simply acts as a mask to exclude the query token itself (where $\eta_{T-1} = 0$). Thus, the attention mechanism effectively searches for the position $t \in \{0, \ldots, T-2\}$ such that t is closest to r_{k-1}.

Now write $r_{k-1} = s_{j,k} + \delta_k$, where

$$\delta_k := \sum_{m=k+1}^{L} \frac{s_{j,m}}{(2T)^{m-k}}.$$

Using $0 \le s_{j,m} \le T-2$, we have

$$0 \le \delta_k \le (T-2) \sum_{p=1}^{\infty} (2T)^{-p} = \frac{T-2}{2T-1} < 0.5. \tag{43}$$

Since $|\delta_k| < 0.5$, the integer $s_{j,k}$ is strictly closer to r_{k-1} than any other integer. Thus $t = s_{j,k}$ is the unique minimizer, i.e., $t^* = s_{j,k}$ is the selected position. Since the maximizer is unique, Average-Hard Attention reduces to standard deterministic selection.

Update step (Left-Shift Operation). We choose the value projection so that the attention output retrieves two quantities: (i) the value $t^* = s_{j,k}$ (read from the first coordinate in (38)), and (ii) the payload bit $b_k = \mathrm{bit}_i(s_{j,k})$ (read from the third coordinate).

The FFN then computes the next register value using $t^* = s_{j,k}$:

$$r_k := 2T(r_{k-1} - s_{j,k}) = 2T\delta_k. \tag{44}$$

This is the base-$2T$ left shift. It is affine on the bounded relevant domain, so a two-layer ReLU FFN implements it exactly. Substituting the definition of δ_k confirms that

$$r_k = \sum_{m=k+1}^{L} \frac{s_{j,m}}{(2T)^{m-(k+1)}},$$

satisfying the inductive hypothesis. ∎

Output Gating via Scalar Addressing. To select the output of the target layer ℓ without violating the constant-width constraint (i.e., avoiding an L-dimensional state), we employ a scalar addressing mechanism. As described above, the first block maps the target index $\mathbf{e}_\ell$ to the scalar $g := \ell$, which is stored in a protected coordinate. At each layer k, we implement a localized indicator function using a ReLU "hat" construction:

$$\mathrm{gate}_k(g) := 2\Big(\mathrm{ReLU}\left(g - \left(k - \frac{1}{2}\right)\right) - 2\,\mathrm{ReLU}(g-k) + \mathrm{ReLU}\left(g - \left(k + \frac{1}{2}\right)\right)\Big). \quad (45)$$

One verifies that for integer $g \in \{1, \ldots, L\}$, $\mathrm{gate}_k(g) = \mathbf{1}\{g = k\}$. We then maintain an accumulator z (initialized to 0) via the conditional update:

$$z \leftarrow z + \mathrm{ReLU}\left(b_k + \mathrm{gate}_k(g) - 1\right). \quad (46)$$

Since $b_k \in \{0, 1\}$, the update term $\mathrm{ReLU}(b_k + \mathrm{gate}_k(g) - 1)$ simplifies to b_k if $k = \ell$ (where $\mathrm{gate}_k(g) = 1$) and vanishes otherwise. Thus, the final state satisfies $z = b_\ell = Y_{j,\ell,i}$. Finally, choosing w_{out} such that the output logit equals $z - \frac{1}{2}$ ensures that the sign activation outputs the correct label.

Parameter Budget Analysis. The construction utilizes an asymmetric parameter allocation. The only parameters that scale with n or L are the first-block weights that read $\mathbf{e}_j$ and $\mathbf{e}_\ell$: $\Theta(n)$ weights choose the encoded value a_j, and $\mathcal{O}(L)$ weights choose $g = \ell$. In contrast, the Transformer backbone requires only $\mathcal{O}(L)$ parameters, as the Attn and FFN weights at each layer implement fixed algebraic operations on a constant-dimensional subspace, independent of n and T. Thus, the total parameter count is $W = \Theta(n) + \mathcal{O}(L)$. For a sufficiently large absolute constant C_0, the regime $W \geq C_0 L$ lets us choose $n = \Theta(W)$. Let $\mathcal{F}^{01}_{\mathrm{RRM}} \in \mathfrak{F}^{01}_{L,W}$ denote the fixed architecture class used in this construction. Since each sample shatters $\Omega(L \log T)$ bits (specifically, B bits across L layers),

$$\mathrm{VCdim}(\mathcal{F}^{01}_{\mathrm{RRM}}) \geq n \cdot L \cdot B = \Omega(WL \log T). \quad (47)$$

C.2. ReLU-Network Subclass and Combined Lower Bound

We complement the Recursive Retrieval Machine construction with a depth-dependent lower bound by identifying a standard ReLU network subclass within the Transformer architecture.

Lemma 16 (ReLU Subclass Containment) *For any $T \geq 2$, $L \geq 1$, and $W \geq L$, there exists a class $\mathcal{F}^{01}_{\mathrm{ReLU}} \in \mathfrak{F}^{01}_{L,W}$ that contains, as a subclass, standard ReLU networks with $\Theta(L)$ layers and $\Theta(W)$ trainable parameters acting on the query position.*

Proof We restrict to inputs where all context tokens are fixed, and only the query token varies. By choosing attention parameters so that, at the query position, the unique score-maximizing key is the query token itself (e.g., using a marker coordinate that equals 1 only on the query token), hard attention acts as an identity map on the query token. Hence, on this subclass, the residual-free Transformer reduces to iterated application of the position-wise $\mathsf{FFN}^{(\ell)}$ on the query token, i.e., a deep ReLU MLP (up to constant-factor depth) with $\Theta(W)$ parameters. ■

It is a standard result that depth-L' ReLU networks with W' parameters have VC-dimension $\Omega(W'L' \log(W'/L'))$ (Bartlett et al., 2019). Applying this to our subclass with $L' = \Theta(L)$ and $W' = \Theta(W)$ yields a lower bound of $\Omega(WL \log(W/L))$.

Combined Lower Bound. Combining the two regimes, one of the two fixed classes witnesses the sum up to constants:

$$\exists\, \mathcal{F}^{01} \in \mathfrak{F}^{01}_{L,W} \quad \text{s.t.}$$
$$\mathrm{VCdim}(\mathcal{F}^{01}) \geq \Omega\left(WL\log T + WL\log\frac{W}{L}\right) = \Omega\left(WL\log\frac{TW}{L}\right). \tag{48}$$

■

Appendix D. Proof of Generic Sample Complexity Bounds for CoT (Theorem 6)

In this appendix, we provide the detailed proofs for Theorem 6. We first establish two auxiliary lemmas, then prove the upper and lower bounds.

