TRANSFORMERS PROVABLY LEARN TO INTERNALIZE CHAIN-OF-THOUGHT

Yixiao Huang[1,*] **Hanlin Zhu**[1], **Zixuan Wang**[2],
Jiantao Jiao[1], **Stuart Russell**[1], **Somayeh Sojoudi**[1], **Song Mei**[1]
[1] UC Berkeley [2] Princeton University

ABSTRACT

Chain-of-Thought (CoT) prompting can substantially improve the sample efficiency of Large Language Models (LLMs), reducing the complexity of tasks like parity learning from exponential to polynomial. However, this benefit comes at the cost of generating explicit reasoning steps, which is computationally expensive during inference. Implicit CoT (ICoT) has been proposed to mitigate this cost by training models to internalize these intermediate steps. In this work, we provide a theoretical analysis of a multi-layer transformer showing that ICoT retains the sample efficiency gains of explicit CoT, enabling models to solve complex tasks efficiently without sacrificing inference speed. Moreover, we propose Log-ICoT, a provably efficient curriculum that reduces training stages from linear dependency on the problem complexity to a logarithmic one. Our theoretical results are verified with numerical experiments, confirming that ICoT offers a path to robust reasoning without the high computational overhead.

1 INTRODUCTIONS

Chain-of-thought (CoT) reasoning (Wei et al., 2022) has become a cornerstone for enabling Large Language Models (LLMs) to solve challenging tasks. By generating explicit intermediate tokens, CoT decomposes complex problems into manageable sub-steps, significantly boosting performance. However, this reasoning power comes at a high cost: the explicit generation of thinking tokens substantially increases inference latency and computational overheads. To address, Implicit CoT (ICoT) (Deng et al., 2024) has emerged as a promising paradigm. ICoT trains models to internalize reasoning steps within their hidden states, removing the need for explicit token generation at inference time. While empirical results are encouraging, the underlying mechanics of how models internalize these steps remain poorly understood. Furthermore, the standard ICoT curriculum, which removes intermediate steps one by one, scales linearly with the length of the reasoning chain, leading to inefficient training.

In this work, we provide a formal analysis of how transformers internalize reasoning steps. We consider the parity learning task, a classic problem requiring exponential samples to learn without CoT (Shalev-Shwartz et al., 2017; Wen et al., 2024). We demonstrate that transformers can internalize CoT with only a polynomial number of samples. This matches the sample complexity of explicit CoT (Kim & Suzuki, 2024; Wen et al., 2024) while significantly reducing inference costs. Building on these insights, we introduce Log-ICoT, an efficient curriculum that further reduces the required training stages from linear to logarithmic. Our contributions are as follows:

- In Theorem 1, we prove that multi-layer transformers can learn to internalize parity computations using a logarithmic number of gradient updates under our Log-ICoT framework.
- We verify our theoretical findings through numerical experiments on a multi-layer transformer in Section 4.

[*]Corresponding author. Email: yixiaoh@berkeley.edu

1.1 Related Work

Chain-of-thought and its variants. Chain-of-thought (CoT) (Wei et al., 2022) can enhance LLM's reasoning capability by letting models output the intermediate thought tokens. It can be encouraged only in the prompt (Khot et al., 2022; Zhou et al., 2022) or be included in the training set (Yue et al., 2023; Yu et al., 2023; Wang et al., 2023a; Shao et al., 2024). A line of theoretical works also studies the advantages of the CoT method via expressivity (Liu et al., 2022; Feng et al., 2023; Merrill & Sabharwal, 2023; Li et al., 2024b) or training dynamics (Zhu et al., 2024b; Wen et al., 2024; Kim & Suzuki, 2024). A more recent line of work studies variants of CoT to better improve the model performance or reduce the inference cost, including ICoT (Deng et al., 2024), pause tokens (Goyal et al., 2023), filler tokens (Pfau et al., 2024), planning tokens (Wang et al., 2023b), token assorted (Su et al., 2025), chain of continuous thought (Hao et al., 2024), etc. Compared to CoT, its variants are much less explored, especially theoretically. London & Kanade (2025) theoretically shows that the pause token can increase expressivity, and Zhu et al. (2025a;b); Gozeten et al. (2025) theoretically shows the advantage of continuous CoT. Our paper contributes to the field by providing the first theoretical results demonstrating the advantage of the ICoT method, i.e., it can bridge the gap between expressivity and learnability.

