# LIMIT ANALYSIS FOR SYMBOLIC MULTI-STEP REASONING TASKS WITH INFORMATION PROPAGATION RULES BASED ON TRANSFORMERS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Transformers are able to perform reasoning tasks, however the intrinsic mechanism remains widely open. In this paper we propose a set of information propagation rules based on Transformers and utilize symbolic reasoning tasks to theoretically analyze the limit reasoning steps. We show that the limit number of reasoning steps is between $2^{L-1}-1$ and $\frac{3^{L-1}-1}{2}$ for a model with $L$ attention layers in a single-pass. In addition, we find that, for problems with a fixed number of reasoning steps, Transformers exhibit a preference for generating sequences that satisfy certain ordering relations. Ultimately, we construct a Transformer and prove that the limiting reasoning depth is achievable.

## 1 INTRODUCTION

The transformer architecture introduced by (Vaswani et al., 2017) has demonstrated capabilities across a wide range of tasks (Liu et al., 2018; Devlin et al., 2019; Radford et al., 2019; Touvron et al., 2023; OpenAI, 2023) These models have reached performance levels comparable to top human contestants in the International Mathematical Olympiad (IMO) (Trinh et al., 2024). While using Chain-of-Thought (CoT) (Wei et al., 2022b) substantially improve multi-step logical reasoning accuracy by prompting models to generate explicit intermediate reasoning steps, they often exhibit an over-thinking phenomenon that consumes excessive computational resources and increases response time. This inefficiency highlights a critical question: what is the intrinsic single-pass reasoning capacity of these models?

In this work, in order to more intuitively understand the mechanism of multi-round information transmission in Transformer models and the boundary of information transmission capacity, we use a symbolic dataset that can simulate a class of multi-step reasoning to conduct our study(Wang et al., 2025a). We propose a set of information propagation rules that faithfully simulate how Transformers transmit information. We then rigorously analyze, under these propagation rules, the theoretical limit of the reasoning depth that can be achieved by an $L$-layer Transformer with a sufficiently large hidden dimension in a single forward pass. Our analysis shows that the achievable number of reasoning steps lies between $2^{L-1}-1$ and $\frac{3^{L-1}-1}{2}$. This result follows from two key structural properties of Transformers: (i) in each layer, tokens can execute reasoning operations in parallel; and (ii) each embedding vector in a hidden layer can encode information from multiple tokens in distinct sublinear subspaces.

Finally, we validate our theoretical analysis with numerical experiments, showing that a trained 3-layer Transformer with a sufficiently large hidden dimension can attain the theoretically predicted maximum reasoning depth and for tasks requiring a fixed number of reasoning steps, Transformers tend to favor output sequences that obey specific ordering relations.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

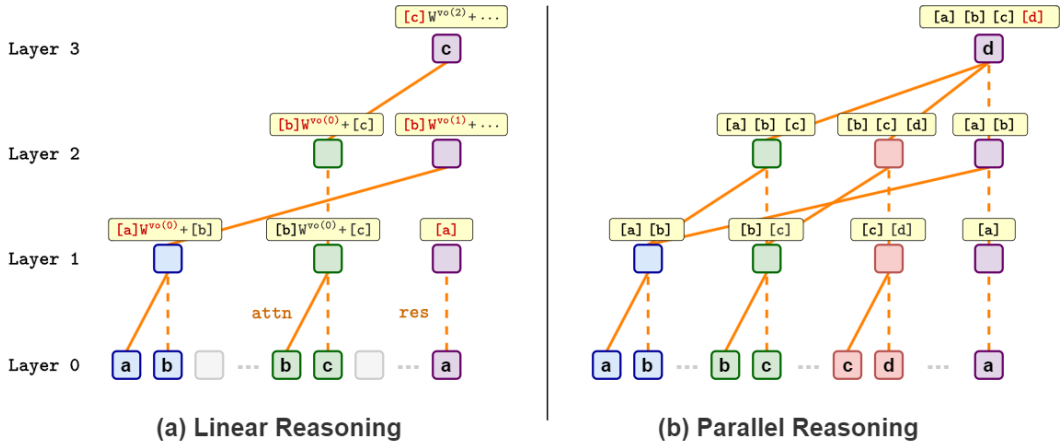



Figure 1: Illustration of linear reasoning and parallel reasoning.

## 2 Transformer and Reasoning Mechanism

### 2.1 Transformer Architecture

We investigate a decoder-only Transformer with $L$-layer attention blocks. For integer $n$, given any sequence $(x_i)_{1\leqslant i\leqslant n}$, we denote its one-hot encoding [1] as $\boldsymbol{X}^{\text{in}} \in \mathbb{R}^{n\times d}$ with $d$ as the dictionary size.

The model first applies an embedding layer including both token embedding and positional encoding to obtain the input representation as $\boldsymbol{X}^{(0)} = \boldsymbol{X}^{\text{emb}} + \boldsymbol{X}^{\text{pos}} \in \mathbb{R}^{n\times d_m}$. Moreover, we denote the set of word embeddings of each word in the dictionary as $W^E$. We shall use the single-head attention in each layer which is computed as follows:

$$\mathcal{A}^{(l)}(\boldsymbol{X}^{(l)}) = \text{SoftMax}\left(\frac{\text{mask}(\boldsymbol{X}^{(l)}\boldsymbol{W}^{q(l)}\boldsymbol{W}^{k(l),\mathsf{T}}\boldsymbol{X}^{(l),\mathsf{T}})}{\sqrt{d_k}}\right),$$
$$\boldsymbol{X}^{\text{qkv}(l)} = \mathcal{A}^{(l)}(\boldsymbol{X}^{(l)})\boldsymbol{X}^{(l)}\boldsymbol{W}^{v(l)}\boldsymbol{W}^{o(l)},$$

where $0 \leqslant l \leqslant L$ and $\tilde{\sigma}$ is the softmax operator. For simplicity of expression, we will abbreviate $\boldsymbol{W}^{q(l)}\boldsymbol{W}^{k(l),\mathsf{T}}$ as $\boldsymbol{W}^{qk(l)}$ and $\boldsymbol{W}^{v(l)}\boldsymbol{W}^{o(l),\mathsf{T}}$ as $\boldsymbol{W}^{vo(l)}$ in the following text. Also, we ignore the normalization coefficient $\sqrt{d_k}$ in later sections for notational simplicity. The output of the $(l+1)$-th layer is obtained as:

$$\boldsymbol{X}^{\text{ao}(l)} = \boldsymbol{X}^{(l)} + \boldsymbol{X}^{\text{qkv}(l)},$$
$$\boldsymbol{X}^{(l+1)} = \text{RMSNorm}\big(f^{(l)}(\boldsymbol{X}^{\text{ao}(l)}) + \boldsymbol{X}^{\text{ao}(l)}\big).$$

where $f^{(l)}(\cdot)$ represents the feedforward neural network of the $(l+1)$-th layer, with the same parameters at every position. Proposition D.7 establishes this sharing for the construction. The final output (in the form of token indices within the vocabulary) is obtained as:

$$\boldsymbol{Y} = \text{SoftMax}(\boldsymbol{X}_n^{(L)}\boldsymbol{W}^p) \in \mathbb{R}^d.$$

### 2.2 Induction Reasoning Mechanism

Based on numerous works on In-Context Learning, Induction Heads (Brown et al., 2020; Garg et al., 2022; Bietti et al., 2024; Nichani et al., 2024), and recent studies on multi-step reasoning (Wang et al., 2025a; Yu et al., 2025), the reasoning capability of Transformers can be largely attributed to a mechanism called the Buffer Mechanism for storing diverse information, together with adjacent position matching and same token matching for achieving information matching and transmission.

**Buffer Mechanism** The Buffer Mechanism is a crucial way for Transformers to store multiple pieces of information (Wang et al., 2025a). Specifically, the interaction of information among tokens in a

[1] One-hot encoding is a technique that represents categorical data as binary vectors, where only one bit is set to 1 others are set to 0.

Transformer occurs in the attention module. Figure 1(a) illustrates the information flow of a 3-layer model performing 2-step reasoning, i.e., given a sentence of the form "...[a][b]...[b][c]...[a]" the model is required to output [c]. The dashed lines denote residual connections, while the solid lines denote information propagation induced by the attention mechanism. When a token (e.g., [b]) attends to a previous token (e.g., [a]), its next-layer state is not simply [a]+[b], but rather $\texttt{[a]}\boldsymbol{W}^{vo} + \texttt{[b]}$. In other words, the Transformer stores the two pieces of information into subspaces spanned by different matrices through a linear transformation.

**Adjacent Position Matching** Similar to humans, language models rely heavily on the immediately preceding word when predicting the next word (Barbero et al., 2024). That is, the model can leverage positional encodings to establish connections between adjacent tokens. In fact, constructing such an attention weight matrix is not difficult. Assuming the positional encodings approximately satisfy $\boldsymbol{p}_i^\mathsf{T}\boldsymbol{p}_i = 1, \boldsymbol{p}_i^\mathsf{T}\boldsymbol{p}_j = 0, i \neq j$, it suffices to construct:

$$\boldsymbol{W}^{qk} = \sum_{i=1}^{[l_{\text{seq}}/2]} \boldsymbol{p}_{2i}\boldsymbol{p}_{2i-1}^\mathsf{T}, \tag{1}$$

$$(x_{2i} + \boldsymbol{p}_{2i})\boldsymbol{W}^{qk}(x_{2i-1} + \boldsymbol{p}_{2i-1})^\mathsf{T} \approx 1, \tag{2}$$

Clearly, by this method, we can construct attention between any adjacent tokens of fixed length. However, due to the inherent diversity of language tasks, only the attention between the most adjacent pair is the most salient. We refer to this mechanism as adjacent position matching.

**Same Token Matching** Same token matching is the most essential mechanism within induction heads. Its existence grants Transformers strong out-of-distribution generalization ability. As shown in the Figure 1(a), because both nodes in the first layer contain the same information [a], they can attend to each other via the same token matching mechanism. Specifically, it suffices that the weight matrices satisfy $\boldsymbol{W}^{qk(1)}\boldsymbol{W}^{vo(0),\mathsf{T}} = I$, in which case

$$\begin{aligned} q(\texttt{[a]})k(\texttt{[a]}\boldsymbol{W}^{vo(0)} + \texttt{[b]})^\mathsf{T} &\approx \texttt{[a]}\boldsymbol{W}^{qk(1)}\boldsymbol{W}^{vo(0),\mathsf{T}}\texttt{[a]}^\mathsf{T} \\ &= \texttt{[a]}\texttt{[a]}^\mathsf{T} \approx 1, \end{aligned} \tag{3}$$

That is, the final token node will allocate nearly all of its attention to the previous node containing the same information [a], thereby transmitting [b] to the final node in the next layer. In this way, a single-step reasoning is achieved. Multi-step reasoning follows the same principle.

### 2.3 Parallel Reasoning

However, we note that the above-described mode, where each layer performs only one step of reasoning, is far from the upper limit of the Transformer model. As shown in Figure 1(b), adjacent position matching and same token matching can occur multiple times within a single layer, thereby enabling even shallow Transformer models to perform multi-step reasoning. We refer to this phenomenon as parallel reasoning. The central question considered in this paper is: given only adjacent position matching and same token matching, what are the upper and lower bounds of the parallel reasoning step that a transformer with $L$ layers attention blocks can perform?

## 3 Informal Theorems

To investigate the above question, we first consider the simplest case in which all reasoning relations are arranged sequentially. As shown in Figure 2(a), the information flow of reasoning in this setting exhibits a clear "binary tree" structure. It then follows directly that the reasoning steps scale as $O(2^{L-1})$. In what follows, we will provide a rigorous proof of this result by mathematical induction. We note that permuting the order of reasoning pairs within the sequential arrangement does not disrupt the flow of sequential reasoning. Hence, when logical relations are arranged in sequence, the reasoning steps of a Transformer constitute the lower bound among all possible cases.

On the other hand, we observe that for a 3-layer model, when the data are arranged as illustrated in Figure 2(b), the final layer carries the maximum amount of information. The data in this case exhibit an evident fractal structure. The advantage of such a configuration is that each local terminal node can simultaneously match two preceding nodes by leveraging both the maximum and minimum

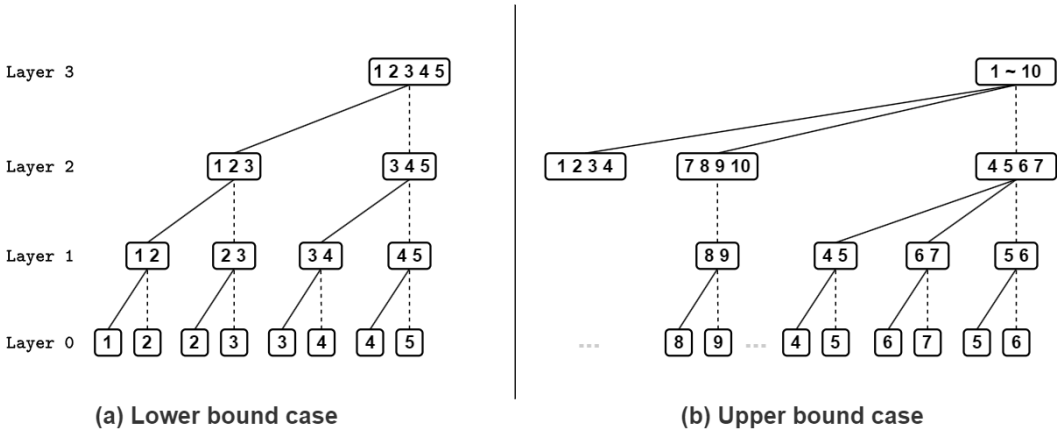



Figure 2: Example of lower bound and upper bound of parallel reasoning.

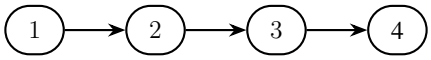



Figure 3: Three steps reasoning task. One step reasoning leads to 2, two steps reasoning leads to 3.

information it carries, thereby expanding its information content. Consequently, the reasoning steps scale as $O(3^{L-1})$. Therefore, we arrive at the following informal conclusion:

**Theorem 3.1** (Informal Corollary 6.4)**.** *The maximal number of reasoning steps a transformer with $L$ layers attention blocks can perform has a lower bound $O(2^{L-1})$ and an upper bound $O(3^{L-1})$.*

Next, we will provide a formal statement of the problem and a rigorous proof of the conclusion.

## 4 Symbolic Reasoning Task

In this section, we give a brief introduction to the reasoning task and some related definitions. Moreover, we shall introduce the rules of information propagation.

A reasoning task typically involves a question and an answer to that question, along with the rule and process to get the answer. For example, given $A_1 \subseteq A_2$ and $A_2 \subseteq A_3$, the question is the relation of $A_1$ and $A_3$, and the answer is $A_1 \subseteq A_3$. We shall use a more symbolic way to express reasoning tasks as in the following example in figure 3.

We use a sequence $(1, 2, 2, 3, 3, 4)$ to denote this reasoning task. Indeed, this sequence is composed of three bigrams $(1, 2)$, $(2, 3)$, $(3, 4)$, and each bigram represents one step of reasoning. We call these bigrams reasoning pairs, which we will define below.

**Definition 4.1.** *A reasoning pair is an element in $\mathbb{Z}^2$ of the form $\boldsymbol{a}_i = (\boldsymbol{a}_i^1, \boldsymbol{a}_i^2)$ where $i \in \mathbb{Z}$, $\boldsymbol{a}_i^1,\ \boldsymbol{a}_i^2 \in \mathbb{Z}$ and $\boldsymbol{a}_i^1 \neq \boldsymbol{a}_i^2$.*

The set of all reasoning pairs is denoted as $\mathcal{A}$. $a_i^1 \to a_i^2$ represents one step of reasoning.

It is natural that we shall define the $s$ step reasoning chain as follows.

**Definition 4.2.** *An $s$ step reasoning chain is a finite sequence $(\boldsymbol{a}_i)_{1 \leqslant i \leqslant s}$ with $\boldsymbol{a}_i \in \mathcal{A}$, and it shall satisfy the following conditions:*

- $\boldsymbol{a}_i^2 = \boldsymbol{a}_{i+1}^1$ *for* $1 \leqslant i \leqslant s-1$*;*
- *For any subsequence $(\boldsymbol{a}_{i_m})_{i_m \in \mathcal{I}}$ of $(\boldsymbol{a}_i)_{1 \leqslant i \leqslant s}$, we have $\boldsymbol{a}^1_{\min\{\mathcal{I}\}} \neq \boldsymbol{a}^2_{\max\{\mathcal{I}\}}$ (no loop), where $\mathcal{I}$ is a subset of $\{1, 2, \cdots, s\}$ containing at least two elements.*

The first condition ensures that the reasoning chain does not break before the final step, and the second condition ensures that there is no loop of arbitrary size in the reasoning task. For example, the sequence $((1, 2), (2, 3), (3, 1))$ and $((1, 2), (3, 4), (4, 5))$ are not reasoning chains.

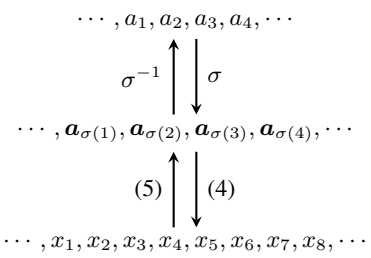



Figure 4: Relationship between reasoning chain and reasoning sequence.

Similarly, the subscript $i$ in this definition can take values from an infinite set like $\mathbb{Z}$, thereby defining an infinite chain of reasoning.

**Remark 4.3.** *For notational simplicity, here and in the sequel, we shall write $(\boldsymbol{a}_i)$ for $(\boldsymbol{a}_i)_{i\in I}$ when the index set $I$ is clear from the context.*

In practice, a sentence may consist of reasoning pairs which are not in order. Due to the mask condition which is common in the LLM, the order of reasoning pairs may influence the information propagation. To describe the order of these reasoning pairs and their relation to the reasoning chain, we define the elements of a symmetric group as permutations, and focus on the symmetric group $\mathrm{Sym}(\mathbb{Z})$.

**Definition 4.4.** *Given a reasoning chain $(\boldsymbol{a}_m)_{m\in\mathbb{Z}}$ and a permutation $\sigma \in \mathrm{Sym}(\mathbb{Z})$, a sequence $(x_i)_{i\in\mathbb{Z}}$ is called a reasoning sequence constructed from $(\boldsymbol{a}_m)_{m\in\mathbb{Z}}$ and $\sigma$ if it satisfies:*

$$x_i = \boldsymbol{a}^{2-(i \bmod 2)}_{\sigma(\lfloor \frac{i+1}{2} \rfloor)}. \tag{4}$$

*Also, $(\boldsymbol{a}_m)_{m\in\mathbb{Z}}$ and $\sigma$ are called the constructing reasoning sequence and constructing permutation of $(x_i)$, respectively.*

When referring to a reasoning sequence $(x_i)$, we are actually denoting the tuple $((x_i), (\boldsymbol{a}_m), \sigma)$. Moreover, if $\sigma = \mathrm{Id}$, then the reasoning sequence is called a sorted reasoning sequence. Note that from the relation (4) we also have

$$\boldsymbol{a}_i = \boldsymbol{a}_{\sigma(\sigma^{-1}(i))} = (x_{2\sigma^{-1}(i)-1}, x_{2\sigma^{-1}(i)}), \tag{5}$$

where $\sigma^{-1}$ is the inverse of $\sigma$ satisfying $\sigma \circ \sigma^{-1} = \sigma^{-1} \circ \sigma = \mathrm{Id} \in \mathrm{Sym}(\mathbb{Z})$.