Full statement (for reference). Fix $(\mathcal{F}, \mathcal{X}, T')$ and let $d_{\mathrm{TS}} := \mathrm{TSdim}(\mathcal{F}; \mathcal{X}, T')$ and $d_{\mathrm{AS}} := \mathrm{ASdim}(\mathcal{F}; \mathcal{X}, T')$. For every realizable pair $(\mathcal{D}, f_*)$ and every $\varepsilon, \delta \in (0,1)$, if

$$m \geq C \cdot \frac{d_{\mathrm{TS}}\log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}, \tag{49}$$

then with probability at least $1-\delta$ over an i.i.d. sample $S_{\mathrm{CoT}} \sim \mathcal{D}^m$ labeled by traces $Z_i = {f_*}^{(T')}(X_i)[-T':]$, the output $\hat{f} = \mathrm{Cons}_{\mathrm{CoT}}(S_{\mathrm{CoT}})$ satisfies $\mathcal{L}_{\mathrm{e2e}}(\hat{f}) \leq \varepsilon$. Moreover, if $d_{\mathrm{AS}} \geq 2$, there exist universal constants $c_0, c_1 > 0$ such that for any learning rule $A : (\mathcal{X} \times \Sigma^{T'})^* \to \Sigma^{\mathcal{X}}$ and any $\varepsilon \in (0, 1/8)$, if $m < c_0 \cdot d_{\mathrm{AS}}/\varepsilon$, then there exist a realizable distribution $\mathcal{D}$ over $\mathcal{X}$ and a target $f_* \in \mathcal{F}$ such that for $S_{\mathrm{CoT}} \sim \mathcal{D}^m$ labeled by traces $Z_i = {f_*}^{(T')}(X_i)[-T':]$,

$$\Pr\left[\Pr_{X\sim\mathcal{D}}\left[A(S_{\mathrm{CoT}})(X) \neq {f_*}^{(T')}(X)[-1]\right] \geq \varepsilon\right] \geq c_1. \tag{50}$$

CoT loss class. For any generator $f \in \mathcal{F}$ and any pair $(X, Z) \in \mathcal{X} \times \Sigma^{T'}$, define the binary CoT loss

$$\ell_f(X, Z) := \mathbf{1}\left[f^{(T')}(X)[-T':] \neq Z\right]. \tag{51}$$

Let $\mathcal{L}_{\mathrm{CoT}} := \{\ell_f : f \in \mathcal{F}\}$.

D.1. Auxiliary Lemmas

Lemma 17 (End-to-end error is dominated by CoT loss) *For every $f \in \mathcal{F}$ and every realizable pair $(\mathcal{D}, f_*)$,*

$$\mathcal{L}_{\mathrm{e2e}}(f) \leq \mathbb{E}_{X\sim\mathcal{D}}\left[\ell_f\left(X, {f_*}^{(T')}(X)[-T':]\right)\right]. \tag{52}$$

Proof Fix $X \in \mathcal{X}$ and write $Z := {f_*}^{(T')}(X)[-T':] \in \Sigma^{T'}$. If $f^{(T')}(X)[-T':] = Z$, then in particular $f^{(T')}(X)[-1] = Z[-1]$, so

$$\mathbf{1}\left[f^{(T')}(X)[-1] \neq {f_*}^{(T')}(X)[-1]\right] \leq \mathbf{1}\left[f^{(T')}(X)[-T':] \neq {f_*}^{(T')}(X)[-T':]\right] = \ell_f(X, Z) \tag{53}$$

pointwise. Taking expectation over $X \sim \mathcal{D}$ proves the claim. ■

Lemma 18 (TSdim **equals the VC-dimension of the CoT loss class)**

$$\mathrm{TSdim}(\mathcal{F}; \mathcal{X}, T') = \mathrm{VCdim}(\mathcal{L}_{\mathrm{CoT}}), \tag{54}$$

where $\mathcal{L}_{\mathrm{CoT}} = \{\ell_f : f \in \mathcal{F}\}$ *with* $\ell_f(X, Z) = \mathbf{1}[f^{(T')}(X)[-T':] \neq Z]$.

Proof *(*$\geq$*)* Suppose $S = \{X_1, \ldots, X_n\}$ is trace shattered with witness $R_S : S \to \Sigma^{T'}$. Consider the n labeled examples in the loss domain: $\{(X_i, R_S(X_i))\}_{i=1}^n$. For any $b \in \{0,1\}^n$, by definition there exists $f_b \in \mathcal{F}$ such that $\ell_{f_b}(X_i, R_S(X_i)) = 0$ when $b_i = 1$ and $= 1$ when $b_i = 0$. Thus $\mathcal{L}_{\mathrm{CoT}}$ shatters these n points, so $\mathrm{VCdim}(\mathcal{L}_{\mathrm{CoT}}) \geq n$.

($\leq$*)* Conversely, suppose $\mathrm{VCdim}(\mathcal{L}_{\mathrm{CoT}}) \geq n$. Then there exist $(X_i, y^{(i)}_{0:T'-1}) \in \mathcal{X} \times \Sigma^{T'}$ for $i \in [n]$ such that for every label vector $a \in \{0,1\}^n$ there exists $f_a \in \mathcal{F}$ with $\ell_{f_a}(X_i, y^{(i)}_{0:T'-1}) = a_i$. We may assume the inputs X_i are distinct. Indeed, if $X_i = X_j$ but $y^{(i)}_{0:T'-1} \neq y^{(j)}_{0:T'-1}$, no deterministic generator can realize loss 0 on both points; if the two traces are equal, the two loss-domain points are identical and cannot both belong to a shattered set of distinct points. Let $S = \{X_1, \ldots, X_n\}$ and define $R_S(X_i) := y^{(i)}_{0:T'-1}$. For any $b \in \{0,1\}^n$, apply shattering with $a_i := 1 - b_i$. Then $\ell_{f_a}(X_i, R_S(X_i)) = 0$ when $b_i = 1$ and $= 1$ when $b_i = 0$, which means ${f_a}^{(T')}(X_i)[-T':] = R_S(X_i)$ if $b_i = 1$ and differs if $b_i = 0$. So S is trace shattered and $\mathrm{TSdim}(\mathcal{F}; \mathcal{X}, T') \geq n$. ■

D.2. Upper Bound

Proof [Proof of the upper bound in Theorem 6] By Lemma 18, $\mathrm{VCdim}(\mathcal{L}_{\mathrm{CoT}}) = d_{\mathrm{TS}}$. In the realizable setting, $\mathrm{Cons}_{\mathrm{CoT}}$ can output some $\hat{f}$ with zero empirical loss on the sample (i.e., $\ell_{\hat{f}}(X_i, Z_i) = 0$ for all i), where $Z_i = {f_*}^{(T')}(X_i)[-T':]$.