Training dynamics of transformers. There is a very rich line of literature studying the optimization of transformer-based models (Jelassi et al., 2022; Bietti et al., 2023; Mahankali et al., 2023; Fu et al., 2023; Zhang et al., 2024; Li et al., 2024a; Huang et al., 2024). Many works focus on how certain attention patterns formed during training (Tian et al., 2023a;b; Guo et al., 2024). Another line of work focuses on various reasoning abilities or patterns of transformers through the lens of training dynamics (Boix-Adsera et al., 2023; Nichani et al., 2024; Wang et al., 2024; Ren et al., 2024; Zhu et al., 2024b; Guo et al., 2025; Huang et al., 2025a; Chen et al., 2024; Huang et al., 2025b; Ma et al., 2026). The most related work to our setting is Wen et al. (2024); Kim & Suzuki (2024), which shows that CoT enables much better sample efficiency when learning parity functions with secret indices. Our work follows the setting of Kim & Suzuki (2024) and takes a further step to study the sample complexity of ICoT via training dynamics. Our work shows that ICoT enjoys both sample efficiency and low inference-time cost. Moreover, we analyze a multi-layer transformer, which is more practical than the one-layer transformer considered in Wen et al. (2024); Kim & Suzuki (2024).

2 Preliminaries

Notations. For any integer $N > 0$, we use $[N]$ to denote the set $\{1, 2, \ldots, N\}$. We use lower-case and upper-case bold letters (e.g., $\boldsymbol{a}, \boldsymbol{A}$) to represent vectors and matrices. Let $\boldsymbol{e}_{d,i} \in \mathbb{R}^d$ denote the d-dimensional one-hot vector with a 1 in the i-th coordinate, $\boldsymbol{I}_K \in \mathbb{R}^{K\times K}$ denote the identity matrix. We generalize the standard linear product by defining multi-linear inner product of vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r \in \mathbb{R}^d$ for any $r \in \mathbb{N}$ as $\langle \boldsymbol{x}_1, \ldots, \boldsymbol{x}_r \rangle := \sum_{i=1}^d \prod_{j=1}^r x_{j,i}$. In particular, $\langle \boldsymbol{x}_1 \rangle = \boldsymbol{x}_1^\top \mathbf{1}_d$ and $\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle = \boldsymbol{x}_1^\top \boldsymbol{x}_2$.

2.1 Task settings

Assume $n \geq k \geq 2$ and denote $\mathcal{S}$ as the set of all subsets $[n]$ with size k. Following the k-parity learning problem in Kim & Suzuki (2024); Wen et al. (2024), we consdier a secret index set S drawn uniformly from $\mathcal{S}$ and an input vector $\boldsymbol{b} = (b_j)_{j=1}^n \sim \mathsf{Unif}(\{\pm 1\}^n)$. The label y is determined by the parity function $y = p_S(\boldsymbol{b}) := \prod_{j\in S} b_j$, where n is the input length and $k = |S|$ determines the hardness of the learning problem. We consider learning the parity in a finite-sample setting where we are given a dataset of size B consisting of i.i.d. samples $\{(\boldsymbol{b}^i, y^i)\}_{i=1}^B$ with $\boldsymbol{b}^i \sim \mathsf{Unif}(\{\pm 1\}^n)$ and $y^i = p_S(\boldsymbol{b}^i)$.

Let $f_\theta : \{\pm 1\}^n \to \mathbb{R}$ be any differentiable parametrized model. We consider the empirical squared loss

$$\mathcal{L}_B(\theta) \;\coloneqq\; \frac{1}{2B} \sum_{i=1}^B \big(y^i - f_\theta(\boldsymbol{b}^i)\big)^2.$$

An *ε-approximate gradient oracle* for $\mathcal{L}_B$ is a (possibly randomized) map $\tilde{\nabla}\mathcal{L}_B(\theta)$ such that $\|\tilde{\nabla}\mathcal{L}_B(\theta) - \nabla\mathcal{L}_B(\theta)\|_2 \leq \varepsilon$ for all queried θ. We evaluate performance by the population L_2

error

$$\mathcal{E}(\theta) := \mathbb{E}_{S\sim\mathsf{Unif}(\mathcal{S})}\,\mathbb{E}_{\boldsymbol{b}\sim\mathsf{Unif}(\{\pm1\}^n)}\big[(p_S(\boldsymbol{b}) - f_\theta(\boldsymbol{b}))^2\big].$$