Likewise, in the case where the index ranges over a finite set, the reasoning sequence will be of finite length. Examples of sorted and permuted reasoning sequences are provided in Appendix F.1.

When considering a finite step reasoning sequence, the concept of reasoning start, which indicates where the reasoning task should begin, is also needed. More specifically, we consider an $s$ step reasoning sequence $(x_i)_{1\leqslant i\leqslant 2s}$ with constructing reasoning chain $(\boldsymbol{a}_m)_{1\leqslant m\leqslant s}$ and constructing permutation $\sigma$. Then one more element $x_{2s+1}$ is added to the end of the reasoning sequence $(x_i)$, and $x_{2s+1} = \boldsymbol{a}^1_{m_0}$ for some $\boldsymbol{a}_{m_0} \in (\boldsymbol{a}_m)$.

An example with a reasoning start is provided in Appendix F.1.

Next we define nodes which serves as containers of information.

**Definition 4.5.** *A node corresponding to a reasoning sequence $(x_i)$ is a set of two sets. More specifically, for $i \in \mathbb{N}$, an $l$th layer node is defined as $N^l_i = \{V^l_i, I^l_i\}$ where $V^l_i$ is called a value set whose elements are integers, and $I^l_i \subseteq \mathbb{Z}$ is an index set. Also, we require that $V^l_i = \bigcup_{i_\alpha \in I^l_i} \{x_{i_\alpha}\}$.*

We define the information quantity of a node $N^l_i = \{V^l_i, I^l_i\}$ as $C^l_i = |V^l_i|$. Moreover, we denote $\mathcal{N}^l$ as the set of all $l$th layer nodes.

We denote the information propagation between two nodes as $N^{l+1}_i = N^l_m \star N^l_i$ where $N^l_m \star N^l_i := \{V^l_m \cup V^{l+1}_i, I^l_m \cup I^{l+1}_i\}$. The case $i \neq m$ represents the attention mechanism, more specifically,

the node $N_m^l$ attends to the node $N_i^l$ and the result is stored in the node $N_i^{l+1}$. The case $i = m$ represents the residual connection, in which case $N_i^{l+1} = N_i^l$. Moreover, there may be more than one node transmitting information to a node $N_i^l$. In this case, we denote the set of all such nodes as $N_{\mathcal{I}}^l \subseteq \mathcal{N}^l$ where $\mathcal{I}$ is some index set. The information propagation process in this case is then defined by $N_i^{l+1} = N_{\mathcal{I}}^l \star N_i^l := \{\bigcup_{i_\alpha \in \mathcal{I}} V_{i_\alpha}^l \cup V_i^{l+1}, \bigcup_{i_\alpha \in \mathcal{I}} I_{i_\alpha}^l \cup I_i^{l+1}\}$.

## 5 Rules of information propagation

We can extract the following rules of information propagation from the behavior of transformer as follows.

- Rule 0 (Initial setup): The nodes in 0th layer are constructed as $N_i^0 = \{\{x_i\}, \{i\}\}$. For $l \geqslant 1$, the $l$th layer of nodes are initially constructed as $N_i^l = \{\emptyset, \emptyset\}$.
- Rule 1 (Mask Condition): Attention happens only from former nodes to later nodes. That is, the attention mechanism $N_i^{l+1} = N_m^l \star N_i^l$ is performed only when $m < i$. For the multiple nodes information transmission case, the operation $N_i^{l+1} = N_{\mathcal{I}}^l \star N_i^l$ is performed only when $i_\alpha < i$ for all $i_\alpha \in \mathcal{I}$.
- Rule 2 (Adjacent position matching): For $l = 1$, the information in an odd position node can be transmitted to the subsequent even position node. In this case, the mask condition is satisfied automatically. More specifically, the nodes in 1st layer are of the form $N_{2i}^1 = N_{2i-1}^0 \star N_{2i}^0$ after position matching.
- Rule 3 (Same token matching): For $l \geqslant 2$, a node $N_i^l$ is updated as $N_i^l = N_{\mathcal{I}}^{l-1} \star N_i^{l-1}$ provided there exists a set $N_{\mathcal{I}}^{l-1} \subseteq \mathcal{N}^{l-1}$ satisfying the mask condition $i_\alpha < i$ and $V_{i_\alpha}^{l-1} \cap V_i^{l-1} \neq \emptyset$ for all $i_\alpha \in \mathcal{I}$.
- Rule 4 (Residual Connection): For $l \geqslant 1$, $\forall N_i^l \in \mathcal{N}^l$, $N_i^l = \{V_i^{l-1} \cup V_i^l, I_i^{l-1} \cup I_i^l\}$.

**Remark 5.1.** *With a slight abuse of notation, we still denote the value set and index set of a node $N_i^l$ as $V_i^l$ and $I_i^l$ after the information propagation process. For example, through residual connection $N_i^l$ is updated as $N_i^l = \{V_i^{l-1} \cup V_i^l, I_i^{l-1} \cup I_i^l\}$, and we still denote the sets $V_i^{l-1} \cup V_i^l$ and $I_i^{l-1} \cup I_i^l$ as $V_i^l$ and $I_i^l$, since we only care about the result after each layer's information propagation.*

**Remark 5.2.** *It makes no difference whether the residual connection happens before or after same token matching or position match. The result stored in the next layer remains unchanged.*

The detailed discussion of alternative matching orders and a worked example are provided in Appendix F.2.

Two concepts of layer arise in this framework: the layer of attention blocks and the layer of nodes. The $l$th layer attention block takes $(l-1)$th layer of nodes as input and produces the $l$th layer of nodes as output. Due to this relation, we use "layer $l$" to refer to the $l$th layer of attention blocks and the $l$th layer of nodes interchangeably.

## 6 Main Theorems

In this section we analyze the information quantity in the process of information propagation according to the above information propagation rules, and our main theorem follows.

**Theorem 6.1.** *Under the rules of information propagation, given any reasoning sequence $((x_i)_{i\in\mathbb{Z}}, (\boldsymbol{a}_m)_{m\in\mathbb{Z}}, \sigma)$, for any given $x \in \{x_i\}_{i\in\mathbb{Z}}$, and for any $l \in \mathbb{Z}^+$ there exists $i \in \mathbb{Z}^+$ such that $x \in V_i^l$, and we have the following bound for $T^l(x) = \max_{i\in\mathbb{Z}}\{C_i^l \mid x \in V_i^l\}$:*

$$2^{l-1} + 1 \leqslant T^l(x) \leqslant 3^{l-1} + 1. \tag{6}$$

The whole proof is based on mathematical induction. Here we only give a sketch of the proof. The complete proof can be found in the Appendix A.

Given a reasoning sequence $((x_i),(a_m),\sigma)$, when considering the lower bound, by Rule 2 the value sets of nodes in layer 1 contain only one reasoning pair except the case where only residual connection happens. Suppose that $j<i$ and the node $N_i^1$ contains $\boldsymbol{a}_{m_i}^1, \boldsymbol{a}_{m_i}^2$ as value set, or simply we say $N_i^1$ contains $\boldsymbol{a}_{m_i}$, and $N_j^1$ contains $\boldsymbol{a}_{m_j}$. Two cases may happen, $n_i+1=n_j$ and $\boldsymbol{a}_{m_i}^2=\boldsymbol{a}_{m_j}^1$ or $n_j+1=n_i$ and $\boldsymbol{a}_{m_j}^2=\boldsymbol{a}_{m_i}^1$. Both cases will lead to $N_i^2$ containing $\boldsymbol{a}_{m_i}$ and $\boldsymbol{a}_{m_j}$ by Rule 3. This process is the same for any other two nodes containing two adjacent reasoning pairs respectively. The process for layer 3 is analogous to that for layer 2: information contained in two nodes in layer 2 is propagated to one node in layer 3. The whole structure is in fact a binary tree and hence the bound is powers of 2.

Regarding the upper bound, due to the mask condition (Rule 1), the permutation $\sigma$ may affect the information propagation. However, the upper bound is always bounded by the case when the mask condition is lifted. Therefore, we ignore the mask condition to find the upper bound. Just like the proof of the lower bound which use two adjacent reasoning pairs, now we use three adjacent reasoning pairs. Suppose the nodes $N_i^1$, $N_j^1$ and $N_k^1$ contain the reasoning pairs $\boldsymbol{a}_{m_i}$, $\boldsymbol{a}_{m_j}$ and $\boldsymbol{a}_{m_k}$ respectively and $m_i+1=m_j=m_k-1$. Then by Rule 3, at least one of the node $N_j^2$ contains $\boldsymbol{a}_{m_i}$, $\boldsymbol{a}_{m_j}$ and $\boldsymbol{a}_{m_k}$. As for layer 3, there are nodes that contain information propagated from three nodes like $N_j^3$. The whole structure is a ternary tree and hence the bound is powers of 3.

**Theorem 6.2.** *Under the rules of information propagation* (5)*, for any $s$ step reasoning sequence $((x_i)_{1\leqslant i\leqslant 2s},(\boldsymbol{a}_m)_{1\leqslant m\leqslant s},\sigma)$ with reasoning start $x_{2s+1}$, for any $l$ satisfying that $1\leqslant l\leqslant 1+\log_2 s$, we have the following estimate for $C_{2s+1}^l$.*

$$2^{l-1}\leqslant C_{2s+1}^l\leqslant 3^{l-1}. \tag{7}$$

The proof of this theorem is similar to the proof of Theorem 6.1 with two main differences. First, we need to take into account the reasoning start which requires using mathematical induction twice: once on the node of reasoning start and once on the nodes in the sequence. Second, the reasoning sequence is now finite, and we require it to be long enough to support the structure of the binary tree and the ternary tree. The complete proof is included in the Appendix B.

**Remark 6.3.** *In the proof of this theorem, the mask condition was relaxed to obtain the upper bound. However, we showed in Appendix C that there is a way to construct a large class of reasoning sequences such that the upper bound is attained under mask condition. This proves that the upper bound is indeed tight.*

**Corollary 6.4.** *For a given transformer with $L$ layers of attention blocks and an input sequence of $n=2s+1$, where $s\in\mathbb{Z}^+$ and $1\leqslant L\leqslant 1+\log_2 s$, the maximal number of reasoning steps $S_p$ it can perform satisfies the following bounds:*

$$2^{L-1}-1\leqslant S_p\leqslant \frac{3^{L-1}-1}{2}. \tag{8}$$

**Remark 6.5.** *Although the upper bound on information quantity as we show in Theorem 6.2 is $3^{L-1}$, this only implies that up to $3^{L-1}-1$ reasoning steps are involved. However, since the process does not track information back through the reasoning chain, only $\frac{3^{L-1}-1}{2}$ reasoning steps are effective in the reasoning tasks. A quick example is the sequence $(0,1,1,2,1)$ with reasoning start $1$. For $l=2$ there are two reasoning steps in total but only one effective reasoning step $1\to 2$.*

## 7 Experiments and Discussions

### 7.1 Training task

To verify whether the Transformer model can achieve the expected reasoning capability, we conducted training specifically on Transformer architectures. As the number of layers increases, both the required amount of reasoning and the model capacity grow exponentially, leading to significant optimization difficulty. Therefore, we primarily focus on training a 3-layer Transformer model.

Our task aligns with the reasoning sequence described earlier. Specifically, we begin with a reasoning sequence denoted as $(x_i)_{1\leq i\leq 2s}$. Subsequently, we introduce $x_{2s+1}$ as the starting point. The complete sequence $(x_i)_{1\leq i\leq 2s+1}$ serves as the input to the transformer. The output corresponds to the reasoning result from the starting point with a fixed reasoning step (an example in Fig. 5).




Figure 5: A two-step reasoning example.

The detailed training settings are provided in the Appendix E. Reasoning pairs in the test set are totally different from the training set in order to prove transformer can learn reasoning instead of remembering these sequences.

Firstly, this model is capable of solving 3-step reasoning problems with perfect accuracy as predicted by our theoretical analysis. We can see that such a model can be successfully trained in the Appendix E.3.

Then, We investigate scenarios where the model exhibits partial or complete failure. According to the theoretical analysis presented earlier Corollary 6.4, for reasoning step lengths in the interval $[2^{L-1}, \frac{3^{L-1}-1}{2}]$, the model can produce correct answers under certain sequence ordering conditions. However, when the number of steps exceeds $\frac{3^{L-1}-1}{2}$, information propagation to the final token becomes insufficient, resulting in incorrect answers. Since this implies that high accuracy is unattainable in such regimes, experimental validation remains partial.

We consider a 3-layer Transformer. Given that $2^{3-1} = 4$ and $\frac{3^{3-1}-1}{2} = 4$, the model can solve 4-step reasoning tasks when sequence order conditions are satisfied, but fails for 5-step reasoning. Experimental results under these settings are as follows:

- For 4-step reasoning, the model achieves a test accuracy of 46.1%.
- For 5-step reasoning, the test accuracy drops to 25.1%.

The decrease in accuracy for the 5-step case provides empirical support for theoretical results.

For clarity, we divide the 4-step reasoning data into two categories according to the sequence order structure predicted by our theory:

- Canonical sequence: In the 4-step reasoning task, a sentence in which the 3rd fact appears after the 2nd and 4th facts.
- Non-canonical sequence: A sentence that does not satisfy the above condition.

According to our theoretical analysis, the model is expected to correctly resolve canonical sequences, whereas for non-canonical sequences it is unable to accumulate the necessary information and therefore effectively resorts to guessing. It can be seen that the Transformer exhibits a strong preference for canonical sequences, achieving approximately 90% accuracy on canonical inputs but only around 20% accuracy on non-canonical inputs, as shown in Figure 6. Additional detailed experimental verification can be seen in Appendix E.5 and Appendix E.6.

Complete training curves for 4-step reasoning tasks are provided in Appendix E.5.

To demonstrate the generality of our theory, we verify it on a 2-layer Transformer in Appendix E.7.

## 8 CONSTRUCTION OF THE TRANSFORMER

To demonstrate that a Transformer can indeed perform reasoning at the limiting depth, we construct in the Appendix D a model that stores the reasoning sequence using circulant matrices and is capable of achieving the aforementioned limiting reasoning depth.

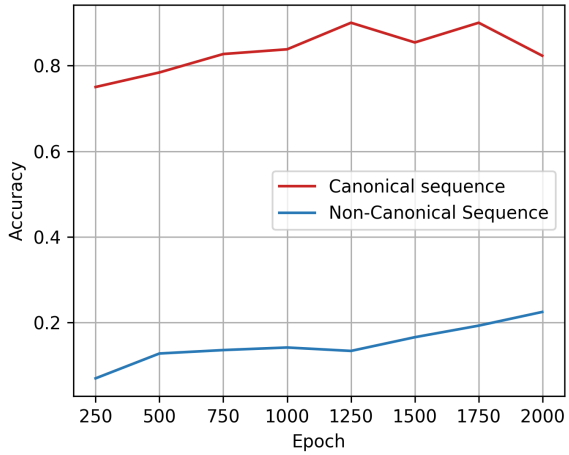



Figure 6: Test accuracy of sentences with different sequence relationships

## 9 Related Work

The ability of Transformers to perform complex reasoning is supported by theoretical work on their expressiveness. Prior research has shown that Transformers are Turing-complete in theory (Pérez et al., 2019) and are universal approximators of sequence-to-sequence functions (Yun et al., 2019). However, there is a known gap between this theoretical capability and what Transformers can reliably learn from data (Wei et al., 2022a), and their representational power has specific strengths and limitations Sanford et al. (2023).

To study multi-step reasoning in practice, various reasoning tasks have been proposed, such as recognizing context-free grammars (Zhao et al., 2023), matrix digits task (Webb et al., 2023), SET game tasks (Altabaa et al., 2023), reasoning tasks designed by anchor function (Zhang et al., 2024b). Kil et al. (2024); Li et al. (2024a) use Chain-of-Thought (CoT) reasoning (Wei et al., 2022b) to achieve multi-step reasoning via prompt engineering. Zhang et al. (2023) introduced the beam retrieval framework for multi-hop QA improving the few-shot QA performance of LLMs. Li et al. (2024b); Yang et al. (2024); Shalev et al. (2024) locate the potential intermediate answers within middle layers which play a causative role in shaping the final explicit reasoning results. Wang et al. (2025a) use a symbolic dataset to study the mechanism of multi-step reasoning in Transformer models. Their work focuses on serial reasoning, where each layer performs one step of reasoning. As an extension of that work, we adopt the same dataset setting, but investigate the mechanism of parallel reasoning. Zhang et al. (2024a); Yao et al. (2025) show that with small initialization, Transformers in condense regime (Luo et al., 2021; Xu et al., 2025) can learn reasoning better. Wang et al. (2025b) investigated the $k$-fold task which is similar to the k-hop induction head task in Sanford et al. (2024).

Sanford et al. (2024) study the asymptotic depth complexity of the $k$-hop induction task. They construct a causally masked, single-head Transformer with constant width and depth $\lfloor \log_2 k \rfloor + 2$, and establish its asymptotic optimality through an $\Omega(\log k)$ depth lower bound under an MPC conjecture and model-size constraints. Their model permits arbitrary token-wise query/key/value maps and input/output MLPs, and their experiments examine depth scaling and attention patterns. In their task (their Definition 4.1), each successive hop searches the prefix ending at the position reached by the preceding hop; reordering the input can therefore change the target answer itself. In our task, permuting the same reasoning pairs preserves the target answer for a fixed query and reasoning length. We analyze how these permutations affect parallel information aggregation and the attainability of explicit upper and lower bounds on reasoning steps at a fixed depth $L$ under the proposed information propagation rules, and investigate the corresponding behavior in trained Transformers.

### AI USE STATEMENT

During the preparation of this manuscript, we used DeepSeek-V3 (July 2024) to assist with language polishing (grammar, wording). All substantive content, ideas, and core writing remain the authors' own, and the authors are fully responsible for any error including those introduced during polishing. OpenAI Codex was used to adapt the LaTeX formatting to the ICLR 2027 template and integrate the supplementary material as appendices.

## REFERENCES


Awni Altabaa, Taylor Webb, Jonathan Cohen, and John Lafferty. Abstractors: Transformer modules for symbolic message passing and relational reasoning. *arXiv preprint arXiv*, 2304, 2023.

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. Round and round we go! what makes rotary positional encodings useful? *arXiv preprint arXiv:2410.06205*, 2024.

Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. *Advances in Neural Information Processing Systems*, 36, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598, 2022.

Jihyung Kil, Farideh Tavazoee, Dongyeop Kang, and Joo-Kyung Kim. Ii-mmr: Identifying and improving multi-modal multi-hop reasoning in visual question answering. *arXiv preprint arXiv:2402.11058*, 2024.

Yanyang Li, Shuo Liang, Michael R Lyu, and Liwei Wang. Making long-context language models better multi-hop reasoners. *arXiv preprint arXiv:2408.03246*, 2024a.

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. Understanding and patching compositional reasoning in llms. *arXiv preprint arXiv:2402.14328*, 2024b.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

Tao Luo, Zhi-Qin John Xu, Zheng Ma, and Yaoyu Zhang. Phase diagram for two-layer relu neural networks at infinite-width limit. *Journal of Machine Learning Research*, 22(71):1–47, 2021.

Eshaan Nichani, Alex Damian, and Jason D Lee. How transformers learn causal structure with gradient descent. *arXiv preprint arXiv:2402.14735*, 2024.

R OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5):1, 2023.

Jorge Pérez, Javier Marinković, and Pablo Barceló. On the turing completeness of modern neural network architectures. *arXiv preprint arXiv:1901.03429*, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Clayton Sanford, Daniel J Hsu, and Matus Telgarsky. Representational strengths and limitations of transformers. *Advances in Neural Information Processing Systems*, 36:36677–36707, 2023.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43276–43327. PMLR, 2024. URL https://proceedings.mlr.press/v235/sanford24a.html.