It remains to show that *any* hypothesis in a VC class that is consistent on m samples has small true error. A standard VC generalization bound (realizable case) states that there exists a universal constant $C > 0$ such that if

$$m \geq C \cdot \frac{d_{\mathrm{TS}} \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}, \tag{55}$$

then with probability at least $1 - \delta$,

$$\mathbb{E}_{X \sim \mathcal{D}}\left[\ell_{\hat{f}}\left(X, {f_*}^{(T')}(X)[-T':]\right)\right] \leq \varepsilon. \tag{56}$$

Finally, Lemma 17 gives $\mathcal{L}_{\mathrm{e2e}}(\hat{f}) \leq \varepsilon$. ■

D.3. Lower Bound

Proof [Proof of the lower bound in Theorem 6] Let $d := d_{\mathrm{AS}}$ and assume $d \geq 2$ (otherwise the claim is trivial). By Definition 5, there exist a set $S = \{X_1, \ldots, X_d\} \subseteq \mathcal{X}$, a prefix assignment $R_S : S \to \Sigma^{T'-1}$, such that for every $b \in \{0,1\}^d$ there exists $f_b \in \mathcal{F}$ with

$$ {f_b}^{(T')}(X_i)[-T': -1] = R_S(X_i), \qquad {f_b}^{(T')}(X_i)[-1] = b_i \quad \forall i \in [d], \tag{57}$$

so the first $T' - 1$ generated tokens are fixed while the final token equals b_i.

Fix $\varepsilon \in (0, 1/8)$. We define a hard distribution $\mathcal{D}$ supported on S: put a large mass on X_1 and spread the remaining mass over $X_2, \ldots, X_d$:

$$\mathcal{D}(X_1) = 1 - 4\varepsilon, \qquad \mathcal{D}(X_i) = \frac{4\varepsilon}{d-1} \quad \text{for } i = 2, \ldots, d. \tag{58}$$

Random target choice. Let $B = (B_1, \ldots, B_d) \in \{0,1\}^d$ be uniformly random, and choose the corresponding target generator $f_B \in \mathcal{F}$ guaranteed above. We analyze the performance of an arbitrary learning rule A under the joint randomness of B and the sample $S_{\mathrm{CoT}} \sim \mathcal{D}^m$ labeled by full traces of f_B.

Let $\widehat{g} := A(S_{\mathrm{CoT}}) \in \Sigma^{\mathcal{X}}$ be the final-token predictor returned by the learner. Define the *rare-points risk*

$$\mathcal{L}_{\mathrm{rare}} := \sum_{i=2}^{d} \mathcal{D}(X_i) \cdot \mathbf{1}\big[\widehat{g}(X_i) \neq B_i\big]. \tag{59}$$

Note that $0 \leq \mathcal{L}_{\mathrm{rare}} \leq \sum_{i=2}^{d} \mathcal{D}(X_i) = 4\varepsilon$.

Key observation: if a point is unseen, its label is unpredictable. Fix $i \in \{2, \ldots, d\}$ and let E_i be the event that X_i does not appear in the sample S_{CoT}. On the event E_i, the entire observed sample (inputs and traces) is independent of the bit B_i: indeed, the sample contains traces only for points X_j that were drawn, and for every such point $X_j \in S$ the trace prefix is the fixed value $R_S(X_j)$ (independent of B), while the final token reveals only B_j for those drawn indices j. Formally, conditioned on E_i, the sample is a function of $\{B_j : X_j \text{ appears in } S_{\mathrm{CoT}}\}$ and is therefore independent of B_i. Since X_i never appears under E_i, the bit B_i is never revealed in the data and remains uniform $\{0,1\}$ even conditioned on the realized sample. Therefore,

$$\mathbb{P}\big[\widehat{g}(X_i) \neq B_i \mid E_i\big] \geq \frac{1}{2}. \tag{60}$$

Taking expectation, $\mathbb{P}\big[\widehat{g}(X_i) \neq B_i\big] \geq \mathbb{P}(E_i) \cdot \frac{1}{2}$.

Compute $\mathbb{P}(E_i)$. Each draw hits X_i with probability $4\varepsilon/(d-1)$, hence $\mathbb{P}(E_i) = \big(1 - \frac{4\varepsilon}{d-1}\big)^m$.

Lower bound the expected risk. By linearity of expectation and symmetry over $i = 2, \ldots, d$,

$$\mathbb{E}[\mathcal{L}_{\mathrm{rare}}] = \sum_{i=2}^{d} \mathcal{D}(X_i) \cdot \mathbb{P}\big[\widehat{g}(X_i) \neq B_i\big] \geq \sum_{i=2}^{d} \frac{4\varepsilon}{d-1} \cdot \frac{1}{2} \cdot \Big(1 - \frac{4\varepsilon}{d-1}\Big)^m = 2\varepsilon\Big(1 - \frac{4\varepsilon}{d-1}\Big)^m. \tag{61}$$

Using Bernoulli's inequality $(1-u)^m \geq 1 - mu$ for $u \in [0,1]$, if $m \leq \frac{d-1}{16\varepsilon}$, then $(1 - \frac{4\varepsilon}{d-1})^m \geq \frac{3}{4}$, so $\mathbb{E}[\mathcal{L}_{\mathrm{rare}}] \geq 2\varepsilon \cdot \frac{3}{4} = \frac{3}{2}\varepsilon$. Since $d \geq 2$, the assumption $m < d/(32\varepsilon)$ implies $m \leq (d-1)/(16\varepsilon)$, so this condition is satisfied after adjusting constants.

From expectation to probability. Let

$$p := \mathbb{P}_{B, S_{\mathrm{CoT}}}(\mathcal{L}_{\mathrm{rare}} \geq \varepsilon).$$

Since $\mathcal{L}_{\mathrm{rare}} \leq 4\varepsilon$, we have $\mathbb{E}[\mathcal{L}_{\mathrm{rare}}] \leq \varepsilon \cdot (1-p) + 4\varepsilon \cdot p = \varepsilon + 3\varepsilon p$. Thus $\mathbb{E}[\mathcal{L}_{\mathrm{rare}}] \geq \frac{3}{2}\varepsilon$ implies $p \geq \frac{1}{6}$. In other words, $\mathbb{P}_{B, S_{\mathrm{CoT}}}(\mathcal{L}_{\mathrm{rare}} \geq \varepsilon) \geq \frac{1}{6}$.