Proposition 1 (adapted from Theorem 2 in Kim & Suzuki (2024)). *Assume $k = \Theta(n)$. Suppose the sample size satisfies $B = \Omega(n^\nu)$ and the gradient is bounded by $\|\nabla f_\theta\| = O(n^{\nu_1})$. Then, there exists an $O(n^{-\nu_2})$-accurate approximate gradient oracle $\tilde{\nabla}$ such that, with probability at least $1 - \exp(-\Omega(n))$ over the random sampling of the dataset, the output $\theta(\mathcal{A})$ of any (possbly randomized) iterative algorithm $\mathcal{A}$ that makes at most $O(n^{\nu_3})$ queries to the finite-sample gradient $\tilde{\nabla}\mathcal{L}_B$ satisies*

$$\mathcal{E}(\theta(\mathcal{A})) \geq 1 - O(n^{-\nu_4}),$$

where $\nu = 4\nu_1 + 4\nu_2 + 2\nu_3 + 2\nu_4 + 1$.

2.2 Model

Transformer architectures. Similar to Wang et al. (2025), we consider an L-layer simplified transformer. The input to the model is a sequence of length T, represented by a matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T] \in \mathbb{R}^{d\times T}$, where each $\boldsymbol{x}_j \in \mathbb{R}^d$ is a d-dimensional embedding.

Embedding layer. The embedding layer in a standard transformer maps every token in the vocabulary to a vector with dimension d and adds a positional embedding. Following Kim & Suzuki (2024), we stack the input sequence such that the j-th element of the batch is represented by a vector $\boldsymbol{b}_j \in \mathbb{R}^B$ for each position $j \in [T]$. Here, $\boldsymbol{b}_j = \left[b_j^1, b_j^2, \ldots, b_j^B\right]^\top$ aggregates the j-th bit across all B samples in the batch. This layout allows the entire batch to use a shared positional encoding as specified below. Let $\boldsymbol{D} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_T] \in \mathbb{R}^{B\times T}$ be the training data. We define the input sequence length $T := n + k - 1$ and embedding dimension $d := T + B(L+1)$. For every $j \in [T]$, the input bits $\boldsymbol{b}_j$ is embedded into a higher-dimensional space as:

$$\boldsymbol{x}_j = \boldsymbol{E}(\boldsymbol{D})_j = \begin{bmatrix} \boldsymbol{p}_j \\ \boldsymbol{b}_j \\ \boldsymbol{0}_{B\times L} \end{bmatrix} \in \mathbb{R}^d,$$

where $\boldsymbol{p}_j := \boldsymbol{e}_{T,j} \in \mathbb{R}^T$ is a one-hot positional encoding and the final BL dimensions are reserved for the residual stream (Elhage et al., 2021).

Attention Layer. While standard multi-layer transformers typically adopt multiple self-attention heads, we consider transformers with a single head per layer to simplify the analysis. We also reparameterize the key-query matrix by $\boldsymbol{W}_{\mathsf{KQ}} = \boldsymbol{W}_{\mathsf{K}}\boldsymbol{W}_{\mathsf{Q}}^\top \in \mathbb{R}^{d\times d}$ in line with Tian et al. (2023a); Zhu et al. (2024b); Li et al. (2024a). Moreover, we only optimize the lower $T\times T$ block, $\boldsymbol{W} \in \mathbb{R}^{T\times T}$, such that the attention scores are determined by the positional encodings only. In contrast, the value matrix is simply set to the identity matrix, which is omitted in the following analysis.

$$B_L := B(L+1), \boldsymbol{W}_{\mathsf{KQ}} := \boldsymbol{W}_{\mathsf{K}}\boldsymbol{W}_{\mathsf{Q}}^\top = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{0}_{T\times B_L} \\ \boldsymbol{0}_{B_L\times T} & \boldsymbol{0}_{B_L\times B_L} \end{bmatrix}, \boldsymbol{W}_{\mathsf{V}} = \boldsymbol{I}_d.$$

Definition 1 (Causal Self-Attention). *Given an attention matrix $\boldsymbol{W} \in \mathbb{R}^{T\times T}$, the causal self-attention module $\mathsf{Attn}(\cdot, \boldsymbol{W}) : \mathbb{R}^{d\times T} \to \mathbb{R}^{d\times T}$ is given by:*

$$\mathsf{Attn}(\boldsymbol{X}) := \mathsf{Attn}(\boldsymbol{X}; \boldsymbol{W}) = \boldsymbol{X}\mathbb{S}(\boldsymbol{X}^\top\boldsymbol{W}_{\mathsf{KQ}}\boldsymbol{X} + \boldsymbol{C}).$$

where $\mathbb{S}(\cdot)$ is the softmax operation applied column-wise and $\boldsymbol{C} \in \mathbb{R}^{T\times T}$ is the modified causal attention mask to be defined in Section 3.2.