Yuval Shalev, Amir Feder, and Ariel Goldstein. Distributional reasoning in llms: Parallel reasoning processes in multi-hop reasoning. *arXiv preprint arXiv:2406.13858*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Zhiwei Wang, Yunji Wang, Zhongwang Zhang, Zhangchen Zhou, Hui Jin, Tianyang Hu, Jiacheng Sun, Zhenguo Li, Yaoyu Zhang, and Zhi-Qin John Xu. Understanding the language model to solve the symbolic multi-step reasoning problem from the perspective of buffer mechanism, 2025a. URL https://arxiv.org/abs/2405.15302.

Zixuan Wang, Eshaan Nichani, Alberto Bietti, Alex Damian, Daniel Hsu, Jason D. Lee, and Denny Wu. Learning compositional functions with transformers from easy-to-hard data, 2025b. URL https://arxiv.org/abs/2505.23683.

Taylor Webb, Keith J Holyoak, and Hongjing Lu. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541, 2023.

Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers. *Advances in Neural Information Processing Systems*, 35:12071–12083, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022b.

Zhi-Qin John Xu, Yaoyu Zhang, and Zhangchen Zhou. An overview of condensation phenomenon in deep learning. *arXiv preprint arXiv:2504.09484*, 2025.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? *arXiv preprint arXiv:2402.16837*, 2024.

Junjie Yao, Zhongwang Zhang, and Zhi-Qin John Xu. An analysis for reasoning bias of language models with small initialization. *Forty-second International Conference on Machine Learning*, 2025.

Zeping Yu, Yonatan Belinkov, and Sophia Ananiadou. Back attention: Understanding and enhancing multi-hop reasoning in large language models. *arXiv preprint arXiv:2502.10835*, 2025.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? *arXiv preprint arXiv:1912.10077*, 2019.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Yong Liu, and Shen Huang. Beam retrieval: General end-to-end retrieval for multi-hop question answering. *arXiv preprint arXiv:2308.08973*, 2023.

Zhongwang Zhang, Pengxiao Lin, Zhiwei Wang, Yaoyu Zhang, and Zhi-Qin John Xu. Initialization is critical to whether transformers fit composite functions by inference or memorizing. *arXiv preprint arXiv:2405.05409*, 2024a.

Zhongwang Zhang, Zhiwei Wang, Junjie Yao, Zhangchen Zhou, Xiaolong Li, Zhi-Qin John Xu, et al. Anchor function: a type of benchmark functions for studying language models. *arXiv preprint arXiv:2401.08309*, 2024b.

Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. Do transformers parse while predicting the masked word?, 2023. URL https://arxiv.org/abs/2303.08117.

# APPENDIX

## A PROOF OF THEOREM 6.1

We first prove the lower bound.

According to Rule 0, the nodes in first layer are constructed as $N_i^1 = \{(x_i), \{i\}, \{i\}\}$. According to Rule 2, we have the nodes in second layer are constructed as $N_{2i}^1 = \{\{x_{2i-1}, x_{2i}\}, \{2i\}, \{2i-1, 2i\}\}$ for $i \in \mathbb{Z}$. Clearly, we have $C_{2i}^1 = 2$.

We use mathematical induction to prove that for $k \geqslant 2$, for any $x_j \in \{x_i\}_{i\in\mathbb{Z}}$, there exists an integer $i \in \mathbb{Z}$ such that

- $x_j \in V_i^k$,
- We denote $\sigma(I_i^k)$ to be the set $\{\sigma(\lfloor \frac{i_m+1}{2} \rfloor)\}_{i_m \in I_i^k}$, and we have $\max \sigma(I_i^k) - \sigma(\lfloor \frac{j+1}{2} \rfloor) \geqslant 2^{k-1} - 1$ which implies that $C_i^k \geqslant 2^{k-1} + 1$.

For $k = 2$, given any $x_j \in (x_i)$, according to Rule 2, we know that $x_j$ is contained in the value set $V^1_{2\lfloor \frac{j+1}{2} \rfloor}$ of node $N^1_{2\lfloor \frac{j+1}{2} \rfloor}$.

For simplicity, we assume that $j$ is an even number, then $V^1_{2\lfloor \frac{j+1}{2} \rfloor} = V_j^1 = \{x_{j-1}, x_j\}$ and $I_j^1 = \{j-1, j\}$. We now want to find a node $N_2^i$ such that $V_i^1 \cap V_j^1 = \{x_j\}$. To do so, by definition of reasoning sequence and using the relation (4) we know that $x_{j-1}$ and $x_j$ comes from the reasoning pair $\boldsymbol{a}_{\sigma(\frac{j}{2})}$. By definition of reasoning sequence and relation (5) we have

$$\begin{aligned} \boldsymbol{a}^2_{\sigma(\frac{j}{2})} &= \boldsymbol{a}^1_{\sigma(\frac{j}{2})+1}, \\ \boldsymbol{a}_{\sigma(\frac{j}{2})+1} &= \left(x_{2\sigma^{-1}\left(\sigma(\frac{j}{2})+1\right)-1}, x_{2\sigma^{-1}\left(\sigma(\frac{j}{2})+1\right)}\right). \end{aligned} \tag{9}$$

Set $i = 2\sigma^{-1}\left(\sigma(\frac{j}{2}) + 1\right)$. It is now clear that the node $N_i^1$ have value set $V_i^1$ satisfying $V_i^1 \cap V_j^1 = \{x_j\}$ and index set $I_i^1 = \{i-1, i\}$. By Rule 3, the node $N^2_{\max\{i,j\}}$ have value set $V^2_{\max\{i,j\}}$ such that $\{x_{j-1}, x_j, x_{i+1}\} \subseteq V^2_{\max\{i,j\}}$. Hence, $C^2_{\max\{i,j\}} \geqslant 3$ and $\sigma(\frac{j}{2}) + 1 - \sigma(\frac{j}{2}) = 1 \geqslant 2^{2-1} - 1$.

Now we assume the induction hypotheses hold for $3 \leqslant k \leqslant k_0$ and consider the case $k = k_0 + 1$. There exists a node $N_{m_0}^{k_0}$ such that

$$x_j \in V_{m_0}^{k_0}, \tag{10}$$

$$\max \sigma(I_{m_0}^{k_0}) - \sigma(\frac{j}{2}) \geqslant 2^{k_0-1} - 1, \tag{11}$$

$$C_{m_0}^{k_0} \geqslant 2^{k_0-1} + 1. \tag{12}$$

Set $\alpha = \max \sigma(I_{m_0}^{k_0})$ then using relation (5) we know that $\boldsymbol{a}_\alpha^2 = x_{2\sigma^{-1}(\alpha)} \in V_{m_0}^{k_0}$. Using definition of reasoning sequence and relation (4) we know that

$$x_{2\sigma^{-1}(\alpha+1)-1} = \boldsymbol{a}^1_{\alpha+1} = \boldsymbol{a}^2_\alpha = x_{2\sigma^{-1}(\alpha)}. \tag{13}$$

Note that $\sigma(\lfloor \frac{2\sigma^{-1}(\alpha+1)-1+1}{2} \rfloor) = \alpha + 1$. And then the induction hypotheses implies that there is a node $N_{m_1}^{k_0}$ such that

$$x_{2\sigma^{-1}(\alpha+1)-1} \in V_{m_1}^{k_0}, \tag{14}$$

$$\max \sigma(I_{m_1}^{k_0}) - (\alpha + 1) \geqslant 2^{k_0-1} - 1, \tag{15}$$

$$C_{m_1}^{k_0} \geqslant 2^{k_0-1} + 1. \tag{16}$$

This construction ensures that $x_{2\sigma^{-1}(\alpha+1)-1} = x_{2\sigma^{-1}(\alpha)} \in V_{m_1}^{k_0} \cap V_{m_0}^{k_0}$, then by Rule 3, we know that the node $N^{k_0+1}_{\max\{m_0,m_1\}}$ have value set $V^{k_0+1}_{\max\{m_0,m_1\}}$ and index set $I^{k_0+1}_{\max\{m_0,m_1\}}$ satisfying that

$$x_j \in V^{k_0+1}_{\max\{m_0,m_1\}}, \tag{17}$$

$$I_{m_0}^{k_0} \cup I_{m_1}^{k_0} \subseteq I^{k_0+1}_{\max\{m_0,m_1\}}. \tag{18}$$

Combining (11), (15) and (18) leads to

$$\max \sigma(I^{k_0+1}_{\max\{m_0,m_1\}}) - \sigma(\frac{j}{2}) \geqslant 2^{k_0} - 1,$$

which implies that $C^{k_0+1}_{\max\{m_0,m_1\}} \geqslant 2^{k_0+1-1} + 1$. This completes the proof for lower bound.

Next we prove the upper bound.

It is clear that if Rule 1 is removed, the information quantity in each node can only maintain unchanged or increase. Therefore, we consider the no-mask condition, without loss of generality, we consider the reasoning sequence to be $\{x_i = \lfloor \frac{i}{2} \rfloor\}_{i\in\mathbb{N}}$ with constructing permutation $\sigma = \mathrm{Id}$ and constructing reasoning chain $(\boldsymbol{a}_m) = ((m-1, m))$.

We will use mathematical induction to prove that for $k \geqslant 2$ and $m \in \mathbb{Z}$, the value set of node $N^k_{2m}$ satisfies $V^k_{2m} \subseteq \{m - \frac{3^{k-1}+1}{2}, m - \frac{3^{k-1}+1}{2} + 1, \cdots, m + \frac{3^{k-1}-1}{2}\}$ and $C^k_{2m} \leqslant 3^{k-1} + 1$.

Since the index set of a node will play no rule in this proof, we will just ignore them in the expression of a node.

When $k = 2$, By Rule 2 the nodes $N^1_i$ are of the form $N^1_{2i} = \{\{x_{2i-1}, x_{2i}\}, \{2i\}\}$. Then by Rule 3 the nodes in layer 3 are of the form $N^2_{2i} = \{\{i-2, i-1, i, i+1\}\{2i\}\}$. Therefore, the conclusion holds for $k = 2$.

Assuming the conclusion holds for $k \leqslant k_0$. When $k = k_0 + 1$, by inductive hypotheses, there are three nodes $N^{k_0}_{2m}, N^{k_0}_{2m_1}, N^{k_0}_{2m_2}$ with value sets $V^{k_0}_{2m}, V^{k_0}_{2m_1}, V^{k_0}_{2m_2}$. We require that

$$\begin{aligned} m + \frac{3^{k_0-1}-1}{2} &= m_1 - \frac{3^{k_0-1}+1}{2}, \\ m_2 + \frac{3^{k_0-1}-1}{2} &= m - \frac{3^{k_0-1}+1}{2}. \end{aligned} \tag{19}$$

Simple calculation shows that

$$m_1 = m + 3^{k_0-1},\ m_2 = m - 3^{k_0-1}. \tag{20}$$

And the three nodes $N^{k_0}_{2m}, N^{k_0}_{2(m+3^{k_0-1})}, N^{k_0}_{2(m-3^{k_0-1})}$ have value sets

$$V^{k_0}_{2m} \subseteq \{m - \frac{3^{k_0-1}+1}{2}, m - \frac{3^{k_0-1}+1}{2} + 1, \cdots, m + \frac{3^{k_0-1}-1}{2}\}, \tag{21}$$

$$V^{k_0}_{2(m+3^{k_0-1})} \subseteq \{m + \frac{3^{k_0-1}-1}{2}, m + \frac{3^{k_0-1}-1}{2} + 1, \cdots, m + \frac{3^{k_0-1}-1}{2}\}, \tag{22}$$

$$V^{k_0}_{2(m-3^{k_0-1})} \subseteq \{m - \frac{3^{k_0-1}+1}{2}, m - \frac{3^{k_0-1}+1}{2} + 1, \cdots, m - \frac{3^{k_0-1}+1}{2}\}. \tag{23}$$

Again by Rule 3, we know that the node $N^{k_0+1}_{2m} = N^{k_0}_{2(m+3^{k-2})} \star N^{k_0}_{2(m-3^{k-2})} \star N^{k_0}_{2m}$ have value set $V^{k_0+1}_{2m} \subseteq \{m - \frac{3^{k_0}+1}{2}, m - \frac{3^{k_0}+1}{2} + 1, \cdots, m + \frac{3^{k_0}-1}{2}\}$, and therefore, $C^{k_0+1}_{2m} \leqslant 3^{k_0} + 1$. This completes the proof.

## B Proof of Theorem 6.2

We shall also use the mathematical induction to prove this theorem.

We first assert that for $1 \leqslant l \leqslant 1 + \log_2 s$ and for $j$ satisfying $1 \leqslant \sigma(\lfloor \frac{j+1}{2} \rfloor) \leqslant s - 2^{l-1} + 1$ there exists a node $N^l_k$ such that

$$\bigcup_{0\leqslant\alpha\leqslant 2^{l-1}-1} \{x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+\alpha)-1}, x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+\alpha)}\} \subseteq V^l_k. \tag{24}$$

For $l = 1$, the relation (24) can be verified easily since after position matching the nodes in $\mathcal{N}^1$ are of the form $N^1_{2m} = \{\{x_{2m-1}, x_{2m}\}, \{2m\}, \{2m-1, 2m\}\}$. And note that $x_j \in \{x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor))-1}, x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor))}\} \subseteq V^1_{2\lfloor \frac{j+1}{2} \rfloor}$.

Now we assume that (24) holds for $l = l_0$, then we prove it still holds for $l = l_0+1$. Given $x_j \in (x_i)$ with $1 \leqslant \sigma(\lfloor \frac{j+1}{2} \rfloor) \leqslant s - 2^{l_0} + 1$, it is clear that $1 \leqslant \sigma(\lfloor \frac{j+1}{2} \rfloor) \leqslant s - 2^{l_0-1} + 1$. Then by inductive hypotheses we know that there exists a node $N_{k_1}^{l_0}$ such that

$$\bigcup_{0 \leqslant \alpha \leqslant 2^{l_0-1}-1} \{x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+\alpha)-1}, x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+\alpha)}\} \subseteq V_{k_1}^{l_0}. \tag{25}$$

On the other hand, since $1 \leqslant \sigma(\lfloor \frac{j+1}{2} \rfloor) \leqslant s-2^{l_0}+1$ we have $1 \leqslant \sigma(\lfloor \frac{j+1}{2} \rfloor)+2^{l_0-1} \leqslant s-2^{l_0-1}+1$, and again by inductive hypotheses there exists a node $N_{k_2}^{l_0}$ such that

$$\bigcup_{0 \leqslant \alpha \leqslant 2^{l_0-1}-1} \{x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+2^{l_0-1}+\alpha)-1}, x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+2^{l_0-1}+\alpha)}\} \subseteq V_{k_2}^{l_0}. \tag{26}$$

Note that by the definition of reasoning sequence we have

$$x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+2^{l_0-1})-1} = x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2}+\rfloor)+2^{l_0-1}-1)}.$$

By Rule 3, there exists a node $N_{\max\{k_1,k_2\}}^{l_0+1}$ such that

$$\bigcup_{0 \leqslant \alpha \leqslant 2^{l_0}-1} \{x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+\alpha)-1}, x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+\alpha)}\} \subseteq V_{k_1}^{l_0} \cup V_{k_2}^{l_0} \subseteq V_{\max\{k_1,k_2\}}^{l_0+1}.$$

This completes the proof of our assertion.

Set the reasoning start as $x_{2s+1} = x_j$, we then assert that

$$\bigcup_{0 \leqslant \alpha \leqslant 2^{l-1}-2} \{x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+\alpha)-1}, x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+\alpha)}\} \subseteq V_{2s+1}^{l}. \tag{27}$$

When $l = 2$ the assertion (27) can be easily verified. We assume (27) holds for $l = l_0$ and consider the case $l = l_0 + 1$. By assertion (24), we know there exists $N_k^{l_0}$ such that

$$\bigcup_{0 \leqslant \alpha \leqslant 2^{l_0-1}-1} \{x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+2^{l_0-1}-1+\alpha)-1}, x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+2^{l_0-1}-1+\alpha)}\} \subseteq V_{k}^{l_0}. \tag{28}$$

Again, by definition of reasoning sequence we have

$$x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+2^{l_0-1}-1)-1} = x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2}+\rfloor)+2^{l_0-1}-2)}.$$

And by Rule 3 we know that

$$\bigcup_{0 \leqslant \alpha \leqslant 2^{l_0}-2} \{x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+\alpha)-1}, x_{2\sigma^{-1}(\sigma(\lfloor \frac{j+1}{2} \rfloor)+\alpha)}\} \subseteq V_{k}^{l_0} \cup V_{2s+1}^{l_0} \subseteq V_{2s+1}^{l_0+1}.$$

This completes the proof of assertion (27), which implies that for $l \geqslant 2$, $C_{2s+1}^{l} \geqslant 2^{l-1}$.

For $l = 1$, it is clear that $C_{2s+1}^{2} = 1$ since only residual connection happens. And we complete the proof for the lower bound.

The upper bound is proved similarly as in the proof of Theorem 6.1, except we consider mainly the reasoning start position, which results in one fewer element.

## C EXAMPLES WHERE THE THEORETICAL BOUNDS ARE ACHIEVED

In this section we give some examples related to Theorem 6.2. In fact, both the lower bound and upper bound can be attained.

### LOWER BOUND

We construct a reasoning sequence $(1, 2, 2, 3, 3, 4, 4, 5 \cdots, 2s-1, 2s, 1)$. Assume that $1 \leqslant l \leqslant 1 + \log_2 s$, it is clear that $C_{2s+1}^{l} \geqslant 2^{l-1}$

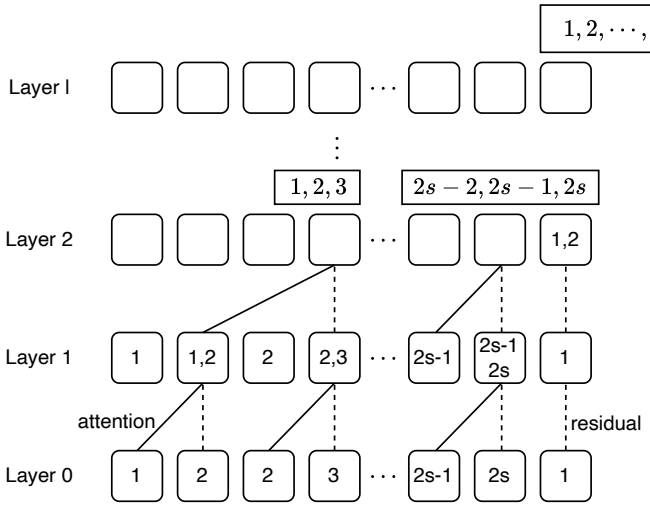



Figure 7: Lower bound example

UPPER BOUND

In this section, we shall use the concept of truncation of a reasoning sequence. This truncated reasoning sequence contains a finite step reasoning chain and some irrelevant reasoning pairs which serves as some redundant information as in practice. In fact, we truncate a reasoning sequence $(x_i)$ as follows:

1. Firstly, we truncate the constructing reasoning sequence $(\boldsymbol{a}_m)$ to be an $s$ step reasoning chain $(\tilde{\boldsymbol{a}}_k) = (\boldsymbol{a}_i, \boldsymbol{a}_{i+1}, \cdots, \boldsymbol{a}_{i+s-1})$.
2. Secondly, we use the relation (5) to find those elements constructed from this $s$ step reasoning chain. That is
$$E = \{x_{2(\sigma^{-1}(i)-1)}, x_{2(\sigma^{-1}(i)-1)+1}, \cdots, x_{2(\sigma^{-1}(i+s-1)-1)}, x_{2(\sigma^{-1}(i+s-1)-1)+1}\}. \quad (29)$$
3. Thirdly, we set
$$\begin{aligned}\mathcal{I}_E = \{&2(\sigma^{-1}(i)-1), 2(\sigma^{-1}(i)-1)+1, \cdots, \\ &2(\sigma^{-1}(i+s-1)-1), 2(\sigma^{-1}(i+s-1)-1)+1\},\end{aligned} \quad (30)$$
which is the set of all subscripts of elements in $E$. Moreover, we take the subsequence $(\tilde{x}_i) = (x_{\min\{\mathcal{I}_E\}+i-1})$ as the truncated reasoning sequence, where $1 \leqslant i \leqslant \max\{\mathcal{I}_E\} - \min\{\mathcal{I}_E\} + 1$.