Fix a hard target. By averaging over B, there exists a particular $b^* \in \{0,1\}^d$ such that, letting $f_* := f_{b^*}$,

$$\mathbb{P}_{S_{\mathrm{CoT}} \sim \mathcal{D}^m}\left(\sum_{i=2}^{d} \mathcal{D}(X_i) \cdot \mathbf{1}\big[A(S_{\mathrm{CoT}})(X_i) \neq {f_*}^{(T')}(X_i)[-1]\big] \geq \varepsilon\right) \geq \frac{1}{6}. \tag{62}$$

Since the full end-to-end risk dominates the rare-points risk $\mathcal{L}_{\mathrm{rare}}$, the same lower bound holds for $\mathbb{P}\big(\mathbb{P}_{X\sim\mathcal{D}}[A(S_{\mathrm{CoT}})(X) \neq {f_*}^{(T')}(X)[-1]] \geq \varepsilon\big)$. Setting $c_0 = 1/32$ (absorbing constants) and $c_1 = 1/6$ proves the theorem. ■

Appendix E. Proof of CoT Sample Complexity Upper Bound (Theorem 7)

Proof [Proof of Theorem 7] Throughout this proof, we use the notation from Theorem 7: T' denotes the generation length, and $T_{\mathrm{ctx}} = T + T'$ denotes the total context length. Fix any $\mathcal{F} \in \mathfrak{F}_{L,W}$.

We prove the TSdim bound for this fixed class by upper bounding the VC-dimension of the CoT loss class. Recall the CoT loss class:

$$\ell_f(X, Z) := \mathbf{1}\big[f^{(T')}(X)[-T':] \neq Z\big], \qquad \mathcal{L}_{\mathrm{CoT}} := \{\ell_f : f \in \mathcal{F}\}. \tag{63}$$

Equivalently (since f is deterministic), $\ell_f(X, Z) = 0$ iff $f(X \,\|\, Z[:t]) = Z[t]$ for all $t \in \{0, \ldots, T'-1\}$. Let $K := |\Sigma|$. By Lemma 18 (with $T \leftarrow T'$),

$$\mathrm{TSdim}(\mathcal{F}; \mathcal{X}, T') = \mathrm{VCdim}(\mathcal{L}_{\mathrm{CoT}}). \tag{64}$$

Hence it suffices to show $\mathrm{VCdim}(\mathcal{L}_{\mathrm{CoT}}) = \mathcal{O}(WL\log(T_{\mathrm{ctx}}WLK))$.

Step 1: Unroll teacher-forcing prefixes and reduce to a growth bound for $\mathcal{F}$. Fix $n \geq 1$ and consider any n labeled examples $\{(X_i, Z_i)\}_{i=1}^n \subseteq \mathcal{X} \times \Sigma^{T'}$. For each $(i, t) \in [n] \times \{0, \ldots, T'-1\}$, define the teacher-forcing prefix context

$$c_{i,t} := X_i \,\|\, Z_i[:t]. \tag{65}$$

Each $c_{i,t}$ has length at most $T + t \leq T_{\mathrm{ctx}} - 1$.

Let $M := nT'$ and enumerate these contexts as $\{c_j\}_{j=1}^M := \{c_{i,t}\}_{(i,t)\in[n]\times\{0,\ldots,T'-1\}}$. Define the multi-class growth function for next-token generators:

$$\Pi_{\mathcal{F}}(c_{1:M}) := \big|\big\{(f(c_1), \ldots, f(c_M)) \in \Sigma^M : f \in \mathcal{F}\big\}\big|, \qquad \Pi_{\mathcal{F}}(M) := \max_{\substack{c_{1:M} \\ |c_m| \leq T_{\mathrm{ctx}}\ \forall m}} \Pi_{\mathcal{F}}(c_{1:M}). \tag{66}$$

Now observe: for a fixed f, the loss vector $(\ell_f(X_1, Z_1), \ldots, \ell_f(X_n, Z_n)) \in \{0,1\}^n$ is a deterministic function of the prediction vector $(f(c_1), \ldots, f(c_M)) \in \Sigma^M$, since $\ell_f(X_i, Z_i) = 0$ iff $f(c_{i,t}) = Z_i[t]$ for all $t \in \{0, \ldots, T'-1\}$.

Therefore, the number of distinct $\{0,1\}^n$ labelings realized by $\mathcal{L}_{\mathrm{CoT}}$ on $\{(X_i, Z_i)\}_{i=1}^n$ is at most $\Pi_{\mathcal{F}}(c_{1:M}) \leq \Pi_{\mathcal{F}}(M)$. In particular, if $\mathcal{L}_{\mathrm{CoT}}$ shatters these n examples, then 2^n distinct loss vectors occur, so

$$2^n \leq \Pi_{\mathcal{F}}(nT'). \tag{67}$$

Step 2: A growth bound for multi-class Transformer next-token generators. We claim there exist universal constants $C, C' > 0$ such that for all $M \geq 1$,

$$\log \Pi_{\mathcal{F}}(M) \leq C \cdot WL \cdot \log(C'MT_{\mathrm{ctx}}WLK). \tag{68}$$

Proof [Proof of (68)] Fix arbitrary contexts $c_{1:M}$, each of length at most T_{ctx}, and write $\tau_m := |c_m| - 1$ for the last position of context c_m. Consider the parameter vector $\theta \in \mathbb{R}^W$. We apply

the same recursive partitioning construction as in the VC upper bound proof (Appendix B), with $N \leftarrow M$ and context length T_{ctx}. Concretely, at each layer we first refine by the attention score-difference polynomials (freezing Average-Hard routing/ties), and then refine by all internal FFN ReLU pre-activations (freezing ReLU masks), exactly as in Appendix B. Note that since the token embedding matrix W_{TE} is included in θ, the layer-0 representations are affine in θ; this is already accounted for in the base case of Appendix B. Thus we obtain a final partition $\mathcal{P}_L$ of $\mathbb{R}^W$ such that:

(i) For every cell $C \in \mathcal{P}_L$, all branching decisions are fixed on $c_{1:M}$;

(ii) On each $C \in \mathcal{P}_L$, every hidden representation $H_{m,t}^{(L)}(\theta)$ is polynomial in θ with degree $\deg_{\mathrm{out}} = \mathcal{O}(L)$;

(iii) The number of cells satisfies (by the same Warren bound calculation as the VC upper bound proof):

$$\log |\mathcal{P}_L| \leq \mathcal{O}(WL \log(MT_{\mathrm{ctx}}WL)). \tag{69}$$

It remains to bound, for a fixed final cell $C \in \mathcal{P}_L$, how many distinct next-token prediction vectors $(f_\theta(c_1), \ldots, f_\theta(c_M))$ can arise as θ ranges over C.