MLP Layer. The MLP layer contains a linear matrix $\boldsymbol{W}_{\mathsf{O}}$ with a link function ϕ as defined below.

Definition 2 (Link function). *Following Kim & Suzuki (2024), we consider a fixed, symmetric link function $\phi : [-1, 1] \to [-1, 1]$ applied pointwise. We choose ϕ such that it maps sums to parities for bipolar inputs. Specifically, we require $\phi(0) = -1, \phi(\pm1) = 1$, which emsures that $\phi(\frac{a+b}{2}) = ab$ for $a, b \in \{-1, 1\}$. Moreover, we assume ϕ is sufficiently smooth and satisfies $\phi'(0) = \phi'(\pm1) = 0$. This allows for a local Taylor expansion around 0 of the form: $\phi(t) = -1 + ct^2 + O(|t|^4)$ and $\phi'(t) = 2ct + O(|t|^3)$, for some constant $c > 0$.*

Hyper-connections. We use hyper-connections (Zhu et al., 2024a) as a structured alternative to standard residual connections. As observed by Zhu et al. (2024a), conventional residual updates can suffer a trade-off between representation collapse and gradient vainishing. Hyper-connections mitigate this by allowing the model to modulate how strongly information is propagated across layers. Unlike their original formulation that uses globally shared connection weights, we use a *position-wise* variant that mixes the update path and the skip path separately at each token.

Concretely, each layer ℓ is equipped with a gate vector $\boldsymbol{g}^{(\ell)} \in \mathbb{R}^T$ that controls how the layer mixes the newly computed MLP update with the existing representation at each token position. We formalize this mechanism via the following hyper-connection operator:

$$\mathsf{HC}(\boldsymbol{A}, \boldsymbol{B}; \boldsymbol{g}) := \boldsymbol{A}\,\mathrm{diag}(\boldsymbol{g}) + \boldsymbol{B}\,(\boldsymbol{I}_T - \mathrm{diag}(\boldsymbol{g})).$$

where $\boldsymbol{g} \in \mathbb{R}^T$, $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d\times T}$.

Definition 3 (Multi-layer Transformer)**.** *Let L be the depth, in each layer $\ell \in [L]$ the transformer combines a self-attention layer with an MLP with a linear layer $\boldsymbol{W}_{\mathsf{O}} \in \mathbb{R}^{d\times d}$ and fixed link function ϕ. Let $\boldsymbol{\theta} := \{\boldsymbol{W}^{(\ell)}, \boldsymbol{W}_{\mathsf{O}}^{(\ell)}, \boldsymbol{g}^{(\ell)}\}_{\ell\in[L]}$, we initialize $\boldsymbol{X}^{(0)} = \boldsymbol{E}(\boldsymbol{D})$. The layer update with hyper-connections is*

$$\boldsymbol{X}^{(\ell)} = \boldsymbol{X}^{(\ell-1)} + \boldsymbol{W}_{\mathsf{O}}^{(\ell)}\mathsf{HC}\left(\phi(\mathsf{Attn}(\boldsymbol{X}^{(\ell-1)}; \boldsymbol{W}^{(\ell)})), \boldsymbol{X}^{(\ell-1)}; \boldsymbol{g}^{(\ell)}\right), \ell \in [L].$$

The network output is $\mathcal{T}_{\boldsymbol{\theta}}(\boldsymbol{D}) = \boldsymbol{X}^{(L)}$. For $m \in [T]$, the output can be written as:

$$\mathcal{T}_{\boldsymbol{\theta}}(\boldsymbol{D})_m = \boldsymbol{x}_m + \sum_{\ell=1}^{L} \boldsymbol{W}_{\mathsf{O}}^{(\ell)} \left(g_m^{(\ell)} \phi(\sum_{j=1}^{m} \sigma_j(\boldsymbol{w}_m^{(\ell)}) \boldsymbol{x}_j^{(\ell-1)}) + (1 - g_m^{(\ell)}) \boldsymbol{x}_m^{(\ell-1)} \right),$$

where $\boldsymbol{w}_m^{(\ell)} = [w_{j,m}^{(\ell)}]_{j\in[T]} = [\boldsymbol{p}_j^\top \boldsymbol{W}^{(\ell)} \boldsymbol{p}_m]_{j\in[T]}$ and $\sigma_j(\boldsymbol{w}_m^{(\ell)}) = e^{w_{j,m}^{(\ell)}} / \sum_{\alpha=1}^{m} e^{w_{\alpha,m}^{(\ell)}}$.