We call this sequence $(\tilde{x}_i)$ the truncated reasoning sequence containing $(\tilde{\boldsymbol{a}}_k)$.

With this definition of truncation, we provide a method to construct finite step reasoning sequences which allows the upper bound in Theorem 6.2 to be attained. This is achieved in a recursive way.

Consider a reasoning sequence $(\boldsymbol{a}_i)_{i\in\mathbb{Z}}$. For a given integer $i$ we define
$$\boldsymbol{s}_1(i) = (\boldsymbol{a}_i), \quad \boldsymbol{s}_2(i) = (\boldsymbol{a}_i, \boldsymbol{a}_{i+2}, \boldsymbol{a}_{i+1}),$$
and
$$\boldsymbol{s}_3(i) = (\boldsymbol{a}_i, \boldsymbol{a}_{i+2}, \boldsymbol{a}_{i+1}, \boldsymbol{a}_{i+6}, \boldsymbol{a}_{i+8}, \boldsymbol{a}_{i+7}, \boldsymbol{a}_{i+3}, \boldsymbol{a}_{i+5}, \boldsymbol{a}_{i+4}).$$
For simplicity, we use the notation
$$\begin{aligned}\boldsymbol{s}_3(i) &= (\boldsymbol{s}_2(i), \boldsymbol{s}_2(i+6), \boldsymbol{s}_2(i+3)) \\ &:= (\boldsymbol{a}_i, \boldsymbol{a}_{i+2}, \boldsymbol{a}_{i+1}, \boldsymbol{a}_{i+6}, \boldsymbol{a}_{i+8}, \boldsymbol{a}_{i+7}, \boldsymbol{a}_{i+3}, \boldsymbol{a}_{i+5}, \boldsymbol{a}_{i+4}),\end{aligned}$$

and with this notation $\boldsymbol{s}_3(i)$ is still a sequence of reasoning pairs.

For $k \geqslant 3$ we define

$$\boldsymbol{s}_k(i) = (\boldsymbol{s}_{k-1}(i), \boldsymbol{s}_{k-1}(i + 2 \times 3^{k-2}), \boldsymbol{s}_{k-1}(i + 3^{k-2})).$$

Same notation as in $\boldsymbol{s}_3(i)$ and all $\boldsymbol{s}_k(i)$ are sequences of reasoning pairs.

We now construct our reasoning sequence.

Set $r_1 = (\boldsymbol{s}_{l-1}(i - \frac{3^{l-1}-1}{2}), \cdots, \boldsymbol{s}_3(i-13), \boldsymbol{s}_2(i-4), \boldsymbol{s}_1(i-1), \boldsymbol{s}_1(i), \boldsymbol{s}_2(i+1), \boldsymbol{s}_3(i+4), \cdots, \boldsymbol{s}_{l-1}(i + \frac{3^{l-2}-1}{2}))$, it is clear that there exist $\sigma \in \mathrm{Sym}(\mathbb{Z})$ such that $(\boldsymbol{a}_{\sigma(1-\frac{3^{l-1}}{2})}, \cdots, \boldsymbol{a}_{\sigma(1)}, \boldsymbol{a}_{\sigma(2)}, \cdots, \boldsymbol{a}_{\sigma(\frac{3^{l-1}-1}{2})}) = r_1$. By the construction of $r_1$ we know all the reasoning pairs in $r_1$ forms a finite step reasoning chain $(\boldsymbol{a}_{i-\frac{3^{l-1}-1}{2}}, \cdots \boldsymbol{a}_i, \boldsymbol{a}_{i+1}, \cdots, \boldsymbol{a}_{i+\frac{3^{l-1}-3}{2}})$, we extend this finite step reasoning chain to an infinite reasoning chain $\tilde{a}$ by adding reasoning pairs to both sides. And we take a permutation $\tau \in \mathrm{Sym}(\mathbb{Z})$ satisfying $\boldsymbol{a}_{\tau(i_m)} = \boldsymbol{a}_{\sigma(m)}$ where $i_m \in I$ with $I = \{i_{1-\frac{3^{l-1}}{2}}, \cdots, i_1, i_2, \cdots, i_{\frac{3^{l-1}-3}{2}}\}$ and $i_{1-\frac{3^{l-1}}{2}} < \cdots < i_1 < i_2 < \cdots < i_{\frac{3^{l-1}-3}{2}}$.

We truncate the reasoning sequence $((x_j), \tilde{\boldsymbol{a}}, \tau)$ to contain the finite reasoning chain $(\boldsymbol{a}_{i-\frac{3^{l-1}-1}{2}}, \cdots, \boldsymbol{a}_{i+\frac{3^{l-1}-3}{2}})$ to get a sequence $(\tilde{x}_j) = (x_{\min\{\mathcal{I}_E\}+j-1})_{1 \leqslant j \leqslant \max\{\mathcal{I}_E\}-\min\{\mathcal{I}_E\}+1}$. Here $E$ and $\mathcal{I}_E$ are defined as in (29) and (30). This $(\tilde{x}_j)$ with reasoning start $x_{\max \mathcal{I}_E+1} = \boldsymbol{a}_i^1$ is the sequence we want.

**Remark C.1.** *Under the rules of information propagation, the elements in $(\tilde{x}_j)$ contributing to the node $N^l_{\max \mathcal{I}_E+1}$ are those come from the reasoning chain $(\boldsymbol{a}_{i-\frac{3^{l-1}-1}{2}}, \cdots, \boldsymbol{a}_{i+\frac{3^{l-1}-3}{2}})$. Hence, the information in $N^l_{\max \mathcal{I}_E+1}$ will be the same if we consider the reasoning sequence constructed from $(\boldsymbol{a}_{i-\frac{3^{l-1}-1}{2}}, \cdots, \boldsymbol{a}_{i+\frac{3^{l-1}-3}{2}})$ and $\sigma$. However, the above complicated way we construct truncated sequence is still necessary which shows that there is a large class of reasoning sequence allows the upper bound to be attained.*

**Proposition C.2.** *We consider the reasoning sequence $(x_i)$ with constructing reasoning chain $(\boldsymbol{a}_m) = ((m, m+1))_{1-\frac{3^{\tilde{l}-1}-1}{2} \leqslant m \leqslant \frac{3^{\tilde{l}-1}-1}{2}}$ and constructing permutation $\sigma$ satisfying that*

$$(\boldsymbol{a}_{\sigma(1)}, \boldsymbol{a}_{\sigma(2)}, \cdots, \boldsymbol{a}_{\sigma(\frac{3^{\tilde{l}-1}-1}{2})}) = (\boldsymbol{s}_1(1), \boldsymbol{s}_2(2), \boldsymbol{s}_3(5), \cdots, \boldsymbol{s}_l(\frac{3^{l-1}+1}{2}), \cdots \boldsymbol{s}_{\tilde{l}-1}(\frac{3^{\tilde{l}-2}+1}{2})), \tag{31}$$

$$(\boldsymbol{a}_{\sigma(1-\frac{3^{\tilde{l}-1}}{2})}, \boldsymbol{a}_{\sigma(2-\frac{3^{\tilde{l}-1}}{2})}, \cdots, \boldsymbol{a}_{\sigma(0)}) = (\boldsymbol{s}_{\tilde{l}-1}(\frac{3-3^{\tilde{l}-1}}{2}), \cdots, \boldsymbol{s}_l(\frac{3-3^l}{2}), \cdots \boldsymbol{s}_3(-12), \boldsymbol{s}_2(-3), \boldsymbol{s}_1(0), ). \tag{32}$$

*If the reasoning start is set be to $x_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1} = \boldsymbol{a}_1^1$, then for $1 \leqslant l \leqslant \tilde{l}$, we have $C^l_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1} = 3^{l-1}$.*

To prove this proposition we need the following two lemmas.

**Lemma C.3.** *For $m \geqslant 1$, $\forall k \in \{\frac{3^{j-2}+1}{2}\}_{2 \leqslant j \leqslant \tilde{l}}$, there exists $i_{(m,k)} \in [1, 3^{\tilde{l}-1}-1] \cap \mathbb{Z}$ depending on $m$ and $k$ such that the node $N^m_{i_{(m,k)}}$ have value set $V^m_{i_{(m,k)}}$ satisfying the following property.*

$$\{k, k+1, \cdots, k+3^{m-1}\} \subseteq V^m_{i_{(m,k)}}. \tag{33}$$

**Lemma C.4.** *For $m \geqslant 1$, $\forall k \in \{\frac{3-3^{j-1}}{2}\}_{2 \leqslant j \leqslant \tilde{l}}$, there exist $i_{(m,k)} \in [1-3^{\tilde{l}-1}, 0] \cap \mathbb{Z}$ such that the node $N^m_{i_{(m,k)}}$ have value set $V^m_{i_{(m,k)}}$ satisfying the following property.*

$$\{k, k+1, \cdots, k+3^{m-1}\} \subseteq V^m_{i_{(m,k)}}. \tag{34}$$

***Proof of Proposition C.2*** . The case $l = 1$ is trivial and omitted. We use mathematical induction to prove this proposition. We assert that for $l \geqslant 2$ the reasoning start node $N^l_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1}$ have value set $V^l_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1}$ satisfying

$$\{\frac{3-3^{l-1}}{2}, \frac{3-3^{l-1}}{2}+1, \cdots, 0, 1, 2, \cdots, \frac{3^{l-1}+1}{2}\} \subseteq V^l_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1}. \tag{35}$$

The case $l = 2$ can be verified easily. We assume that for $l = l_0 \in [2, \tilde{l}-1] \cap \mathbb{Z}$ the node $N^{l_0}_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1}$ have value set $V^{l_0}_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1}$ satisfying

$$\{\frac{3-3^{l_0-1}}{2}, \frac{3-3^{l_0-1}}{2}+1, \cdots, 0, 1, 2, \cdots, \frac{3^{l_0-1}+1}{2}\} \subseteq V^l_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1}. \tag{36}$$

Now for $l = l_0 + 1$, by Lemma C.3, we know there exists a node $N^{l_0}_{i(l_0, \frac{3^{l_0-1}+1}{2})}$ such that

$$\{\frac{3^{l_0-1}+1}{2}, \cdots, \frac{3^{l_0-1}+1}{2}+3^{l_0-1}\} \subseteq V^{l_0}_{i(l_0, \frac{3^{l_0-1}+1}{2})}.$$

And by Lemma C.4 there exist a node $N^{l_0}_{i(l_0, \frac{3-3^{l_0}}{2})}$ such that

$$\{\frac{3-3^{l_0}}{2}, \cdots, \frac{3-3^{l_0}}{2}+3^{l_0-1}\} \subseteq V^{l_0}_{i(l_0, \frac{3-3^{l_0}}{2})}.$$

We then have

$$\begin{aligned} V^{l_0}_{i(l_0, \frac{3^{l_0-1}+1}{2})} \cap V^{l_0}_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1} &= \{\frac{3^{l_0-1}+1}{2}\}, \\ V^{l_0}_{i(l_0, \frac{3-3^{l_0}}{2})} \cap V^{l_0}_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1} &= \{\frac{3-3^{l_0-1}}{2}\}, \end{aligned} \tag{37}$$

and by Rule 3, we know that the node $N^{l_0+1}_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1}$ have value set $V^{l_0+1}_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1}$ satisfying

$$\begin{aligned} \{\frac{3-3^{l_0}}{2} \cdots, 0, 1, 2, \cdots, \frac{3^{l_0}+1}{2}\} &= V^{l_0}_{i(l_0, \frac{3^{l_0-2}+1}{2})} \cup V^{l_0}_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1} \cup V^{l_0}_{i(l_0, \frac{3-3^{l_0-1}}{2})} \\ &\subseteq V^{l_0+1}_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1}. \end{aligned} \tag{38}$$

This completes the proof of our assertion. From above assertion we know that $C^l_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1} \geqslant 3^{l-1}$. Combining the proof of Theorem 6.2, we know that $C^l_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1} \leqslant 3^{l-1}$. And therefore $C^l_{2\sigma(\frac{3^{\tilde{l}-1}-1}{2})+1} = 3^{l-1}$. □

***Proof of Lemma C.3*** . We use mathematical induction to prove this lemma. The case $m = 2$ can be verified easily through Rule 2. We assume that for $m = m_0$, $\forall k \in \{\frac{3^{l-2}+1}{2}\}_{2 \leqslant l \leqslant \tilde{l}}$,

$$\{k, k+1, \cdots, k+3^{m_0-1}\} \subseteq V^{m_0}_{i_{(m_0,k)}}. \tag{39}$$

For $m = m_0 + 1$, by assumption, there exists $i(m_0, k)$, $i(m_0, k+3^{m_0-1})$, $i(m_0, k+2\times 3^{m_0-1})$ such that

$$\begin{aligned} &\{k, \cdots, k+3^{m_0-1}\} \subseteq V^{m_0}_{i(m_0,k)}, \\ &\{k+3^{m_0-1}, \cdots, k+2\times 3^{m_0-1}\} \subseteq V^{m_0}_{i(m_0,k+3^{m_0-1})}, \\ &\{k+2\times 3^{m_0-1}, \cdots, k+3^{m_0}\} \subseteq V^{m_0}_{i(m_0,k+2\times 3^{m_0-1})}. \end{aligned} \tag{40}$$

Set $i(m_0+1,k) = \max\{i(m_0,k), i(m_0,k+3^{m_0-1}), i(m_0,k+2\times 3^{m_0-1})\}$, by Rule 3 and (40) we know that

$$\{k,\cdots,k+3^{m_0}\} \subseteq V^{m_0}_{i(m_0,k)} \cup V^{m_0}_{i(m_0,k+3^{m_0-1})} \cup V^{m_0}_{i(m_0,k+2\times 3^{m_0-1})} \subseteq V^{m_0+1}_{i(m_0+1,k)}, \tag{41}$$

which completes the proof. □

The proof of Lemma C.4 is analogous to that of Lemma C.3 and is omitted.

# D Construction of transformer

## D.1 Embedding

We assume that $d_m$ is large enough. We choose a suitable embedding (which can be done by choosing a suitable basis of $\mathbb{R}^{d_m}$) such that the non-zero elements of all value vector $\boldsymbol{X}^{pos}$ are located in the first $n$ coordinates and all the elements in $\boldsymbol{X}^{tgt}$ located at the first $n$ coordinates are zero.

We denote $\boldsymbol{X}^{tgt} = (\boldsymbol{X}_1^{tgt,\mathsf{T}}, \boldsymbol{X}_2^{tgt,\mathsf{T}}, \cdots, \boldsymbol{X}_n^{tgt,\mathsf{T}})^{\mathsf{T}}$ with each $\boldsymbol{X}_i^{tgt} = (\boldsymbol{X}_{i,1}^{tgt}, \boldsymbol{X}_{i,2}^{tgt}, \cdots, \boldsymbol{X}_{i,d_m}^{tgt}) \in \mathbb{R}^{d_m}$. Similarly, we denote $\boldsymbol{X}^{pos} = (\boldsymbol{X}_1^{pos,\mathsf{T}}, \boldsymbol{X}_2^{pos,\mathsf{T}}, \cdots, \boldsymbol{X}_n^{pos,\mathsf{T}})^{\mathsf{T}}$ and $\boldsymbol{X}^{(l)} = (\boldsymbol{X}_1^{(l),\mathsf{T}}, \boldsymbol{X}_2^{(l),\mathsf{T}}, \cdots, \boldsymbol{X}_n^{(l),\mathsf{T}})^{\mathsf{T}}$.

We correspond each element in the set $\tilde{E}$ to an index i.e., the set $\tilde{E}$ is of the form $\tilde{E} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_d\}$. Moreover, each $\boldsymbol{v}_i$ is of the form $(\underbrace{0,0,\cdots,0}_{n \text{ zeros}},0,\cdots,1,\cdots,0)$, and we denote $k_i$ the position of 1. Let $n \leqslant k_1 < k_2 < \cdots < k_d$, we require the embedding is chosen such that $k_i$ satisfies that

$$\begin{cases} k_1 - n \geqslant 2(n+1)(3^L+1), \\ k_i - k_{i-1} \geqslant 2(n+1)(3^L+1), \quad \text{for } 2 \leqslant i \leqslant d, \\ d_m - k_d \geqslant 2(n+1)(3^L+1). \end{cases} \tag{42}$$

## D.2 Construction of parameters

We choose the positional encodings to be the first $n$ standard-basis vectors, $\boldsymbol{p}_i = e_i$. The spacing condition above applies to the entire dictionary, so position blocks remain disjoint for every token choice; this will also allow a shared FNN in Proposition D.7. We set the weight matrices as follows:

$$\boldsymbol{W}^{q(l)} = \boldsymbol{I}(0 \leqslant l \leqslant L), \tag{43}$$

$$\boldsymbol{W}^{vo(l)} = \boldsymbol{I}(1 \leqslant l \leqslant L),\ \boldsymbol{W}^{vo(0)} = \boldsymbol{R}, \tag{44}$$

$$\boldsymbol{W}^{k(l),\mathsf{T}} = \sum_{i=-(n+1)3^L}^{-1} \boldsymbol{R}^i (l \geqslant 1),\ \ \boldsymbol{W}^{k(0)} = \sum_{i=1}^{\frac{n-1}{2}} \boldsymbol{p}_{2i-1}\boldsymbol{p}_{2i}^T. \tag{45}$$

Here $\boldsymbol{p}_j$ is the positional encoding and arbitrary two positional encodings are orthogonal to each other. Moreover, $\boldsymbol{R} \in \mathbb{R}^{d_m \times d_m}$ is defined as $\boldsymbol{R} = [\boldsymbol{R}_{ij}]$, with $\boldsymbol{R}_{i+1,i} = 1 = \boldsymbol{R}_{1d_m}$ for $1 \leqslant i \leqslant d_m - 1$, and all the other elements of $\boldsymbol{R}$ are set to be 0. In fact,

$$\boldsymbol{R} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

This matrix $\boldsymbol{R}$ is called the left shift matrix. When we apply $\boldsymbol{R}$ to a vector $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_{d_m})$ then result is $\boldsymbol{vR} = (\boldsymbol{v}_2, \cdots, \boldsymbol{v}_{d_m-1}, \boldsymbol{v}_{d_m}, \boldsymbol{v}_1)$. The notation $\boldsymbol{R}^i$ means that the matrix $\boldsymbol{R}$ act $i$ times, i.e., $\boldsymbol{vR}^i = \boldsymbol{vRR}\cdots\boldsymbol{R}$. Also, the inverse of $mR$ is called the right shift matrix, and $\boldsymbol{R}^{-1} = \boldsymbol{R}^{\mathsf{T}}$.