For each input $m \in [M]$ and token $x \in \Sigma$, define the output logit

$$z_{m,x}(\theta) := \langle W_{\mathsf{DE}}[x, :], H_{m,\tau_m}^{(L)}(\theta) \rangle. \tag{70}$$

On a fixed cell C, $H_{m,\tau_m}^{(L)}(\theta)$ is polynomial in θ of degree $\mathcal{O}(L)$, so the logit $z_{m,x}(\theta)$ is polynomial in θ of degree at most $\mathcal{O}(L)+1 = \mathcal{O}(L)$ (since the decoding matrix W_{DE} is a coordinate projection of θ). The predicted token is $f_\theta(c_m) = \arg\max_{x \in \Sigma} z_{m,x}(\theta)$, with any fixed tie-breaking rule.

For multi-class argmax outputs, it suffices to control the ternary sign pattern of all pairwise logit differences:

$$\Delta_{m,x,x'}(\theta) := z_{m,x}(\theta) - z_{m,x'}(\theta), \qquad m \in [M],\ x < x' \in \Sigma. \tag{71}$$

There are $m_{\mathrm{out}} = M\binom{K}{2} = \mathcal{O}(MK^2)$ such polynomials, each of degree at most $D_{\mathrm{out}} = \mathcal{O}(L)$. On any region where all these ternary signs are fixed, the induced ordering (with ties) of the logits is fixed for every m, hence the argmax output $f_\theta(c_m)$ is fixed as well. Therefore, the number of distinct prediction vectors realized within C is at most the number of distinct ternary sign patterns of $\{\Delta_{m,x,x'}\}$.

Applying Warren's bound (Lemma 14) to these m_{out} polynomials (padding with constant polynomials if $m_{\mathrm{out}} < W$) gives

$$\log \Pi_C \leq \mathcal{O}(W \log(MKL)), \tag{72}$$

where Π_C denotes the number of distinct prediction vectors realized by parameters in cell C.

Finally, summing over cells,

$$\Pi_{\mathcal{F}}(c_{1:M}) \leq \sum_{C \in \mathcal{P}_L} \Pi_C \leq |\mathcal{P}_L| \cdot \max_{C \in \mathcal{P}_L} \Pi_C. \tag{73}$$

Combining (69) and (72), absorbing the within-cell term into $O(WL \log(MT_{\mathrm{ctx}}WLK))$, yields (68). ∎

Step 3: Conclude the VC bound for $\mathcal{L}_{\mathrm{CoT}}$. Combine (67) with (68) at $M = nT'$:

$$n \le \log \Pi_{\mathcal{F}}(nT') \le C \cdot WL \cdot \log(C' nT' T_{\mathrm{ctx}} WLK). \tag{74}$$

Using $\log(nT'T_{\mathrm{ctx}}WLK) \le \log n + \log(T'T_{\mathrm{ctx}}WLK)$ and $T' \le T_{\mathrm{ctx}}$, we get

$$n \le C \cdot WL \cdot \log n + C \cdot WL \cdot \log(T_{\mathrm{ctx}}WLK) + \mathcal{O}(WL). \tag{75}$$

Applying the standard inversion (if $n \le a \log n + b$ then $n = \mathcal{O}(a \log a + b)$), with $a = \Theta(WL)$ and $b = \Theta(WL \log(T_{\mathrm{ctx}}WLK))$, yields

$$\mathrm{VCdim}(\mathcal{L}_{\mathrm{CoT}}) = \mathcal{O}(WL \log(T_{\mathrm{ctx}}WLK)). \tag{76}$$

When $K \le \mathrm{poly}(W)$, this simplifies to $\mathcal{O}(WL \log(T_{\mathrm{ctx}}W))$. By (64), this proves the TSdim bound.

Step 4: Consequence for sample complexity. Plugging $d_{\mathrm{TS}} = \mathrm{TSdim}(\mathcal{F}; \mathcal{X}, T')$ into Theorem 6 gives

$$m(\varepsilon, \delta) = O\left(\frac{d_{\mathrm{TS}} \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}\right), \tag{77}$$

and substituting $d_{\mathrm{TS}} = \mathcal{O}(WL \log(T_{\mathrm{ctx}}WLK))$ yields the corresponding sample-complexity upper bound. Since $\mathcal{F} \in \mathfrak{F}_{L,W}$ was arbitrary, the TSdim bound holds for every class in the family. ■

Appendix F. Proof of CoT Sample Complexity Lower Bound

In this appendix, we prove the lower bounds for CoT sample complexity. We first establish that the answer shattering dimension can grow logarithmically with the generation length (Theorem 8), then prove the Transformer scratchpad ASdim lower bound and combine it with the ordinary supervised obstruction (Corollary 9).

F.1. Logarithmic Shattering via Scratchpad (Theorem 8)

Proof [Proof of Theorem 8] We construct a constant-depth, constant-width autoregressive Transformer whose first $T' - 1$ generated tokens are independent of the labeling, while the final token realizes an arbitrary labeling of $\Omega(\log T_{\mathrm{ctx}})$ inputs.

Next-position convention. At the step that generates the token in absolute position τ, the visible prefix occupies positions $0, \dots, \tau - 1$, and the Transformer output is read from the last visible position $\tau - 1$, as in Section 2.2. Since the positional encoding contains the scalar position, the representation at the last visible position can compute the next-position scalar $\tau = (\tau - 1) + 1$. Below, when we say that the current step uses τ, we mean this computed next-position scalar; no extra query token is introduced.