Finally, we use a readout layer $\Psi \in \mathbb{R}^{d\times B}$ to decode the output $\hat{f}(\boldsymbol{D}) = \Psi^\top \mathcal{T}_{\boldsymbol{\theta}}(\boldsymbol{D})$.

We fix the output matrix $\boldsymbol{W}_{\mathsf{O}}^{(\ell)} \in \mathbb{R}^{d\times d}$ to faciliate the transfer of information between residual blocks:

$$\boldsymbol{W}_{\mathsf{O}}^{(\ell)} = \begin{bmatrix} \boldsymbol{0}_{T\times T} & \boldsymbol{0}_{T\times B_L} \\ \boldsymbol{0}_{B_L\times T} & \boldsymbol{e}_{L+1,\ell+1}\boldsymbol{e}_{L+1,\ell}^\top \otimes \boldsymbol{I}_B \end{bmatrix},$$

which moves information from block in ℓ to $\ell+1$ that acts like a sequential prediction. We then initialize the readout layer $\Psi_\ell \in \mathbb{R}^{d\times B}$ as

$$\Psi_\ell(0) = \begin{bmatrix} \boldsymbol{0}_{T\times B} \\ \beta_0 \boldsymbol{e}_{L+1,\ell+1} \otimes \boldsymbol{I}_B \end{bmatrix}.$$

Given this construction, the readout layer $\Psi_{\ell'}$ at stage ℓ' acts as a selector for the $(\ell'+1)$-th block. Consequently, the product $\Psi_{\ell'}^\top \boldsymbol{W}_{\mathsf{O}}^{(\ell)}$ is non-zero if and only if $\ell = \ell'$. Thus the final relevant output for stage ℓ is the ℓ-th layer $\boldsymbol{W}_{\mathsf{O}}^{(\ell)} \phi(\sum_{j=1}^{m} \sigma_j(\boldsymbol{w}_m^{(\ell)} \boldsymbol{x}_j^{(\ell)}) \boldsymbol{x}_j^{(\ell)})$ as the gradient for the ℓ-layer $\boldsymbol{W}^{(\ell)}$ is non-zero only in stage $t \geq \ell$. Moreover, Lemma 6 shows that the gradient for $\boldsymbol{W}^{(\ell)}$ is also close to zero after being well trained.

2.3 Implicit Chain-of-Thought

Parity is fundamentally difficult for finite-precision gradient-based methods to solve within polynomial steps (Shalev-Shwartz et al., 2017; Wies et al., 2022). Conversely, Kim & Suzuki (2024); Wen et al. (2024) have demonstrated that one-layer transformers can solve parity efficiently when provided with intermediate supervision via CoT reasoning (Kojima et al., 2022; Wei et al., 2022).

However, despite its favorable sample complexity, CoT requires the explicit generation of intermediate reasoning traces, which substantially increases inference latency and computational cost. To address this, implicit chain-of-thought (ICoT) (Deng et al., 2024) is proposed to internalize reasoning in hidden states by progressively removing intermediate steps during fine-tuning. This approach aims to match the high performance of explicit CoT while matching the inference speed of direct answer generation. In this paper, we prove that multi-layer transformers can also learn to solve parity