For any given matrix $\boldsymbol{M} = [m_{ij}]$, we define the mask operation as

$$\text{mask}(\boldsymbol{M}) = \begin{bmatrix} m_{1,1}^{(l)} & -\infty & -\infty & \cdots & -\infty \\ m_{2,1}^{(l)} & m_{2,2}^{(l)} & -\infty & \cdots & -\infty \\ m_{3,1}^{(l)} & m_{3,2}^{(l)} & m_{3,3}^{(l)} & \cdots & -\infty \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n,1}^{(l)} & m_{n,2}^{(l)} & \cdots & m_{n,n-1}^{(l)} & m_{n,n}^{(l)} \end{bmatrix}. \tag{46}$$

Furthermore, we denote

$$\boldsymbol{A}^{(l)} = \text{mask}(\boldsymbol{X}^{(l)}\boldsymbol{W}^{q(l)}\boldsymbol{W}^{k(l),\mathsf{T}}\boldsymbol{X}^{(l),\mathsf{T}}) = \begin{bmatrix} \boldsymbol{A}_{1,1}^{(l)} & -\infty & -\infty & \cdots & -\infty \\ \boldsymbol{A}_{2,1}^{(l)} & \boldsymbol{A}_{2,2}^{(l)} & -\infty & \cdots & -\infty \\ \boldsymbol{A}_{3,1}^{(l)} & \boldsymbol{A}_{3,2}^{(l)} & \boldsymbol{A}_{3,3}^{(l)} & \cdots & -\infty \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{n,1}^{(l)} & \boldsymbol{A}_{n,2}^{(l)} & \cdots & \boldsymbol{A}_{n,n-1}^{(l)} & \boldsymbol{A}_{n,n}^{(l)} \end{bmatrix}. \tag{47}$$

We shall still use the notation $N_i^l$ to denote the $i$th node in $l$th layer with slightly modification: the value set $\boldsymbol{V}_i^l$ is now a set $\boldsymbol{V}_i^l$ which contains the embedded value as element.

Given any reasoning chain $(\boldsymbol{a}_u)_{1\leqslant i\leqslant n}$, after embedding it is of the form

$$(\tilde{\boldsymbol{a}}_u) = ((\tilde{\boldsymbol{a}}_u^1, \tilde{\boldsymbol{a}}_u^2))_{1\leqslant i\leqslant n} = ((\boldsymbol{a}_u^1\boldsymbol{W}^{\text{emb}}, \boldsymbol{a}_u^2\boldsymbol{W}^{\text{emb}}))_{1\leqslant i\leqslant n}.$$

We define a sequence $(\tilde{\boldsymbol{b}}_v)_{1\leqslant v\leqslant n+1}$ as follows:

$$\begin{aligned} &\tilde{\boldsymbol{b}}_v = \tilde{\boldsymbol{a}}_v^1, \ \text{ for } 1 \leqslant v \leqslant n, \\ &\tilde{\boldsymbol{b}}_v = \tilde{\boldsymbol{a}}_n^2 \ \text{ for } v = n+1. \end{aligned} \tag{48}$$

Assume the node $N_i^l$ contains the information from $(\tilde{a}_i)$, then its value set $\boldsymbol{V}_i^l$ must contain the elements from a subsequence of $\tilde{\boldsymbol{b}}_{v\in[v_1,v_2]\cap\mathbb{Z}}$ where $v_1, v_2$ depends on $i$ and $l$. For simplicity, we set $\boldsymbol{b}_{i,v}^l = \tilde{\boldsymbol{b}}_{v_1+v-1}$ for $v \in [1, v_2 - v_1 + 1] \cap \mathbb{Z}$. In fact, $v_2 - v_1 + 1 = C_i^l$.

With these notations, we define the FNN as follows

$$\begin{cases} R_N^l(f_i^l(\boldsymbol{X}_i^{ao(l)}) + (\boldsymbol{X}_i^{ao(l)})) = \displaystyle\sum_{1\leqslant k\leqslant C_i^l} \boldsymbol{b}_{i;k}\boldsymbol{R}^{i3^L-j+k} \text{ for } i \geqslant 2, \\ R_N^l(f_i^l(\boldsymbol{X}_i^{ao(l)}) + (\boldsymbol{X}_i^{ao(l)})) = \boldsymbol{X}_1^{tgt}\boldsymbol{R}^{(n+1)3^L} \text{ for } \ i = 1. \end{cases} \tag{49}$$

Here $R_N$ denotes RMSNorm with a positive stability constant $\epsilon$ and zero bias; we choose a scalar gain $\alpha > 1$. Its definition and injectivity are given in Lemma D.9. The index $j \in [1, n] \cap \mathbb{Z}$ satisfies $\boldsymbol{b}_{i,j}^l = X_i^{tgt}$. The subscript $i$ on the local FNN rules specifies their input domains. Proposition D.7 shows that these rules can be realized, to arbitrary accuracy, by one FNN shared across all positions in each layer.

**Remark D.1.** *Note that this is not an accurate expression, since we can not simply define the output of a FNN. However, we can show that there exist $f_i^{(l)}$ for each layer such that the output of $R_N(f_i^l(\boldsymbol{X}_i^{ao(l)}) + (\boldsymbol{X}_i^{ao(l)}))$ differs very little from the right-hand side of* (49)*. And also the information propagation won't be affected by this error. This will be explained in detail in later sections.*

By this construction the reasoning information contained in the node $N_i^l$ is encoded by $R_N(f_i^l(\cdot) + (\cdot))$.

In order to extract the result after m-step reasoning, we set

$$\boldsymbol{W}^p = \boldsymbol{R}^{-n3^L-m+1}\boldsymbol{Q}\boldsymbol{W}^{emb,\mathsf{T}}, \tag{50}$$

where $\boldsymbol{Q}$ satisfies $\boldsymbol{Q}\boldsymbol{v}_1 = \boldsymbol{v}_1$, ..., $\boldsymbol{Q}\boldsymbol{v}_d = \boldsymbol{v}_d$ and maps other basis to 0. Note that last token on the $s$-th layer is $\boldsymbol{X}_n^{(L)} = \boldsymbol{b}_{n;1}\boldsymbol{R}^{n3^s-j+1} + ..... + \boldsymbol{b}_{n;t}\boldsymbol{R}^{n3^s+t-j}$, and we have

$$\boldsymbol{X}_n^{(L)}\boldsymbol{W}^p = \boldsymbol{b}_{n;1}\boldsymbol{R}^{-j-m+2}\boldsymbol{Q}\boldsymbol{W}^{emb,\mathsf{T}} + ..... + \boldsymbol{b}_{n;t}\boldsymbol{R}^{t-j-m+1}\boldsymbol{Q}\boldsymbol{W}^{emb,\mathsf{T}}, \tag{51}$$

and the output

$$\boldsymbol{Y} = argmax(\tilde{\sigma}(\boldsymbol{X}_n^{(L)}\boldsymbol{W}^p)). \tag{52}$$

When $m + j - 1 \leqslant t$, $\boldsymbol{X}_n^{(L)}\boldsymbol{W}^p = \boldsymbol{b}_{n;m+j-1}\boldsymbol{W}^{emb,\mathsf{T}}$ and the above setting yields the desired reasoning result. However, when $m + j - 1 > t$, $\boldsymbol{X}_n^{(L)}\boldsymbol{W}^p = 0$ and therefore we can not get a right reasoning result.

In fact, the sufficiency of transformer depth $s$ relative to the required reasoning steps $m$ is a key factor in ensuring accurate reasoning result.

We can roughly categorize this relationship into three cases

- **Case 1** $m \leqslant 2^{L-1} - 1$;
- **Case 2** $2^{s-1} \leqslant m \leqslant \frac{3^{L-1}+1}{2}$;
- **Case 3** $m > \frac{3^{L-1}-1}{2}$.

For the first case, note that $t - j + 1 \geq 2^{L-1} - 1$ and therefore $m + j - 1 \leq t$ which means we can get the result. For the third case, since $t - j + 1 \leq \frac{3^{L-1}+1}{2}$ and consequently $m + j - 1 > t$, the model can not derive the result. The second case is more complicated, since we can not derive the relationship of $t - j + 1$ and $t$ from the relationship of $m$ and $s$. Whether the model can derive the reasoning result now depends on $t - j + 1$ and $t$.

### D.3 Explanation

First, we explain how the same token matching rule works in this construction. More specifically, the attention matrices defined above satisfy the following property.

**Lemma D.2.** *For $l \geqslant 1$, we have*

$$\begin{aligned}
&\boldsymbol{A}_{i,j}^{(l)} = 0, \ \textit{if} \ \ j = 1, \\
&\boldsymbol{A}_{i,j}^{(l)} = 0, \ \textit{if} \ \ i \geqslant j \geqslant 2, \ \boldsymbol{V}_i^l \cap \boldsymbol{V}_j^l = \emptyset, \\
&\boldsymbol{A}_{i,j}^{(l)} \geqslant 1, \ \textit{if} \ \ i \geqslant j \geqslant 2, \ \boldsymbol{V}_i^l \cap \boldsymbol{V}_j^l \neq \emptyset.
\end{aligned} \tag{53}$$

*Proof.* Start with given any $\boldsymbol{X}^{(0)} = (\boldsymbol{X}_1^{(0)}, \boldsymbol{X}_2^{(0)}, \cdots, \boldsymbol{X}_n^{(0)})^\mathsf{T}$, we denote $\boldsymbol{X}^{(l)} = (\boldsymbol{X}_1^{(l)}, \boldsymbol{X}_2^{(l)}, \cdots, \boldsymbol{X}_n^{(l)})$ with $\boldsymbol{X}_i^{(l)} = \sum_{u=1}^{C_i^l} \boldsymbol{b}_{i;u}^l \boldsymbol{R}^{i3^L+u-d_i^{(l)}}$, where $C_i^l = |\boldsymbol{V}_i^l|$ and $\boldsymbol{b}_{i;u}^l \in \tilde{E}$.

$$\boldsymbol{A}_{i,j}^{(l-1)} = \boldsymbol{X}_i^{(l-1)}\boldsymbol{R}^{q(l-1)}\boldsymbol{R}^{k(l-1),\mathsf{T}}\boldsymbol{X}_j^{(l-1),\mathsf{T}} \tag{54}$$

$$= (\sum_{u=1}^{C_i^{l-1}} \boldsymbol{b}_{i;u}^{l-1}\boldsymbol{R}^{i3^L+u-d_i^{(l-1)}})(\sum_{m=-(n+1)3^L}^{-1} \boldsymbol{R}^m)(\sum_{v=1}^{C_j^{l-1}} \boldsymbol{b}_{j;v}^{l-1}\boldsymbol{R}^{j3^L+v-d_j^{(l-1)}})^\mathsf{T} \tag{55}$$

$$= \sum_{u=1}^{C_i^{l-1}} \sum_{m=-(n+1)3^L}^{-1} \sum_{v=1}^{C_j^{l-1}} \boldsymbol{b}_{i;u}^{l-1}\boldsymbol{R}^{i3^L+u-d_i^{(l-1)}-j3^L-v+d_j^{(l-1)}+m}\boldsymbol{b}_{j;v}^{l-1,\mathsf{T}} \tag{56}$$

For any $i \geqslant j \geqslant 2$ and $\boldsymbol{V}_i^{l-1} \cap \boldsymbol{V}_j^{l-1} = \emptyset$.

Since $-(n+1)3^L \leqslant i3^L + u - d_i^{(l-1)} - j3^L - v + d_j^{(l-1)} + m \leqslant (n+1)3^L$ and $\boldsymbol{b}_{i;u}^l \neq \boldsymbol{b}_{j;v}^l$, we have $\boldsymbol{b}_{i;u}^{l-1}\boldsymbol{R}^{i3^L+u-d_i^{(l-1)}-j3^L-v+d_j^{(l-1)}+m}\boldsymbol{b}_{j;v}^{l-1,\mathsf{T}} = 0$. Therefore, $\boldsymbol{A}_{i,j}^{(l-1)} = 0$.

When $i \geqslant j \geqslant 2$ and $\boldsymbol{V}_i^{l-1} \cap \boldsymbol{V}_j^{l-1} \neq \emptyset$,

$$\boldsymbol{A}_{i,j}^{(l-1)} = \sum_{u=1}^{C_i^{l-1}} \sum_{m=-(n+1)3^L}^{-1} \sum_{v=1}^{C_j^{l-1}} \boldsymbol{b}_{i;u}^{l-1}\boldsymbol{R}^{i3^L+u-d_i^{(l-1)}-j3^L-v+d_j^{(l-1)}+m}\boldsymbol{b}_{j;v}^{l-1,\mathsf{T}} \tag{57}$$

Since $\boldsymbol{V}_i^{l-1} \cap \boldsymbol{V}_j^{l-1} \neq \emptyset$, there exists $j, v, i, v_0$ such that $\boldsymbol{b}_{j;v}^{l-1} = \boldsymbol{b}_{i;v_0}^{l-1}$. Moreover, since $1 \leqslant i3^L + u - d_i^{(l-1)} - j3^L - v + d_j^{(l-1)} + m \leqslant (n+1)3^L$ and there exists $j, v, i, v_0$ such that $\boldsymbol{b}_{j;v}^{l-1} = \boldsymbol{b}_{i;v_0}^{l-1}$, there exists $m_0$ s.t. $\boldsymbol{b}_{j;v}^{l-1}\boldsymbol{R}^{i3^L+u_0-d_i^{(l-1)}-j3^L-v_0+d_j^{(l-1)}+m_0}\boldsymbol{b}_{j;v_0}^{(l-1),\mathsf{T}} = 1$. This indicates that $\boldsymbol{A}_{i,j}^{(l-1)} \geq 1$

When $j = 1$,

$$\boldsymbol{A}_{i,1}^{(l-1)} = \sum_{u=1}^{C_i^{l-1}} \sum_{m=-(n+1)3^L}^{-1} \boldsymbol{b}_{i;u}^{l-1}\boldsymbol{R}^{i3^L+u-d_i^{(l-1)}-(n+1)3^L-1+m}\boldsymbol{X}_1^{tgt,\mathsf{T}} \tag{58}$$

Since $-2(n+1)3^L - 1 \leqslant i3^L + u - d_i^{(l-1)} - (n+1)3^L - 1 + m \leqslant -1$, we know that $\boldsymbol{A}_{i,1}^{(l-1)} = 0$. □

### D.4 Information propagation

Next, we explain how the above defined transformer extract reasoning result.

$\boldsymbol{X}^{ao(0)} = \boldsymbol{X}^{(0)} + \boldsymbol{X}^{qkv(0)}$

When $l = 0$, for $i \geqslant j$, using (43), (44), (45) and (47) we have

$$\begin{aligned}
\boldsymbol{A}_{i,j}^{(0)} &= \boldsymbol{X}_i^{(0)}(\sum_{t=1}^{\frac{n-1}{2}} \boldsymbol{p}_{2t}\boldsymbol{p}_{2t-1}^{\mathsf{T}})\boldsymbol{X}_j^{(0),\mathsf{T}} \\
&= \sum_{t=1}^{\frac{n-1}{2}} \boldsymbol{X}_i^{tgt}\boldsymbol{p}_{2t}\boldsymbol{p}_{2t-1}^{\mathsf{T}}\boldsymbol{X}_j^{tgt,\mathsf{T}} + \sum_{t=1}^{\frac{n-1}{2}} \boldsymbol{X}_i^{pos}\boldsymbol{p}_{2t}\boldsymbol{p}_{2t-1}^{\mathsf{T}}\boldsymbol{X}_j^{tgt,\mathsf{T}} + \sum_{t=1}^{\frac{n-1}{2}} \boldsymbol{X}_i^{pos}\boldsymbol{p}_{2t}\boldsymbol{p}_{2t-1}^{\mathsf{T}}\boldsymbol{X}_j^{pos,\mathsf{T}} \\
&+ \sum_{t=1}^{\frac{n-1}{2}} \boldsymbol{X}_i^{tgt}\boldsymbol{p}_{2t}\boldsymbol{p}_{2t-1}^{\mathsf{T}}\boldsymbol{X}_j^{pos,\mathsf{T}}
\end{aligned} \tag{59}$$

It is clear that

$$\begin{aligned}
\boldsymbol{A}_{i,j}^{(0)} &= 1, \text{ when } i = j+1 \text{ and } i \bmod 2 = 0, \\
\boldsymbol{A}_{i,j}^{(0)} &= 0, \text{ when } i \neq j+1 \text{ or } i \bmod 2 \neq 0.
\end{aligned} \tag{60}$$

Then for $m \in [1, n] \cap \mathbb{Z}$, we get $\boldsymbol{X}_m^{ao(0)} = \sum_{i=1}^m \frac{\exp(\boldsymbol{A}_{m,i}^{(0)})}{\sum_{j=1}^m \exp(\boldsymbol{A}_{m,j}^{(0)})}\boldsymbol{X}_i^{(0)}\boldsymbol{R} + \boldsymbol{X}_m^{(0)}$, when $m \bmod 2 = 0$, we can get reasoning chain $(\boldsymbol{b}_{m;1}^{(1)}, \boldsymbol{b}_{m;2}^{(1)})$.

In fact, we set all the coefficient $\frac{\exp(0)}{\sum_{j=1}^m \exp(\boldsymbol{A}_{m,j}^{(0)})}$ to be 0, then $\boldsymbol{X}_m^{ao(1)}$ is of the form $\frac{\exp(1)}{\sum_{t=1}^m \exp(\boldsymbol{A}_{m,t}^{(0)})}\boldsymbol{b}_{m;1}^{(1)}\boldsymbol{R} + \boldsymbol{b}_{m;2}^{(1)}$.

By (49) we have $R_N^0(f_m^{(0)}(\boldsymbol{X}_m^{ao(0)}) + \boldsymbol{X}_m^{ao(0)}) = b_{m;1}^{(1)}\boldsymbol{R}^{m3^L-1} + b_{m;2}^{(1)}\boldsymbol{R}^{m3^L}$

When $m \bmod 2 \neq 0$ and $m > 1$, we can recognize $b_{m;1}^{(0)} = \boldsymbol{X}_m^{tgt}$, and $R_N^0(f_m^{(0)}(\boldsymbol{X}_m^{ao(0)}) + \boldsymbol{X}_m^{ao(0)}) = b_{m;1}^{(1)}\boldsymbol{R}^{m3^s}$

When $m = 1$, we can recognize $\boldsymbol{b}_{1;1}^{(1)} = \boldsymbol{X}_1^{tgt}$, and $R_N^0(f_1^{(0)}(\boldsymbol{X}_1^{ao(0)} + \boldsymbol{X}_1^{ao(0)})) = \boldsymbol{b}_{1;1}^{(1)}\boldsymbol{R}^{(n+1)3^L}$.

After the first layer decoder, the even positions pass information to subsequent odd positions. And we can eliminate the position vectors on the second floor.