Assume first that N is larger than a universal constant; smaller N are absorbed by constants. Let

$$m := \lfloor \log_2 N \rfloor, \qquad M := 2^m, \tag{78}$$

so $M = \Theta(N)$. We answer-shatter m inputs. Set

$$\ell := \lceil \log_2 m \rceil, \qquad L_{\mathrm{in}} := \ell + 1, \qquad L_{\mathrm{out}} := m + 1, \qquad T := L_{\mathrm{in}}. \tag{79}$$

Decorated constant-size alphabet. The main text displays the scratchpad using the symbols $\{0, 1, \#, \mathrm{END0}, \mathrm{END1}\}$. Formally, we use constantly many decorated copies of these tokens. These are still ordinary tokens in a constant-size alphabet; the decorations simply store the finite-state information, such as carry and borrow bits, in the generated prefix itself. Let

$$\Sigma = \{0, 1, \mathrm{END0}, \mathrm{END1}, \mathrm{PAD}\} \cup \Gamma_{\mathrm{sep}} \cup \Gamma_{\mathrm{dec}} \cup \Gamma_{\mathrm{tab}}, \tag{80}$$

where

$$\Gamma_{\mathrm{sep}} = \{\#^z_{\mathrm{in}}, \#^z_{\mathrm{dec}} : z \in \{0, 1\}\} \cup \{\#_{\mathrm{tab}}\}, \tag{81}$$

$$\Gamma_{\mathrm{dec}} = \{D_{b,\rho,z} : b, \rho, z \in \{0, 1\}\}, \qquad \Gamma_{\mathrm{tab}} = \{T_{b,\rho} : b, \rho \in \{0, 1\}\}. \tag{82}$$

Here b is the visible bit. In $D_{b,\rho,z}$, ρ is the outgoing borrow for the decrement rule and z records whether all bits generated so far in the current decode row are zero. In $T_{b,\rho}$, ρ is the outgoing carry for the increment rule. The superscript on $\#^z_{\mathrm{in}}$ and $\#^z_{\mathrm{dec}}$ records whether the preceding row is all zero. These decorations are deterministic functions of the input and the already generated prefix, and are never label-dependent.

For each integer a, let $\mathrm{bin}_r(a)$ be the r-bit binary representation of a in least-significant-bit first order. Define

$$x_i := \mathrm{bin}_\ell(i)\#_{\mathrm{in}}^{\mathbf{1}[i=0]} \in \Sigma^T, \qquad i = 0, \ldots, m-1, \tag{83}$$

and let $S := \{x_0, \ldots, x_{m-1}\}$.

Addressing and finite-state primitives. We use fixed quadratic positional features: the token at absolute position p contains positional coordinates $(p, -p^2, 1)$. Thus a hard-attention head can retrieve any visible integer position r: with query $q(r) = (2r, 1, 0)$ and keys $k_p = (p, -p^2, 1)$, the score is

$$\langle q(r), k_p \rangle = 2rp - p^2 = r^2 - (p - r)^2, \tag{84}$$

whose unique visible maximizer is $p = r$.

We also use constant-size ReLU equality gates. For integer a, define

$$\mathrm{eq}_a(u) := 2\left(\mathrm{ReLU}\left(u - a + \frac{1}{2}\right) - 2\mathrm{ReLU}(u - a) + \mathrm{ReLU}\left(u - a - \frac{1}{2}\right)\right). \tag{85}$$

For every integer u, $\mathrm{eq}_a(u) = \mathbf{1}[u = a]$. Any Boolean function of constantly many token-type indicators, state bits, and such equality gates can be implemented by a constant-size ReLU FFN. All constants such as m, M and the final position are fixed as part of the architecture for this value of N.

The label-invariant scratchpad. Fix an arbitrary labeling $y = (y_0, \ldots, y_{m-1}) \in \{0, 1\}^m$ and encode it as

$$s = s(y) := \sum_{i=0}^{m-1} y_i 2^i \in \{0, \ldots, M - 1\}, \qquad \text{so that} \qquad \mathrm{bit}_i(s) = y_i. \tag{86}$$

The only label-dependent trainable scalar is

$$\beta_s := sL_{\mathrm{out}}. \tag{87}$$

It is gated off until the final generation step. Set

$$T' := mL_{\text{in}} + ML_{\text{out}} + 3. \tag{88}$$

For input x_i, the first $T' - 1$ generated tokens form a prefix $R_S(x_i)$ independent of s. If decorations are ignored, this prefix is

$$\text{bin}_\ell(i-1)\#,\ \text{bin}_\ell(i-2)\#,\ \ldots,\ \text{bin}_\ell(0)\#,\ \text{END0} \tag{89}$$

(empty before END0 when $i = 0$), followed by

$$\text{bin}_m(0)\#,\ \text{bin}_m(1)\#,\ \ldots,\ \text{bin}_m(M-1)\#,\ \text{END1}, \tag{90}$$

followed by PAD tokens until the prefix length is exactly $T' - 1$.

Generating the scratchpad. We describe the next-token rule. A constant number of heads retrieve: (i) the latest boundary token among the decorated separators, END0, and END1, using a score with a large type bonus plus the position coordinate; (ii) the unique END0 token if it has appeared; (iii) the unique END1 token if it has appeared; (iv) the same-column predecessor at position $\tau - L_{\text{in}}$; and (v) the same-column predecessor at position $\tau - L_{\text{out}}$. The quadratic addressing primitive implements the constant-offset predecessor retrievals. When such a predecessor position is not visible, the corresponding value is ignored by the FFN.

If END1 has already appeared and the current step is not the final step, the model outputs PAD. If END0 has appeared but END1 has not, the model is in the table phase. Let p_0 be the position of END0. The table ends when

$$\text{eq}_{ML_{\text{out}}}\big(\tau - (p_0 + 1)\big) = 1, \tag{91}$$

at which point the model outputs END1. Otherwise, write $k = \tau - b - 1$, where b is the latest boundary position. If $k = m$, the model outputs $\#_{\text{tab}}$. If $0 \le k < m$, it outputs the next bit of the current table row. The first row after END0 is all zeros. Later rows are obtained by incrementing the previous row by one in LSB-first order:

$$v_k = q_k \oplus c_k, \qquad c_{k+1} = q_k \wedge c_k, \tag{92}$$

where q_k is the bit retrieved from position $\tau - L_{\text{out}}$, and c_k is 1 for $k = 0$ and otherwise the carry bit stored in the previous generated table token. The emitted token is $T_{v_k, c_{k+1}}$.

If END0 has not appeared, the model is in the decode phase. If the latest separator is $\#^1_{\text{in}}$ or $\#^1_{\text{dec}}$, then the preceding row is all zero and the model outputs END0. Otherwise let $k = \tau - b - 1$. If $k = \ell$, the model has completed a decode row; it reads the zero-so-far flag z from the previous generated decode token and outputs $\#^z_{\text{dec}}$. If $0 \le k < \ell$, the model decrements the previous row by one in LSB-first order:

$$v_k = q_k \oplus \rho_k, \qquad \rho_{k+1} = (1 - q_k) \wedge \rho_k, \tag{93}$$

where q_k is the bit retrieved from position $\tau - L_{\text{in}}$, and ρ_k is 1 for $k = 0$ and otherwise the borrow bit stored in the previous generated decode token. The zero-so-far flag is updated as

$$z_{k+1} = \begin{cases} \mathbf{1}[v_0 = 0], & k = 0, \\ z_k \wedge \mathbf{1}[v_k = 0], & k > 0. \end{cases} \tag{94}$$

The emitted token is $D_{v_k,\rho_{k+1},z_{k+1}}$.