In fact, by our construction we have

$$\boldsymbol{X}_i^{ao(l-1)} = \sum_{j=1}^{i} \frac{\exp(\boldsymbol{A}_{i,j}^{(l-1)})}{\sum_{t=1}^{i}\exp(\boldsymbol{A}_{i,t}^{(l-1)})}\boldsymbol{X}_j^{(l-1)} + \boldsymbol{X}_i^{(l-1)} \tag{61}$$

$$= \sum_{j=1}^{i} \frac{\exp(\boldsymbol{A}_{i,j}^{(l-1)})}{\sum_{t=1}^{i}\exp(\boldsymbol{A}_{i,t}^{(l-1)})} \sum_{u=1}^{C_j^{l-1}} \boldsymbol{b}_{j;u}^{l-1}\boldsymbol{R}^{j3^L+u-d_j^{(l-1)}} + \sum_{u=1}^{C_i^{l-1}} \boldsymbol{b}_{i;u}^{l-1}\boldsymbol{R}^{i3^L+u-d_i^{(l-1)}}. \tag{62}$$

For any $i,\ j$ such that $i \neq j$, note that $|u - d_j^{(l-1)}| \leqslant \frac{3^L-1}{2}$ and $|v - d_i^{(l-1)}| \leqslant \frac{3^L-1}{2}$, by theorem 6.2, we have

$$|j3^L + u - d_j^{(l-1)} - i3^L - v + d_i^{(l-1)}| = |(j-i)3^L + u - v - d_j^{(l-1)} + d_i^{(l-1)}| \tag{63}$$

$$\geqslant 3^L - |u - v - d_j^{(l-1)} + d_i^{(l-1)}| \geqslant 1, \tag{64}$$

$$|i3^L + u - d_i^{(l-1)}| \leqslant (n+1)3^L. \tag{65}$$

And therefore, the nonzero element of $\boldsymbol{b}_{j;u}^{l-1}\boldsymbol{R}^{j3^L+u-d_j^{(l-1)}}$ and $\boldsymbol{b}_{i;u}^{l-1}\boldsymbol{R}^{i3^L+u-d_i^{(l-1)}}$ $(i \neq j)$ won't locate at the same coordinate.

Denote $\boldsymbol{X}_i^{ao(l-1)} = (\boldsymbol{X}_{i,1}^{ao(l-1)}, \cdots, \boldsymbol{X}_{i,d_m}^{ao(l-1)})$. Firstly, set the number on the axis which equals to $min_{1\leqslant j\leqslant i}\{\boldsymbol{X}_{i,j}^{ao(l-1)} > 0\}$ to be 0.

Suppose the nonzero component of $X_1^{tgt}$ located at the $k_p$-th axis. Then, $X_{i,k_p-(n+1)3^L}^{ao(l-1)} = \frac{exp(0)}{\sum_{1\leqslant j\leqslant i} exp(A_{i,j}^{(l-1)})}$. Since $\min\limits_{1\leqslant j\leqslant i}\{\boldsymbol{X}_{i,j}^{ao(l-1)} > 0\} = \frac{1}{\sum_{t=1}^{i}\exp(\boldsymbol{A}_{i,t}^{(l-1)})}$, there remains the information propagated from $\boldsymbol{X}_j^{(l-1)}$ s.t. $\boldsymbol{A}_{i,j}^{(l-1)} \geqslant 1$ which indicates that $\boldsymbol{V}_i^{l-1} \cap \boldsymbol{V}_j^{l-1} \neq \emptyset$. We use the sequence $(\boldsymbol{b}_{i;1}^{l-1}, \cdots, \boldsymbol{b}_{i;C_i^{l-1}}^{l-1})$ associated to $N_i^l$ and the sequence $(\boldsymbol{b}_{j;1}^{l-1}, \cdots, \boldsymbol{b}_{j;C_j^{l-1}}^{l-1})$ associated to $N_j^{l-1}$ to construct a new sequence $(\boldsymbol{b}_i^l)$ associated to $N_i^l$ in the following two rules.

- If $\boldsymbol{V}_i \subseteq \boldsymbol{V}_j$(resp. $\boldsymbol{V}_j \subseteq \boldsymbol{V}_i$), then we set $(\boldsymbol{b}_i^l) = (\boldsymbol{b}_i^{l-1})$(resp. $(\boldsymbol{b}_i^l) = (\boldsymbol{b}_j^{l-1})$).
- If $\boldsymbol{V}_i \not\subseteq \boldsymbol{V}_j$ and $\boldsymbol{V}_j \not\subseteq \boldsymbol{V}_i$. Since $\boldsymbol{V}_i \cap \boldsymbol{V}_j \neq \emptyset$ without loss of generality we assume the set $\boldsymbol{V}_i \cap \boldsymbol{V}_j$ is of the form $\{\boldsymbol{b}_{i;1}^{l-1}, \boldsymbol{b}_{i;2}^{l-1}, \cdots, \boldsymbol{b}_{i;k_i}^{l-1}\}$ for some $k_i \leqslant C_i^{l-1}$. Also, there exist $k_j \leqslant C_j^l$ such that $\boldsymbol{b}_{j;k_j}^{l-1} = \boldsymbol{b}_{i;1}^{l-1}$. And therefore the sequence $\boldsymbol{b}_i^l$ is set to be $(\boldsymbol{b}_{j;1}^{l-1}, \boldsymbol{b}_{j;2}^{l-1}, \cdots \boldsymbol{b}_{j;k_j}^{l-1}, \boldsymbol{b}_{i;2}^{l-1}, \boldsymbol{b}_{i;3}^{l-1} \cdots \boldsymbol{b}_{i;C_i^{l-1}}^{l-1})$.

Moreover, for a given node $N_i^{l-1}$ there might exist more than one node $N_j^{l-1}$ satisfying $\boldsymbol{A}_{i,j}^{(l-1)} \geqslant 1$. Denote $\Lambda_i^l = \{j | \boldsymbol{A}_{i,j}^{(l)} \geqslant 1\}$, then the information in each node $N_k^{l-1}$ with $k \in \Lambda_i^l$ is transmitted to $N_i^{l-1}$ as the above way by treating $\boldsymbol{b}_i^l$ as $\boldsymbol{b}_i^{l-1}$ each time. More specifically, we set initially $N_i^l = N_i^{l-1}$ and correspondingly $\boldsymbol{b}_i^l = \boldsymbol{b}_i^{l-1}$. Then for each $k \in \Lambda_i^l$ and for $\boldsymbol{b}_k^{l-1}$ associated to $N_k^{l-1}$, we update $\boldsymbol{b}_i^{(l)}$ as in the above two rules by setting $\boldsymbol{b}_i^{l-1} = \boldsymbol{b}_i^l$ and $\boldsymbol{b}_j^{l-1} = \boldsymbol{b}_k^{l-1}$.

**Remark D.3.** *The information propagation in this transformer satisfies the rules as we defined in section 5. Although we ignore the information propagated from the node $N_1^{l-1}$ by setting $\boldsymbol{A}_{i,1}^{(l-1)} = 0$ for $l \geqslant 1$, there won't be any information loss. Since the first node in each layer will only contain one value which is also contained in $N_2^1$ by Rule 2.*

### D.5 Existence of approximating FNN and error analysis

We find the FNN we required in three steps.

- **Step 1** Find continuous functions that decode the information;
- **Step 2** Extend the continous function to allow small error;
- **Step 3** Use universal approximation theorem to find a FNN to approximate the extended continuous functions.

#### Step 1, Continuous functions on the ideal inputs

Fix the dictionary, sequence length, depth, embeddings, and attention parameters of the construction. Let $\Sigma$ be the finite family of all admissible reasoning inputs of length $n$ over this dictionary. The index $\sigma \in \Sigma$ specifies the input, including its token choices, ordering of facts, and query. For a fixed reasoning chain, this family includes all its admissible orderings. Write

$$Y^l_{\sigma,i} = \boldsymbol{X}^{ao(l)}_{\sigma,i}, \qquad T^{l+1}_{\sigma,i} = \boldsymbol{X}^{(l+1)}_{\sigma,i}, \qquad 0 \le l < L, \tag{66}$$

where $T^{l+1}_{\sigma,i}$ is the ideal binary sparse code prescribed by the information-propagation construction. These ideal states are defined before choosing any approximating FNN.

To define a function of $Y^l_{\sigma,i}$, it is necessary that equal inputs have equal targets. Lemma D.6, proved below directly from the attention inputs, establishes this consistency. In addition, Lemma D.9 gives an inverse image under $R^l_N$ for every binary target. Thus the local pre-normalization map $g^l_i$ is well-defined on the finite set $\mathcal{S}^l_i = \{Y^l_{\sigma,i} : \sigma \in \Sigma\}$ by

$$g^l_i(Y^l_{\sigma,i}) = (R^l_N)^{-1}(T^{l+1}_{\sigma,i}), \qquad R^l_N(g^l_i(Y^l_{\sigma,i})) = T^{l+1}_{\sigma,i}. \tag{67}$$

A function on a finite subset of Euclidean space is continuous in the relative topology. Step 2 extends this map to neighborhoods of the ideal inputs. The map $g^l_i$ denotes the total pre-normalization target; the FNN branch must approximate $g^l_i(y) - y$ to account for the residual in (49).

We use the following form of approximation after normalization.

**Lemma D.4.** *Let $K \subseteq \mathbb{R}^p$ be nonempty and compact, and let $f : K \to \mathbb{R}^{d_m}$ be continuous. Fix RMSNorm with a positive stability constant and a fixed positive scalar gain. For every $\epsilon' > 0$, there exists a single-hidden-layer FNN $f' : \mathbb{R}^p \to \mathbb{R}^{d_m}$, with a fixed continuous sigmoidal activation and sufficiently many hidden units, such that*

$$\sup_{x \in K} \|R_N(f(x)) - R_N(f'(x))\|_\infty < \epsilon'. \tag{68}$$

Here and in the sequel, $\|\cdot\|$ denotes the $\infty$ norm: for $v \in \mathbb{R}^{d_m}$, $\|v\| = \max_{1 \le a \le d_m} |v_a|$. Continuity on the approximation domain is essential; a discontinuous simple function on an arbitrary compact set need not admit uniform approximation by continuous neural networks. In the construction below, the local maps are constant on finitely many separated compact sets and hence continuous on their union.

As discussed in Remark D.1, we use arbitrarily accurate approximations to the prescribed ideal outputs rather than assuming that an FNN can be assigned arbitrary exact outputs on neighborhoods. We next construct the compact domains on which these approximations will be used.

#### Step 2: Extension of the local decoding maps

For $\sigma \in \Sigma$, position $i$, and $0 \le l < L$, define the ideal input set independently of any decoding function:

$$D^{l+1}_{\sigma,i} = \{Y^l_{\sigma,i}\} \subseteq \mathbb{R}^{d_m}. \tag{69}$$

We group inputs according to their desired output at the next layer.

**Definition D.5.** *For fixed $i$ and $l$, two inputs $\sigma, \tau \in \Sigma$ are equivalent if $T^{l+1}_{\sigma,i} = T^{l+1}_{\tau,i}$. Their equivalence classes and the family of all classes are denoted by*

$$[\sigma]^{l+1}_i = \{\tau \in \Sigma : T^{l+1}_{\tau,i} = T^{l+1}_{\sigma,i}\}, \qquad E^{l+1}_i = \Sigma/\!\sim^{l+1}_i. \tag{70}$$

*For a class $c = [\sigma]_i^{l+1}$, let $T_{c,i}^{l+1} = T_{\sigma,i}^{l+1}$ and*

$$D_{c,i}^{l+1} = \bigcup_{\tau \in c} D_{\tau,i}^{l+1}. \tag{71}$$

The union in (71) includes every input in the class, not just a representative. All these sets are finite because $\Sigma$ is finite.

**Lemma D.6.** *For any $\sigma, \tau \in \Sigma$, fixed position $i$, and $0 \le l < L$,*

$$Y_{\sigma,i}^{l} = Y_{\tau,i}^{l} \quad \Longrightarrow \quad T_{\sigma,i}^{l+1} = T_{\tau,i}^{l+1}. \tag{72}$$

*Consequently, ideal input sets belonging to different target classes are disjoint.*

If $E_i^{l+1}$ has at least two classes, define

$$d_i^{l+1} = \min_{\substack{c,c' \in E_i^{l+1},\ c \neq c' \\ x \in D_{c,i}^{l+1},\ y \in D_{c',i}^{l+1}}} \|x - y\|_\infty > 0. \tag{73}$$

If there is only one target class, set $d_i^{l+1} = 1$; no separation between different targets is needed in that case. Choose

$$0 < \delta_i^{l+1} < \frac{d_i^{l+1}}{3}, \tag{74}$$

and form the compact class neighborhoods and their union:

$$\begin{aligned} K_{c,i}^{l} &= D_{c,i}^{l+1} + [-\delta_i^{l+1}, \delta_i^{l+1}]^{d_m}, \\ K_i^{l} &= \bigcup_{c \in E_i^{l+1}} K_{c,i}^{l}. \end{aligned} \tag{75}$$

Here the addition is Minkowski addition in $\mathbb{R}^{d_m}$. For different classes, the distance between these neighborhoods is at least $d_i^{l+1} - 2\delta_i^{l+1} > 0$, so they cannot intersect. Neighborhoods of centers within the same class may overlap, since they require the same target.

Define the extended pre-normalization map on $K_i^l$ by

$$\begin{aligned} \tilde{f}_i^{l}(y) &= (R_N^l)^{-1} \left( \sum_{c \in E_i^{l+1}} T_{c,i}^{l+1} \mathbf{1}_{K_{c,i}^{l}}(y) \right), \qquad y \in K_i^l, \\ &= (R_N^l)^{-1}(T_{c,i}^{l+1}) \qquad \text{when } y \in K_{c,i}^l. \end{aligned} \tag{76}$$

At each point of $K_i^l$, exactly one class indicator is active. Its target is a binary vector in the range of RMSNorm. Hence (76) is well-defined, is constant on each class neighborhood, and extends $g_i^l$. The finitely many class neighborhoods are separated compact sets, so $\tilde{f}_i^l$ is continuous on their union. No continuity of the displayed indicator formula outside $K_i^l$ is asserted or needed; a continuous extension to the ambient space will be used for neural-network approximation.

### ONE FNN SHARED ACROSS POSITIONS

The position index on the local decoding functions does not require different network parameters at different positions. The receiver's position can be identified from its attention-residual input. We establish this for the ideal states first, and then choose disjoint neighborhoods on which the local functions can be joined continuously.

**Proposition D.7.** *Fix a layer and the finite family of admissible length-$n$ inputs over the chosen dictionary. The compact neighborhoods of the ideal FNN inputs at different positions can be chosen pairwise disjoint. The continuous local pre-normalization maps constructed in Step 2 can then be approximated, after the FNN residual and RMSNorm, by one position-wise FNN shared by all positions in that layer.*

*Proof.* Put $B = 3^L$ and $h = (B-1)/2$. For $i \geq 2$, all coordinates that can occur in the ideal sparse code at position $i$ belong to

$$\begin{aligned} \mathcal{B}_i &= \bigcup_{t=1}^{d} \{k_t - iB - r : r \in \mathbb{Z},\ |r| \leq h\}, \quad 2 \leq i \leq n, \\ \mathcal{B}_1 &= \{k_t - (n+1)B : 1 \leq t \leq d\}. \end{aligned} \tag{77}$$

Here the shift within a chain is its token rank minus the rank of the original token, as in (49). The information-quantity bound gives a maximum rank difference of $3^{l-1}$ at hidden layer $l \leq L$, which is at most $h$. The full-dictionary spacing in (42) prevents collisions between different token embeddings and prevents wrap-around. For the same token, regular position-block centers are $B$ apart, while $2h = B - 1 < B$; the special block $\mathcal{B}_1$ is also separated from them. Thus $\mathcal{B}_i \cap \mathcal{B}_j = \varnothing$ whenever $i \neq j$, uniformly over all token choices and permutations.

For $1 \leq l < L$, write $w_{ij}^{(l)}$ for the causal softmax weights. Since $W^{vo(l)} = I$, the ideal input to the next FNN is

$$Y_i = \boldsymbol{X}_i^{ao(l)} = (1 + w_{ii}^{(l)})\boldsymbol{X}_i^{(l)} + \sum_{j<i} w_{ij}^{(l)} \boldsymbol{X}_j^{(l)}. \tag{78}$$

The ideal codes have coefficients zero or one and nonempty, disjoint supports. At every coordinate of the receiver's own support, $Y_i$ therefore equals $1 + w_{ii}^{(l)} > 1$. At every other coordinate it is either zero or a weight $w_{ij}^{(l)} < 1$. For $i = 1$, $w_{11}^{(l)} = 1$ and $Y_1 = 2\boldsymbol{X}_1^{(l)}$, so the same conclusion holds. Hence

$$\varnothing \neq \{a : (Y_i)_a > 1\} = \operatorname{supp}(\boldsymbol{X}_i^{(l)}) \subseteq \mathcal{B}_i. \tag{79}$$

For inputs from different positions, these sets of coordinates lie in disjoint blocks. In particular, the input vectors cannot be equal, even when they arise from different sequences.

The first layer has a separate position marker. With $\boldsymbol{p}_i = e_i$, its residual contributes 1 at positional coordinate $i$. The shifted positional contributions from indices $j \leq i$ occupy coordinates $j - 1 < i$, except that $e_1 R = e_{d_m}$. Token coordinates do not enter the first $n$ coordinates. Thus $Y_i^0$ is zero at positional coordinates greater than $i$, and equals 1 at coordinate $i$. If $i > j$, $Y_i^0$ and $Y_j^0$ differ by 1 at coordinate $i$. The first-layer inputs are therefore separated as well; no claim of a coefficient greater than 1 is needed for that layer.

Let $\mathcal{S}_i^l$ denote the finite set of ideal attention-residual inputs at position $i$. The preceding separation implies

$$\gamma_l = \min_{i \neq j} \min_{x \in \mathcal{S}_i^l,\, y \in \mathcal{S}_j^l} \|x - y\|_\infty > 0. \tag{80}$$

Choose a common radius $0 < \rho_l < \min\{\gamma_l/3, \min_i \delta_i^{l+1}\}$ and replace each $\delta_i^{l+1}$ by $\rho_l$ in (75) and (76). The resulting compact sets $K_i^l$ are pairwise disjoint, since their distance is at least $\gamma_l - 2\rho_l > 0$. Within each position, different target classes also remain separated because the radii have only decreased. All subsequent uses of $\delta_i^{l+1}$ refer to these reduced radii. They are chosen from the fixed ideal inputs before any approximation error is selected. Separation is used on neighborhoods, so the implemented network need not perform a discontinuous threshold test at 1.

Write $g_i^l = \tilde{f}_i^l|_{K_i^l}$ for the continuous extension of the local map in Step 1 to these reduced neighborhoods. Its normalized output equals the required target throughout each class neighborhood. On $K^l = \bigcup_i K_i^l$, define

$$F^l(y) = g_i^l(y) - y \qquad (y \in K_i^l). \tag{81}$$

Because the finitely many compact sets are disjoint, this is a single well-defined continuous function on $K^l$. The subtraction of $y$ accounts for the FNN residual: $R_N(F^l(y) + y) = R_N(g_i^l(y))$. Extend each coordinate of $F^l$ continuously from the closed set $K^l$ to $\mathbb{R}^{d_m}$. By universal approximation, with a fixed continuous sigmoidal activation and sufficiently many hidden units, one FNN $\widehat{F}^l$ approximates this extension uniformly on $K^l$. Continuity of RMSNorm with $\epsilon > 0$, uniformly on a compact neighborhood of the relevant pre-normalization vectors, then gives, for every $\eta > 0$,

$$\max_i \sup_{y \in K_i^l} \left\| R_N(\widehat{F}^l(y) + y) - R_N(g_i^l(y)) \right\|_\infty < \eta. \tag{82}$$

Thus one set of FNN parameters serves all positions. Different layers may still have different FNN parameters, as in the Transformer architecture. □

**STEP 3**, APPROXIMATING FNN

By Proposition D.7, choose one shared FNN $\widehat{F}^l$ on the compact union of the position domains, with normalized-output error less than $\min_i \eta_i^{l+1}$. In the formulas below that write $R_N^l(\hat{f}_i^l(y))$ without displaying the residual separately, $\hat{f}_i^l(y)$ denotes the total pre-normalization map $\widehat{F}^l(y)+y$ restricted to the position-$i$ domain. Thus the actual FNN branch is the same $\widehat{F}^l$ at every position. With the neighborhoods shrunk as above, we have

$$\sup_{\boldsymbol{X}\in D_{\sigma,i}^{l+1}+[-\delta_i^{l+1},\delta_i^{l+1}]^{d_m}} \|R_N^l(\hat{f}_i^l(\boldsymbol{X})) - R_N^l(\tilde{f}_i^l(\boldsymbol{X}))\|_\infty < \eta_i^{(l+1)}. \tag{83}$$

The shared FNN branch $\widehat{F}^l$ therefore realizes the required position-wise approximations. In fact we have the following proposition.

**Proposition D.8.** *With the local decoding neighborhoods constructed above, for any prescribed output tolerance the shared FNNs can be chosen so that their normalized outputs approximate the ideal hidden states at every position and layer within that tolerance. This remains valid when the FNN inputs include the errors from the preceding layers.*

*Proof.* We estimate the errors in the attention scores, softmax weights, and attention-residual inputs explicitly. All vector norms below are the $\infty$ norm. The approximation tolerance $\epsilon$ used here is distinct from the fixed positive stability constant in RMSNorm. When $L = 1$, only the first-layer approximation argument at the end of the proof is needed; the estimates below concern $L \geq 2$.

For $1 \leq l < L$, write the perturbed hidden states as

$$\bar{\boldsymbol{X}}_{\sigma,i}^{(l)} = \boldsymbol{X}_{\sigma,i}^{(l)} + \epsilon_i^{(l)} \boldsymbol{r}_i^{(l)}, \qquad 0 \leq \epsilon_i^{(l)} \leq \epsilon, \quad \|\boldsymbol{r}_i^{(l)}\|_\infty \leq 1. \tag{84}$$

Put $K = (n+1)3^L$ and $H = \sum_{m=-K}^{-1} \boldsymbol{R}^m$. For brevity, write $X_i = \boldsymbol{X}_{\sigma,i}^{(l)}$ in the score calculation. Only unmasked entries $1 \leq j \leq i \leq n$ are considered; masked positions always have zero attention weight. Expanding the perturbed score gives all three error terms:

$$\begin{aligned} \bar{A}_{i,j}^{(l)} &= (X_i + \epsilon_i^{(l)} \boldsymbol{r}_i^{(l)}) H (X_j + \epsilon_j^{(l)} \boldsymbol{r}_j^{(l)})^\mathsf{T} \\ &= A_{i,j}^{(l)} + \epsilon_i^{(l)} \boldsymbol{r}_i^{(l)} H X_j^\mathsf{T} + \epsilon_j^{(l)} X_i H (\boldsymbol{r}_j^{(l)})^\mathsf{T} \\ &\quad + \epsilon_i^{(l)} \epsilon_j^{(l)} \boldsymbol{r}_i^{(l)} H (\boldsymbol{r}_j^{(l)})^\mathsf{T}. \end{aligned} \tag{85}$$

Every $\boldsymbol{R}^m$ is a permutation matrix, so $H$ has nonnegative entries and $\sum_{a,b} H_{ab} = d_m K$. The ideal codes satisfy $\|X_i\|_\infty \leq 1$. Thus the first two error terms are of first order, and

$$\begin{aligned} |\bar{A}_{i,j}^{(l)} - A_{i,j}^{(l)}| &\leq d_m K \big(\epsilon_i^{(l)} + \epsilon_j^{(l)} + \epsilon_i^{(l)} \epsilon_j^{(l)}\big) \\ &\leq d_m K (2\epsilon + \epsilon^2) \leq \eta_0, \end{aligned} \tag{86}$$

provided $0 < \epsilon \leq \min\{1, \eta_0/(3 d_m K)\}$. In particular, the score error still tends to zero as $\epsilon \to 0$.

Define the normalizing sums and causal attention weights by

$$\begin{aligned} Z_i^{(l)} &= \sum_{t=1}^{i} e^{A_{i,t}^{(l)}}, \quad \bar{Z}_i^{(l)} = \sum_{t=1}^{i} e^{\bar{A}_{i,t}^{(l)}}, \\ w_{ij}^{(l)} &= \frac{e^{A_{i,j}^{(l)}}}{Z_i^{(l)}}, \qquad \bar{w}_{ij}^{(l)} = \frac{e^{\bar{A}_{i,j}^{(l)}}}{\bar{Z}_i^{(l)}}. \end{aligned} \tag{87}$$

Since the value projection is the identity for these layers, the perturbed attention-residual input is

$$\begin{aligned} \bar{\boldsymbol{X}}_{\sigma,i}^{ao(l)} &= \sum_{j=1}^{i} \bar{w}_{ij}^{(l)} \bar{\boldsymbol{X}}_{\sigma,j}^{(l)} + \bar{\boldsymbol{X}}_{\sigma,i}^{(l)} \\ &= \sum_{j=1}^{i} \bar{w}_{ij}^{(l)} (\boldsymbol{X}_{\sigma,j}^{(l)} + \epsilon_j^{(l)} \boldsymbol{r}_j^{(l)}) + \boldsymbol{X}_{\sigma,i}^{(l)} + \epsilon_i^{(l)} \boldsymbol{r}_i^{(l)}. \end{aligned} \tag{88}$$

Subtracting the ideal attention-residual input yields

$$\begin{aligned}\bar{\boldsymbol{X}}_{\sigma,i}^{ao(l)} - \boldsymbol{X}_{\sigma,i}^{ao(l)} = &\sum_{j=1}^{i} (\bar{w}_{ij}^{(l)} - w_{ij}^{(l)}) \boldsymbol{X}_{\sigma,j}^{(l)} \\ &+ \sum_{j=1}^{i} \bar{w}_{ij}^{(l)} \epsilon_j^{(l)} \boldsymbol{r}_j^{(l)} + \epsilon_i^{(l)} \boldsymbol{r}_i^{(l)}.\end{aligned} \tag{89}$$

The ideal hidden states at distinct positions have disjoint supports and coefficients zero or one, as established in Proposition D.7. Consequently,

$$\|\bar{\boldsymbol{X}}_{\sigma,i}^{ao(l)} - \boldsymbol{X}_{\sigma,i}^{ao(l)}\|_\infty \le \mathrm{I} + \mathrm{II}, \tag{90}$$

where

$$\mathrm{I} = \max_{l,\, 1\le j\le i\le n} |\bar{w}_{ij}^{(l)} - w_{ij}^{(l)}|, \tag{91}$$

$$\mathrm{II} = \left\| \sum_{j=1}^{i} \bar{w}_{ij}^{(l)} \epsilon_j^{(l)} \boldsymbol{r}_j^{(l)} + \epsilon_i^{(l)} \boldsymbol{r}_i^{(l)} \right\|_\infty. \tag{92}$$

Choose $0 < \eta_0 \le \log 2$, so that $|e^x - 1| \le 2|x|$ for $|x| \le \eta_0$. Set $M = d_m K + 1$, which bounds the absolute values of both the ideal and perturbed unmasked scores under (86). Hence $\bar{Z}_i^{(l)} \ge e^{-M}$ and $|e^{\bar{A}_{i,j}^{(l)}} - e^{A_{i,j}^{(l)}}| \le 2\eta_0 e^M$. Keeping the two contributions from the numerator and denominator separate, we obtain

$$\begin{aligned}\mathrm{I} &\le \max_{l,\, j\le i} \left\{ \frac{|e^{\bar{A}_{i,j}^{(l)}} - e^{A_{i,j}^{(l)}}|}{\bar{Z}_i^{(l)}} + \frac{e^{A_{i,j}^{(l)}}}{Z_i^{(l)}} \frac{|Z_i^{(l)} - \bar{Z}_i^{(l)}|}{\bar{Z}_i^{(l)}} \right\} \\ &\le 2\eta_0 e^{2M} + 2n\eta_0 e^{2M} \le 4n\eta_0 e^{2M},\end{aligned} \tag{93}$$

$$\mathrm{II} \le \sum_{j=1}^{i} \bar{w}_{ij}^{(l)} \epsilon + \epsilon = 2\epsilon \le (n+1)\epsilon. \tag{94}$$

Combining (89), (93), and (94) gives the attention-residual error bound:

$$\|\bar{\boldsymbol{X}}_{\sigma,i}^{ao(l)} - \boldsymbol{X}_{\sigma,i}^{ao(l)}\|_\infty \le 4n\eta_0 e^{2M} + (n+1)\epsilon. \tag{95}$$

Let $\delta_* = \min_{i,\, 0\le l<L} \delta_i^{l+1} > 0$ be a common lower bound for the admissible neighborhood radii after the shrinking in Proposition D.7. A sufficient choice is

$$\begin{aligned}0 < \eta_0 &< \min\left\{ \log 2, \frac{\delta_*}{8ne^{2M}} \right\}, \\ 0 < \epsilon &< \min\left\{ 1, \frac{\eta_0}{3d_m K}, \frac{\delta_*}{2(n+1)} \right\}.\end{aligned} \tag{96}$$

The tolerance $\epsilon$ may also be taken smaller than any prescribed output tolerance. These choices imply both (86) and

$$\|\bar{\boldsymbol{X}}_{\sigma,i}^{ao(l)} - \boldsymbol{X}_{\sigma,i}^{ao(l)}\|_\infty < \delta_* \le \delta_i^{l+1}. \tag{97}$$

Thus the perturbed FNN input remains in the same admissible decoding neighborhood as its ideal counterpart. Choose the normalized-output approximation errors in Step 3 to satisfy $\eta_i^{l+1} < \epsilon$ for all positions and layers. Since the extended local decoder has the same ideal target on this neighborhood, the shared FNN gives

$$R_N^l\big(\hat{f}_i^l(\bar{\boldsymbol{X}}_{\sigma,i}^{ao(l)})\big) \in \{\boldsymbol{X}_{\sigma,i}^{(l+1)}\} + [-\epsilon, \epsilon]^{d_m}. \tag{98}$$

As specified in Step 3, $\hat{f}_i^l(y) = \widehat{F}^l(y) + y$ includes the FNN residual. The first layer starts from exact embeddings and can attain the same error tolerance by uniform approximation. Induction over the remaining finitely many layers now applies (95) and (98) at each step. The two first-order terms change the required approximation accuracy, but do not prevent it from being chosen sufficiently small. □

**Lemma D.9.** *For $x \in \mathbb{R}^{d_m}$, define its mean square by*

$$q(x) = \frac{1}{d_m} \sum_{i=1}^{d_m} (x^i)^2. \tag{99}$$

*RMSNorm with a scalar gain $\alpha > 0$, zero bias, and a stability constant $\epsilon > 0$ is*

$$R_N(x) = \alpha \frac{x}{\sqrt{q(x) + \epsilon}}. \tag{100}$$

*This map is injective. Its range consists of the vectors $z$ satisfying $q(z) < \alpha^2$, and its inverse on that range is*

$$R_N^{-1}(z) = \sqrt{\frac{\epsilon}{\alpha^2 - q(z)}}\, z. \tag{101}$$

*In the construction we choose $\alpha > 1$, so that every vector in $\{0,1\}^{d_m}$ belongs to this range.*

***Proof of Lemma D.9.*** The denominator in (100) is strictly positive for every input. Suppose that $R_N(x_1) = R_N(x_2)$. Written coordinate by coordinate, this means

$$\alpha \frac{x_1^i}{\sqrt{q(x_1) + \epsilon}} = \alpha \frac{x_2^i}{\sqrt{q(x_2) + \epsilon}}, \qquad 1 \le i \le d_m. \tag{102}$$

If $x_2 = 0$, these equations imply $x_1 = 0$ as well. For all other inputs, we can cancel the nonzero scalar $\alpha$ and define the positive number

$$c = \frac{\sqrt{q(x_1) + \epsilon}}{\sqrt{q(x_2) + \epsilon}} > 0. \tag{103}$$

Equation (102) then gives

$$x_1^i = c x_2^i \quad (1 \le i \le d_m), \qquad x_1 = c x_2. \tag{104}$$

Taking the mean square of both sides yields

$$q(x_1) = \frac{1}{d_m} \sum_{i=1}^{d_m} (c x_2^i)^2 = c^2 q(x_2). \tag{105}$$

On the other hand, squaring (103) and clearing the positive denominator gives

$$c^2 \big(q(x_2) + \epsilon\big) = q(x_1) + \epsilon. \tag{106}$$

Substituting the preceding mean-square identity, we obtain

$$c^2 q(x_2) + c^2 \epsilon = c^2 q(x_2) + \epsilon, \qquad (c^2 - 1)\epsilon = 0. \tag{107}$$

Since $\epsilon > 0$ and $c > 0$, it follows that $c = 1$. Equation (104) therefore gives $x_1 = x_2$, proving injectivity. The positive stability constant is essential to this argument.

To determine the range, take the mean square of the normalized output:

$$q(R_N(x)) = \frac{\alpha^2 q(x)}{q(x) + \epsilon} < \alpha^2. \tag{108}$$

Conversely, suppose that $q(z) < \alpha^2$ and put $u = \sqrt{\epsilon/(\alpha^2 - q(z))}\, z$. Direct calculation gives

$$q(u) + \epsilon = \frac{\epsilon q(z)}{\alpha^2 - q(z)} + \epsilon = \frac{\epsilon \alpha^2}{\alpha^2 - q(z)}. \tag{109}$$

Using $\alpha > 0$ in (100), we thus have

$$R_N(u) = \frac{\alpha \sqrt{\epsilon/(\alpha^2 - q(z))}\, z}{\sqrt{\epsilon \alpha^2/(\alpha^2 - q(z))}} = z. \tag{110}$$

This proves the range statement and the inverse formula. Finally, a binary vector satisfies $q(z) \le 1$; choosing $\alpha > 1$ therefore supplies a preimage for every binary target used by the construction. □

***Proof of Lemma D.6***. We show directly that the attention-residual input determines the prescribed next-layer target. This argument uses only the ideal sparse codes and the fixed attention parameters, before any approximating FNN is chosen.

**First-layer inputs ($l = 0$).** Let $t_j$ denote the dictionary index of the token at input position $j$, so its embedding is $e_{k_{t_j}}$. By the embedding spacing, the token coordinates $k_t$, shifted token coordinates $k_t - 1$, and positional coordinates do not collide. The residual therefore identifies the original token at position $i$ through

$$(Y^0_{\sigma,i})_{k_t} = \mathbf{1}_{\{t=t_i\}}. \tag{111}$$

For odd $i$, the first-layer target contains only this original token, with the prescribed position shift (including the special shift at $i = 1$). Thus equality of the FNN inputs implies equality of their targets in this case.

For even $i$, the target encodes the ordered pair $(t_{i-1}, t_i)$, so we must also recover the preceding token. Under the unscaled-score convention stated in the architecture section, the only causal score equal to 1 in this row is the score at $j = i - 1$; all other scores are 0. The conventional factor $1/\sqrt{d_k}$ can be retained explicitly by multiplying the displayed query matrices by $\sqrt{d_k}$, giving the same effective scores. The softmax denominator is consequently $Z_i = i - 1 + e$, independent of the token choices. If

$$N_t = |\{j : 1 \le j \le i,\ t_j = t\}|, \tag{112}$$

then the coefficient at the shifted token coordinate satisfies

$$Z_i(Y^0_{\sigma,i})_{k_t-1} = N_t + (e-1)\mathbf{1}_{\{t=t_{i-1}\}}. \tag{113}$$

The counts $N_t$ are integers. Since $e-1$ is irrational, the unique noninteger value in (113), as $t$ ranges over the dictionary, identifies $t_{i-1}$. Equivalently, if two inputs had different predecessor tokens but identical attention-residual vectors, equality at one of these coordinates would equate an integer with an integer plus $e - 1$, a contradiction. Together with (111), this determines the entire first-layer target, even when token occurrences in the prefix repeat.

**Subsequent inputs ($1 \le l < L$).** Use the coordinate blocks $\mathcal{B}_j$ in (77). Their separation follows solely from the fixed embedding spacing and the rank-offset bound, as verified in the geometric part of the proof of Proposition D.7; it does not depend on the existence of the local maps or of an FNN.

All causal softmax weights are strictly positive. The ideal codes are binary and their supports lie in disjoint position blocks. Consequently, for every source position $j \le i$, the input vector determines its full ideal code through

$$(\boldsymbol{X}^{(l)}_{\sigma,j})_a = \mathbf{1}_{\{a\in\mathcal{B}_j,\ (Y^l_{\sigma,i})_a>0\}}, \qquad 1 \le a \le d_m. \tag{114}$$

This formula applies to the receiver as well, since its nonzero coefficients in $Y^l_{\sigma,i}$ equal $1 + w^{(l)}_{ii} > 0$. At position 1 it recovers the sentinel code in the same way.

The coordinates of each recovered regular code specify its tokens and their offsets relative to the original token: a coordinate $k_t - j3^L - r$ identifies token $t$ and offset $r$. The token at offset zero is the original token, and sorting offsets gives the encoded chain order. The sentinel code identifies its single token. Hence the recovered prefix codes determine which earlier chains match the receiver, their relative order, and the prescribed next-layer target. More explicitly, each matching chain shares a token with the receiver; aligning their offsets at that token fixes its placement in the merged chain. The admissible reasoning inputs use a common chain, so these alignments agree on shared tokens. The union with the receiver then determines the target code, anchored at the receiver's original token.