All these rules are Boolean functions of constantly many retrieved symbols, state bits, and equality gates, and hence are implemented exactly by a constant-size FFN. They generate the desired decode segment, then the universal table, then padding. In particular, END0 appears at absolute position

$$p_0(i) = (i+1)L_{\text{in}}. \tag{95}$$

The entire generated prefix is independent of s.

Gating the label-dependent scalar. Let

$$\tau_\star := T + T' - 1 \tag{96}$$

be the absolute position of the final generated token, and set $G_{\text{fin}} := \text{eq}_{\tau_\star}(\tau)$. Since $0 \le \beta_s \le ML_{\text{out}}$, choose a hard-wired constant $B > ML_{\text{out}}$. The product of the label-dependent scalar with the final-step gate is implemented by

$$\beta_s^{\text{gated}} := \text{ReLU}\big(\beta_s - B(1 - G_{\text{fin}})\big). \tag{97}$$

This equals 0 before the final step and equals β_s at the final step. The output logits are gated similarly, so the scratchpad rule is used before the final step and the readout rule below is used at the final step.

Final readout. At the final step, the visible prefix contains END0, the full table, END1, and padding. The readout uses two consecutive attention sublayers. The first attention sublayer retrieves the unique END0 token and writes its position $v = p_0(i)$ into a work coordinate. The following FFN computes

$$r = \left(1 + \frac{1}{L_{\text{in}}}\right) v + \beta_s^{\text{gated}}. \tag{98}$$

At the final step, $\beta_s^{\text{gated}} = \beta_s = sL_{\text{out}}$. Since $v = p_0(i) = (i+1)L_{\text{in}}$,

$$r = p_0(i) + i + 1 + sL_{\text{out}}. \tag{99}$$

The table starts at position $p_0(i) + 1$, and row s starts at $p_0(i) + 1 + sL_{\text{out}}$. Thus the position of the i-th bit of row s is

$$p^\star = p_0(i) + 1 + sL_{\text{out}} + i = r. \tag{100}$$

The second attention sublayer uses query $q(r) = (2r, 1, 0)$ and retrieves the token at $p^\star$. Its visible bit is $\text{bit}_i(s) = y_i$, and the output layer maps visible bit $0/1$ to the answer token $0/1$.

Therefore, for every labeling $y \in \{0,1\}^m$, choosing $\beta_s = sL_{\text{out}}$ gives a generator f_s such that, for every $i \in \{0, \dots, m-1\}$,

$$f_s^{(T')}(x_i)[-T' : -1] = R_S(x_i), \qquad f_s^{(T')}(x_i)[-1] = y_i. \tag{101}$$

Hence S is answer shattered, and

$$\text{ASdim}(\mathcal{F}; \Sigma^T, T') \ge |S| = m. \tag{102}$$

Size and length. The construction uses a constant number of heads, a constant number of layers, and constant hidden dimension: the only work coordinates are constantly many positions, Boolean flags, and token-type indicators. Residual connections or identity sublayers copy these work coordinates between the two readout hops.

Finally,

$$T = L_{\text{in}} = O(\log m) = O(\log \log N), \tag{103}$$

and

$$T' = mL_{\text{in}} + ML_{\text{out}} + 3 = \Theta(Mm) = \Theta(N \log N). \tag{104}$$

Thus

$$T_{\text{ctx}} = T + T' = \Theta(T') = \Theta(N \log N), \tag{105}$$

and so $m = \Theta(\log N) = \Omega(\log T_{\text{ctx}})$. This proves the theorem. ■

F.2. CoT Lower Bounds for Transformers (Corollary 9)

Proof [Proof of Corollary 9] We prove the scratchpad amplification lower bound.

Scratchpad amplification. Let C_{amp} be a sufficiently large universal constant and set

$$R := \left\lfloor \frac{L}{C_{\text{amp}}} \right\rfloor. \tag{106}$$

The constant C_{amp} accounts for one recursive retrieval round, the constant-depth scratchpad transducer of Theorem 8, and the constant-depth final readout. Choose $L_0 \geq 2C_{\text{amp}}$; since $L \geq L_0$, we have $R \geq 1$.

Choose m as the largest integer such that, with $M := 2^m$,

$$M(m+1) \leq T'/C \tag{107}$$

for a sufficiently large universal constant C. Then $m = \Theta(\log T')$ and $M \leq T'$. Let

$$\ell := \lceil \log_2 m \rceil, \qquad L_{\text{in}} := \ell + 1, \qquad L_{\text{out}} := m + 1. \tag{108}$$

Alphabet and inputs. Let Γ be the constant-size decorated alphabet used in the proof of Theorem 8. Enlarge it by identifier and round tokens:

$$\Sigma := \Gamma \cup \{\texttt{ID}_1, \ldots, \texttt{ID}_n\} \cup \{\texttt{LYR}_1, \ldots, \texttt{LYR}_R\}. \tag{109}$$

Here n will be chosen as $\Theta(W)$, so $|\Sigma| = O(W + L)$. For each triple $(j, k, i) \in [n] \times [R] \times \{0, \ldots, m-1\}$, define

$$X_{j,k,i} := \texttt{ID}_j \parallel \texttt{LYR}_k \parallel \text{bin}_\ell(i) \parallel \#_{\text{in}}^{\mathbf{1}[i=0]} \in \Sigma^T, \qquad T := 2 + L_{\text{in}}. \tag{110}$$

Let $S := \{X_{j,k,i}\}$ and $\mathcal{X} := S$. Then $|S| = nRm$.

Common prefix. For each $X_{j,k,i}$, the generator runs the scratchpad construction of Theorem 8 on the suffix $\text{bin}_\ell(i) \parallel \#_{\text{in}}^{\mathbf{1}[i=0]}$ and ignores the two-token header while generating the first $T' - 1$ tokens. The header only shifts absolute scratchpad positions by

$$H := 2. \tag{111}$$

Thus all parameter settings share the same prefix assignment $R_S : S \to \Sigma^{T'-1}$: the DECODE segment, the universal TABLE segment, and PAD tokens, exactly as in Theorem 8. This prefix depends on i but not on the labeling.

Encoding labels. Fix any labeling $b \in \{0,1\}^S$, written as $b_{j,k,i}$. For each $(j,k) \in [n] \times [R]$, encode the m labels into

$$s_{j,k} := \sum_{i=0}^{m-1} b_{j,k,i} 2^i \in \{0, \dots, M-1\}, \qquad \mathrm{bit}_i(s_{j,k}) = b_{j,k,i}. \tag{112}$$

Multiplexing row indices. The embedding of $\mathtt{ID}_j$ stores the packed scalar

$$r_j^{(0)} := \sum_{u=1}^{R} \frac{s_{j,u}}{(2T')^{u-1}}, \tag{113}$$

and the embedding of $\mathtt{LYR}_k$ stores $g := k$. At the final generation step, a constant number of heads copies $(r_j^{(0)}, g)$ from the two header tokens into protected work coordinates. These coordinates are ignored before the final step, so they cannot affect R_S.

The recursive retrieval then proceeds for R rounds. At the beginning of round u, the register has the form

$$r^{(u-1)} = \sum_{v=u}^{R} \frac{s_{j,v}}{(2T')^{v-u}} = s_{j,u} + \delta_u, \qquad 0 \le \delta_u < \frac{1}{2}. \tag{114}$$

Using quadratic positional features, a hard-attention head with query $q = (2r^{(u-1)}, 1, 0)$ selects the unique visible position $t^\star = s_{j,u}$. The FFN performs the base-$2T'$ left shift

$$r^{(u)} := 2T'(r^{(u-1)} - t^\star). \tag{115}$$

To keep only the selected digit, use the usual ReLU hat gate $\mathrm{gate}_u(g) = \mathbf{1}[g = u]$ on integer $g \in [R]$ and update a protected accumulator by

$$\hat{s} \leftarrow \hat{s} + \Big(\mathrm{ReLU}(t^\star + M_0(\mathrm{gate}_u(g) - 1)) - \mathrm{ReLU}(M_0(\mathrm{gate}_u(g) - 1))\Big), \tag{116}$$

where $M_0 > T'$ is fixed. This adds $t^\star$ iff $u = k$ and adds zero otherwise. Hence after R rounds, $\hat{s} = s_{j,k}$. Set $\beta := L_{\mathrm{out}} \hat{s}$.

Final readout. At the final step, the prefix contains the complete scratchpad. Let v be the absolute position of the unique END0 token. Because of the header,

$$v = H + (i+1)L_{\mathrm{in}}. \tag{117}$$

A first readout attention sublayer retrieves this token and writes v into a work coordinate. The next FFN computes

$$r := v + \beta + \frac{v - H}{L_{\mathrm{in}}} = H + (i+1)L_{\mathrm{in}} + s_{j,k} L_{\mathrm{out}} + (i+1). \tag{118}$$

This is exactly the absolute position of the i-th bit in row $s_{j,k}$ of the TABLE segment. A second readout attention sublayer retrieves that token, whose visible bit is $\mathrm{bit}_i(s_{j,k}) = b_{j,k,i}$. The output layer maps visible bit $0/1$ to two answer tokens.

Shattering and parameter budget. The scratchpad subspace never reads the multiplexing registers, and the output rule for the first $T' - 1$ generated tokens depends only on the scratchpad subspace. Therefore every labeling produces the same prefix assignment R_S, and only the final token depends on $s_{j,k}$. Thus S is answer shattered and

$$\mathrm{ASdim}(\mathcal{F}_\star; \mathcal{X}, T') \geq |S| = nRm = \Omega(WL \log T_{\mathrm{ctx}}), \tag{119}$$

where $m = \Theta(\log T') = \Theta(\log T_{\mathrm{ctx}})$ because $T = O(\log \log T')$. The labeling-dependent parameters are the n packed ID embeddings, costing $O(n)$ parameters; the round-token embeddings and backbone cost $O(R) \subseteq O(L)$. Choosing $n = \Theta(W)$ and $W \geq C_0 L$ gives total parameter count at most W.

Sample-complexity lower bound. *Scratchpad witness.* The ASdim bound above and Theorem 6 give

$$\Omega\left(\frac{WL \log T_{\mathrm{ctx}}}{\varepsilon}\right) \tag{120}$$

samples.

Supervised ReLU witness via token prompts. We use the same fixed-prefix reduction, but realize the supervised inputs as token prompts rather than embedding-valued prompts. By the ReLU subclass from Lemma 16, after adjusting constants the Transformer family contains a ReLU-network subclass $\mathcal{H}$ with

$$\mathrm{VCdim}(\mathcal{H}) = \Omega(WL \log(W/L)). \tag{121}$$

Let $D := \lfloor c_{\mathrm{R}} WL \log(W/L) \rfloor$ and choose shattered points $a_1, \ldots, a_D$ for $\mathcal{H}$. The "sufficiently large T'" condition includes $T' \geq C_{\mathrm{R}} D$, so the prompt length below is at most a constant fraction of the total context length.

Let $\Sigma_{\mathrm{R}} = \{\texttt{blank}, \texttt{mark}, \texttt{query}, \#, 0, 1\}$. For each $i \in [D]$, define a token prompt $X_i \in \Sigma_{\mathrm{R}}^{D+1}$ with `mark` at position i, `query` at the last prompt position, and `blank` elsewhere. The positional encoding is fixed and non-learned; we set its first coordinates at position i to be the ReLU input a_i. Thus the inputs a_i are supplied by the fixed positional encoding and do not consume the W trainable parameters.

At the final generation step, a constant-size attention wrapper retrieves the unique `mark` token and copies these positional-encoding coordinates to the current output position. The remaining layers simulate the ReLU network $h \in \mathcal{H}$ on the copied vector. Before the final step, a positional gate forces the output to be the fixed dummy token $\#$; at the final step, the dummy branch is disabled and the classifier branch outputs 0 or 1. Hence each $h \in \mathcal{H}$ gives a token-prompt generator f_h satisfying

$$f_h^{(T')}(X_i)[-T' : -1] = \#^{T'-1}, \qquad f_h^{(T')}(X_i)[-1] = h(a_i). \tag{122}$$

The prefix is independent of h, so any CoT learner for this token-prompt subclass is also a supervised learner for $\mathcal{H}$: given labeled examples $(a_i, h(a_i))$, feed the learner the CoT examples $(X_i, \#^{T'-1} \,\|\, h(a_i))$ and read off its final-token predictor. The VC lower bound in Theorem 13 gives

$$\Omega\left(\frac{WL \log(W/L)}{\varepsilon}\right) \tag{123}$$

samples.

Taking the harder of these two witnesses,

$$\max\left\{\Omega\left(\frac{WL\log T_{\mathrm{ctx}}}{\varepsilon}\right),\Omega\left(\frac{WL\log(W/L)}{\varepsilon}\right)\right\}=\Omega\left(\frac{WL\log(T_{\mathrm{ctx}}W/L)}{\varepsilon}\right), \tag{124}$$

after adjusting constants. ■