Thus if $Y^l_{\sigma,i} = Y^l_{\tau,i}$, (114) gives the same ordered prefix codes for both inputs and therefore the same next-layer target. This proves (72) for every layer. Different targets must consequently have different ideal FNN inputs; finiteness yields the positive distances used in (73). □

***Proof of Lemma D.4***. Since $f$ is continuous on the compact set $K$, it is bounded there. Extend each coordinate continuously from the closed set $K$ to $\mathbb{R}^p$. Theorem D.10 applied to this extension gives, for every $\epsilon_0 > 0$, a single-hidden-layer FNN $f'$ such that

$$\sup_{x\in K} \|f(x) - f'(x)\|_\infty < \epsilon_0. \tag{115}$$

We now bound the change produced by RMSNorm. The norm in this argument is the maximum norm, as fixed above. Put $M = \sup_{u\in K} \|f(u)\|_\infty$ and restrict $0 < \epsilon_0 \le 1$. For $x = f(u)$ and $y = f'(u)$, we then have

$$\|x - y\|_\infty < \epsilon_0, \qquad \|x\|_\infty \le M, \qquad \|y\|_\infty \le M + 1. \tag{116}$$

Writing $a = \sqrt{q(x) + \epsilon}$ and $b = \sqrt{q(y) + \epsilon}$, we have $a, b \ge \sqrt{\epsilon}$. The normalized difference can be decomposed without subtracting a coordinate mean:

$$\begin{aligned} R_N(x) - R_N(y) &= \alpha \left( \frac{x}{a} - \frac{y}{b} \right) \\ &= \alpha \left( \frac{x - y}{a} + y \left( \frac{1}{a} - \frac{1}{b} \right) \right). \end{aligned} \tag{117}$$

Consequently,

$$\|R_N(x) - R_N(y)\|_\infty \le |\alpha| \left( \frac{\|x - y\|_\infty}{\sqrt{\epsilon}} + \|y\|_\infty \frac{|a - b|}{\epsilon} \right). \tag{118}$$

To estimate the second term, use the definition of the mean square in (99):

$$\begin{aligned} |q(x) - q(y)| &\le \frac{1}{d_m} \sum_{i=1}^{d_m} |x^i - y^i| \, |x^i + y^i| \\ &\le \|x - y\|_\infty (\|x\|_\infty + \|y\|_\infty) \\ &\le \epsilon_0 (2M + 1). \end{aligned} \tag{119}$$

The difference between the two square roots is obtained by multiplying by their sum:

$$\begin{aligned} |a - b| &= \frac{|q(x) - q(y)|}{\sqrt{q(x) + \epsilon} + \sqrt{q(y) + \epsilon}} \\ &\le \frac{\epsilon_0 (2M + 1)}{2\sqrt{\epsilon}}. \end{aligned} \tag{120}$$

Combining these inequalities gives the explicit bound

$$\begin{aligned} \|R_N(x) - R_N(y)\|_\infty &\le |\alpha| \epsilon_0 \left( \frac{1}{\sqrt{\epsilon}} + \frac{(M + 1)(2M + 1)}{2\epsilon^{3/2}} \right) \\ &= C_{M,\alpha,\epsilon} \epsilon_0. \end{aligned} \tag{121}$$

Here $C_{M,\alpha,\epsilon}$ is finite and does not depend on the point $u$ or on the approximating network. Given the desired tolerance $\epsilon' > 0$, choose

$$0 < \epsilon_0 < \min\{1, \epsilon'/C_{M,\alpha,\epsilon}\}. \tag{122}$$

The preceding uniform approximation then yields

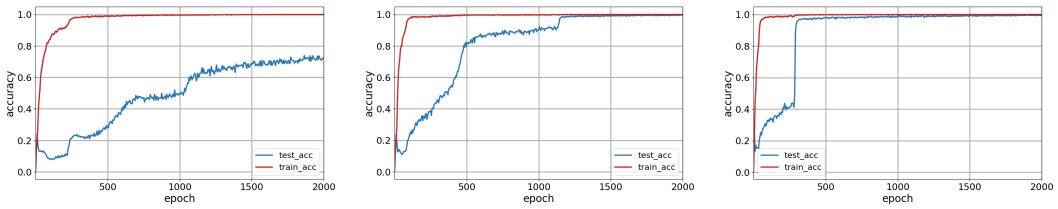
$$\sup_{u\in K} \|R_N(f(u)) - R_N(f'(u))\|_\infty \le C_{M,\alpha,\epsilon} \epsilon_0 < \epsilon'.$$

□

**Theorem D.10** (Universal Approximation Theorem(Cybenko, 1989))**.** *Fix a continuous sigmoidal hidden activation. For any continuous function $f : \mathbb{R}^d \to \mathbb{R}^n$, compact set $K \subseteq \mathbb{R}^d$, and error tolerance $\epsilon > 0$, there exists a single-hidden-layer neural network $f_\theta$ with biases, a linear output layer, and sufficiently many hidden units, such that:*

$$\sup_{x\in K} ||f(x) - f_\theta(x)||_\infty < \epsilon,$$

*where $K \subseteq \mathbb{R}^d$ is an arbitrary compact set.*




(a) $d_m = 256$ (b) $d_m = 512$ (c) $d_m = 1024$

Figure 8: Training curves of the 3-layer Transformer on 3-step reasoning under different hidden dimensions.

# E Details of the experiment

## E.1 Dataset

We require reasoning sequence $(x_i)_{1 \le i \le 2s}$ of the training set satisfy the following condition.

$$x_{2i} - x_{2i-1} \bmod 5 \in \{0, 1, 4\} \tag{123}$$

The sequence of the test set satisfy:

$$x_{2i} - x_{2i-1} \bmod 5 \in \{2, 3\} \tag{124}$$

The values of each token range from 20 to 100,i.e.,$x_i \in [20, 100]$.

## E.2 Hyperparameters

In this section, the fixed and tunable hyperparameters employed in the model are outlined.

The fixed hyperparameters are as follows. Transformer architecture uses one attention head per layer. The dataset is partitioned into a training set comprising 90% of the data and a test set comprising the remaining 10%. Training is conducted over 2000 epochs. For embedding initialization, use $N(0, 1)$; for all other parameters, use Kaiming uniform initialization with a power of 1 for small initialization. A weight decay of 0.1 is applied. The dimension of the model $d_m$ is set equal to the key dimension $d_k$. The feed-forward network dimension $d_{feedforward}$ is set to 1200. The AdamW optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) was employed, together with a gradual warmup scheduler followed by cosine annealing learning rate decay.

Table 1

| the number of reasoning steps | 3 | 4 | 5 |
|---|---|---|---|
| the size of datasets | 1200000 | 6000000 | 30000000 |

## E.3 Lower Bound: Experimental Results of a 3-Layer Transformer on 3-Step Reasoning

We train a 3-layer Transformer on sequences of length 21, using a batch size of 1000 and a learning rate of $5e{-}5$, to perform 3-step reasoning. Fig. 8 shows our training results. As the hidden dimension $d_m$ increases, the model generalizes more easily during training.

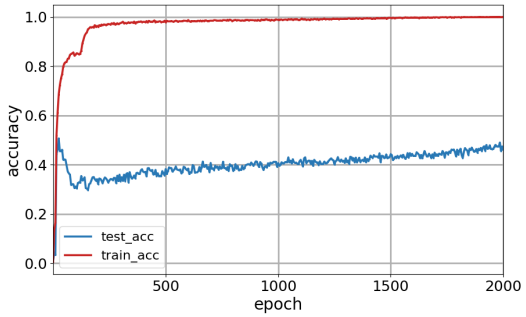



Figure 9: Training curves of the 3-layer Transformer on 4-step reasoning

### E.4 Lower Bound: The size of $d_m$ required by a 3-Layer Transformer on 3-Step Reasoning

We observe that the model dimension $d_m$ in our construction is notably large and a higher hidden dimension aids in storing more intermediate information. To validate these findings, we investigate whether a large $d_m$ is indeed necessary. Our experiments confirm that a sufficiently large $d_m$ is critical for achieving optimal performance. The relationship between the model dimension $d_m$ and test accuracy is summarized in the table below.

Table 2: Relationship between $d_m$ and test accuracy

| $d_m$ | 64 | 128 | 256 | 512 | 1024 |
|---|---|---|---|---|---|
| Test accuracy (%) | 9.6 | 11.9 | 72.2 | 99.6 | 99.6 |

As shown in Table 2, test accuracy increases monotonically with dimension $d_m$, eventually approaching $100\%$. It confirms that a large $d_m$ is indeed necessary.

### E.5 Upper Bound: Experimental Results of a 3-Layer Transformer on 4-Step Reasoning

We train a 3-layer Transformer with $d_m = 1024$ on sequences of length 25, using a batch size of 1000 and a learning rate of $5e-5$, to perform 4-step reasoning. Fig. 9 shows our training results.

From Fig. 9, we observe that the test accuracy remains around 35%–40%. In fact, this is fully consistent with our theoretical expectations. Our theory indicates that a Transformer exhibits generalization only on data that satisfy a specific order structure.

As shown in Fig. 10, we draw out 1000 sequences from canonical sequences and non-canonical sequences to calculate accuracy. And we found that canonical sequences exhibit consistently high accuracy, while non-canonical sequences display substantially lower accuracy, thereby providing empirical support for our theoretical predictions.

### E.6 Causal intervention experiment

In this section, we investigate whether transformer is capable of genuine reasoning or merely memorizes the answers, under the settings of 4-step and 5-step reasoning. We then describe the experimental methodology employed to obtain the results.

First, a sequence that can be answered correctly will be selected. Subsequently, a specific attention line or residual connection is masked. If transformer produces an incorrect output after the masking of a particular attention line or residual connection, that line will be marked in blue. If the model's output remains correct, the line will left unchanged. The resulting attention graph retains only those connections that critically influence the outcome. This approach allows for conclusions to be drawn regarding whether the model has learned to perform reasoning.

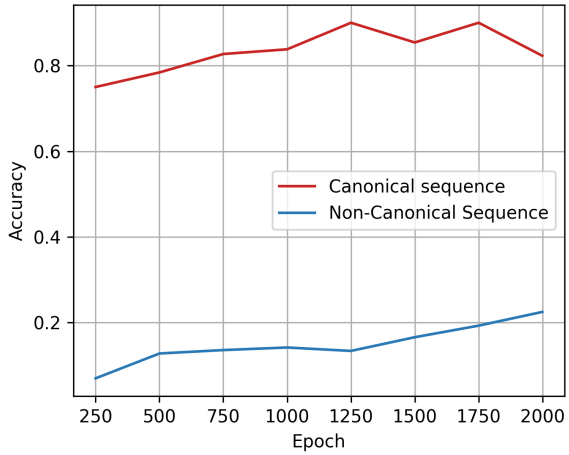



Figure 10: Test accuracy of sentences with different sequence relationships

To further support the validity of our theoretical analysis, we provide two additional contrastive plots (Fig.11). Both plots use inputs containing the same reasoning chain, except that the order of the chain elements is arranged differently. According to our theory, one particular ordering enables the model to aggregate sufficient information to arrive at the correct answer, whereas the alternative ordering prevents the model from collecting enough information to respond correctly. The experimental results align perfectly with these theoretical predictions: when (67, 74) is placed after (74, 42) and (59, 67), the critical attention path propagates exactly along the route predicted by our theory. However, if we change the relative order of these three arrays so that (67, 74) is no longer positioned after (74, 42) and (59, 67), the model's key attention patterns become chaotic and disorganized.

### E.7 Experimental Results of a 2-Layer Transformer

Table 3

| the number of reasoning steps | 1 | 2 |
|---|---|---|
| test accuracy(%) | 100 | 8.4 |

The experiment involves training a two-layer, single-headed transformer model with the following hyperparameter configuration: a model dimension ($d_m$) and key/value dimension ($d_k$) of 1024, a feedforward dimension ($d_{ffn}$) of 1200, a sequence length of 25, and training for 2000 epochs.

According to Corollary 6.4, a 2-layer transformer can theoretically solve at most one step of reasoning. The empirical results presented in Table 3 support the theoretical predictions: The model achieves complete accuracy on the one-step reasoning test set, which verifies the theoretical prediction for one-step reasoning. And a low test accuracy of $8.4\%$ is observed on the two-step reasoning test set, which indicates that the two-layer transformer's performance is not significantly better than random guessing, thereby validating the theoretical limitation for two-step reasoning.

### E.8 Guess about $d_m$

Based on the aforementioned experiments, it can be observed that training a model capable of parallel reasoning—where the number of reasoning steps exceeds the depth of the transformer (i.e., number of layers minus one)—requires a substantially large model dimension $d_m$. It is therefore hypothesized that for string reasoning, wherein the number of reasoning steps equals the depth of the transformer (layers minus one), a significantly smaller $d_m$ may suffice.

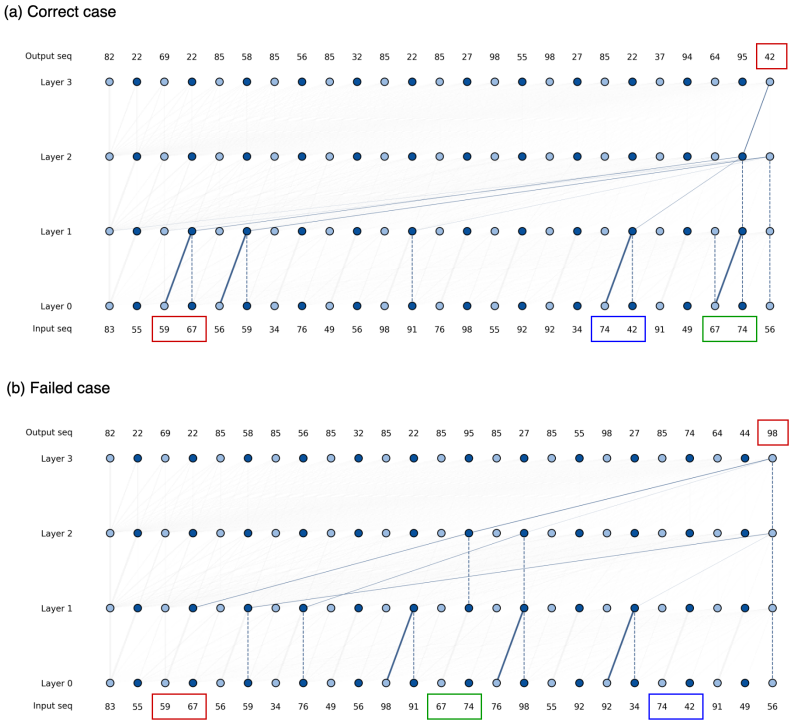



Figure 11: (a) Attention propagation under the valid ordering predicted by our theory. When (67, 74) appears after both (74, 42) and (59, 67), the model is able to consolidate information and the critical attention route follows exactly the theoretically derived path.(b) Attention breakdown under an invalid ordering. If the relative order of the three arrays is altered such that (67, 74) is no longer placed after the other two, the model fails to aggregate sufficient information, resulting in fragmented and chaotic attention patterns

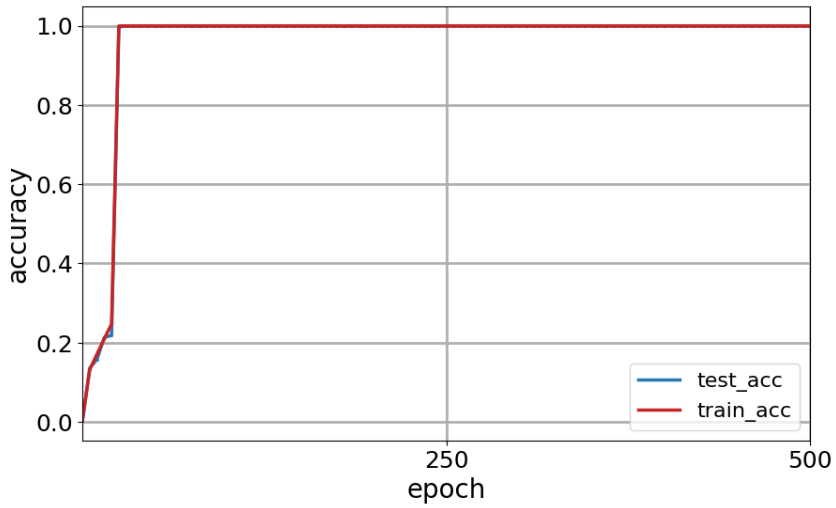



Figure 12: L=4 $d_m = 128$

We train a 4-layer Transformer model to perform 3-step reasoning. In this experiment, the sequence length is set to 21, the batch size is 1000, and the learning rate is $5 \times 10^{-5}$. The model is trained for 500 epochs with a hidden dimension of $d_m = 128$.

The experimental results indicate that the blue and red strings both rapidly approach 100% success rates. Under the string reasoning condition, a transformer model with 128 hidden dimensions demonstrates the capability to effectively perform 3-step reasoning tasks (figure 12). In contrast, under the parallel reasoning condition, an architecturally equivalent model with the same number of hidden dimensions achieves only an 11.9% success rate.

### E.9 Computational Resources

The experiments were conducted on a server with the following configuration:

- 64 AMD EPYC 7742 64-Core Processor.
- 256GB of total system memory.
- 4 A100 GPUs with 40GB of memory.

The task can be completed within 3 weeks with 2 NVIDIA A100 GPU.

# F Additional Examples and Matching-Order Discussion

## F.1 Permuted reasoning sequences and reasoning starts

The following remark and examples supplement the definitions in Section 4.

**Remark F.1.** *In the definition of reasoning sequence we use the permutation to change the order of reasoning pairs which does not break the relation inside each reasoning pair. Moreover, no permutation should be applied to the original sequence* $(x_i)$*. For example, the sequence* $(1, 2, 2, 3, 3, 4)$ *can be* $(2, 3, 3, 4, 1, 2)$ *or* $(3, 4, 2, 3, 1, 2)$ *under some certain permutations. Both of these sequences are related to the reasoning chain* $((1, 2), (2, 3), (3, 4))$*. However, it cannot be transformed into* $(1, 3, 3, 4, 2, 3)$ *through any permutation that acts on reasoning pairs.*

**Example F.2.** *The sequence* $(x_i)_{i\geqslant 1} = (\lfloor \frac{i}{2} \rfloor)_{i\geqslant 1}$ *can be seen as a sorted reasoning sequence with constructing permutation* $\sigma = \mathrm{Id}$ *and constructing reasoning chain* $((0, 1), (1, 2), (2, 3), (3, 4), \cdots)$.

**Example F.3.** *The sequence* $(x_i) = (1, 2, 6, 3, 2, 4, 3, 5, 4, 6)$ *has constructing reasoning chain*

$$(\boldsymbol{a}_m) = ((1, 2), (2, 4), (4, 6), (6, 3), (3, 5))$$

*and constructing permutation* $\sigma$ *satisfying*

$$\sigma(1) = 1, \quad \sigma(2) = 4, \quad \sigma(3) = 2,$$
$$\sigma(4) = 5, \quad \sigma(5) = 3.$$

*In this example, In this example,*

$$\boldsymbol{a}_1 = (1, 2) = (x_1, x_2),$$
$$\boldsymbol{a}_2 = (2, 4) = (x_5, x_6),$$
$$\boldsymbol{a}_3 = (4, 6) = (x_9, x_{10}).$$

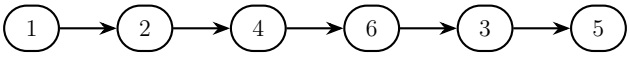



Figure 13: Reasoning task represents by $(x_i)$ in example F.3.

**Example F.4.** *We set the sequence* $(x_i) = (1, 2, 6, 3, 2, 4, 3, 5, 4, 6, 4)$ *with constructing reasoning chain* $(\boldsymbol{a}_m)$ *and constructing permutation* $\sigma$ *as in example F.3. In addition, the reasoning start is set to be* $4$*. Then one step reasoning result is* $6$ *and two steps reasoning result is* $3$*.*

## F.2 Alternative matching order

The following discussion refers to the information propagation rules in Section 5.

**Remark F.5.** *In the above information propagation rules we require that the adjacent position matching only happens when* $l = 1$ *and the same token matching only happens when* $l \geqslant 2$*. We can also set the same token matching to happen when* $l = 1$ *and adjacent position matching to happen when* $l \geqslant 2$*, since the index set* $I_i^l$ *in fact encodes the position information. The necessary condition for adjacent position matching to happen is* $\exists I_k^l$ *and* $I_j^l$ *s.t. there exist* $i_j \in I_j^l$ *and* $i_k \in I_k^l$ *satisfying* $\min i_k, i_j \bmod 2 = 1$ *and* $|i_j - i_k| = 1$*. It is easy to see that the index set* $I_i^l$ *in this same token matching first rules plays the same role as* $V_i^l$ *in above adjacent position matching first rules, and there will be no essential difference for the result in our main theorems under these two different rules. For simplicity, we only consider the above adjacent position matching first rules.*

If value matches occurs in the first layer, then in subsequent layers, only position matches can lead to an increase in information. This is because the only way a value match can occur in later layers is if at least one of the nodes involved comes from the residual. This however, would not lead to the increase of information quantity.

A simple example: consider the task $a \to b \to c \to d \to e \to f$, and the corresponding reasoning sequence $(a, b, e, f, b, c, d, e, c, d)$, with positions denoted as $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$. If a value

match occurs in the first layer, the nodes in the second layer take the following form:

$$\{\{a\},\{1\}\},\{\{b\},\{2,5\}\},\{\{c\},\{6,9\}\},$$
$$\{\{d\},\{7,10\}\},\{\{e\},\{3,8\}\},\{\{f\},\{4\}\},$$

along with some residual nodes. Clearly, no value match can happen between these nodes. Moreover, value matches involving residual nodes do not result in an increase in the amount of information.

Similarly, if a position match occurs in the first layer, then in subsequent layers, only value matches can lead to an increase in the amount of information. Therefore, only two cases are important: position match first rules and value match first rules. Mixed scenarios need not be considered. Additionally, based on the explanation in Remark F.5, we will only consider the position match first rules adopted in the subsequent discussion.