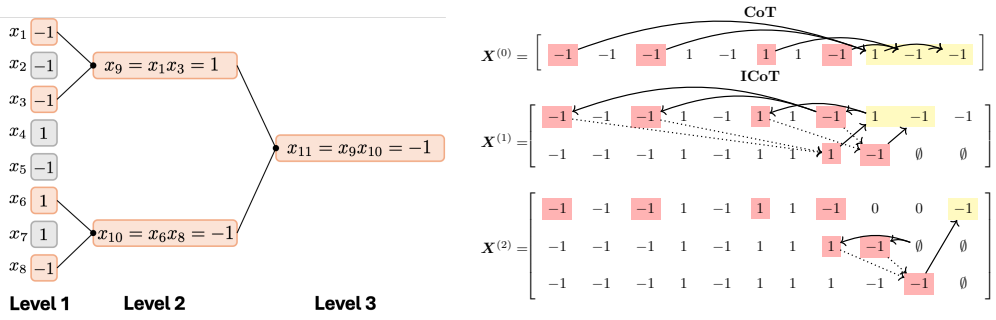


Figure 1: **Illustration of parity learning task with input length $n = 8$ and secret set size $k = 4$.** *Left:* The task can be decomposed into a hierarchical two-parity computation. *Right:* Comparison of the training curriculum of Chain-of-Thought (CoT) and Implicit CoT (ICoT). Both methods initially leverage complete thinking traces derived from the hierarchical decomposition. As the ICoT curriculum progresses, these intermediate reasoning steps are replaced by padding tokens (0), forcing the model to internalize the computation within its hidden states.

Algorithm 1 Training Algorithm (Log-ICoT Curriculum)

Input: learning rate η
Initialize $\boldsymbol{W}^{(\ell)}(0) = 0_{T\times T}$ for $\ell \in [L]$, $\boldsymbol{\theta} := \{\boldsymbol{W}^{(\ell)}\}_{\ell\in[L]}$
for $t = 1, \cdots, L$ **do**
$\quad \boldsymbol{\theta}(t) \leftarrow q\left(\boldsymbol{\theta}(t-1) - \eta\tilde{\nabla}_{\boldsymbol{\theta}}\mathcal{L}^{(t)}(\boldsymbol{\theta}(t-1))\right)$
end for
Output: $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}(L)$.

efficiently with a revised ICoT curriculum called Log-ICoT. This provides a theoretical guarantee that models can solve complex tasks efficiently without sacrificing inference speed.

We first introduce a task decomposition scheme used in Wies et al. (2022); Kim & Suzuki (2024) for parity and illustrate the difference between CoT and ICoT scheme for this task. We assume $k = 2^v$ for some integer v for simplicity and decompose the problem into a hierarchy of two-parity computations which can be efficiently learned in a sequential manner by the multi-layer transformers.

The decomposition is illustrated in Figure 1, where the task is represented as a complete binary tree of height v containig $2k - 1$ total nodes. The leaf nodes (Level 1) represent the input bits x_j for $j \in [n]$. The internal nodes are indexed sequentially as $x_{n+1}, \ldots, x_{n+k-1}$, moving from the bottom level upward and left to right. We denote the maximum index at each level $\ell \in [1, v+1]$ as $n_1 = n, n_\ell = n + \sum_{j=1}^{\ell-1} 2^{v-j} = n + k(1 - 2^{-(\ell-1)})$. Moreover, for any internal node x_m $(m > n)$, let $c_1[m], c_2[m]$ be its two child indcies where $c_1[m] < c_2[m] < m$ and let $p[m]$ denote its parent index. Finally, the level of a node x_m is given by $h[m]$ satisfying $n_{h[m]-1} < m \leq n_{h[m]}$.

3 Theoretical Results

3.1 Training Scheme

We adopt the following curriculum termed Log-ICoT similar to Deng et al. (2024):

- Stage 1: Full CoT
- Stage t $(2 \leq t < L)$: Replace the first $k(1 - 2^{-(t-1)})$ CoT steps with padding token 0.
- Stage L: All CoT steps except the final output are removed.

Since $L = \log_2 k$, the curriculum contains only a logarithmic number of steps. This curriculum is also reflected in the right side of Figure 1.

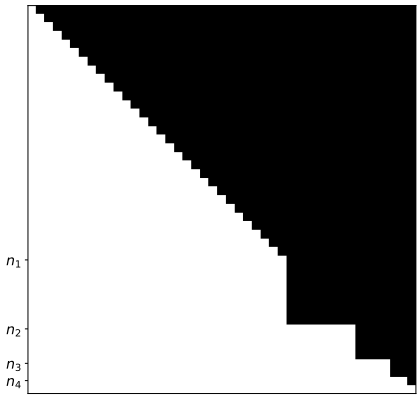


Figure 2: **Illustration of the customized attention mask.** Each intermediate state $\boldsymbol{x}_m$ at CoT positions ($m \geq n$) only depends on tokens $\boldsymbol{x}_j$ up to the previous level, i.e., $j \leq h[m]$.

Loss function. Following Kim & Suzuki (2024), we use least square loss $\ell(\hat{y}, y) = \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|^2$ as the loss function. For ICoT at stage t, the loss function is given by:

$$\mathcal{L}^{(t)}(\boldsymbol{\theta}) = \frac{1}{2B} \sum_{m=n_t}^{n_{t+1}-1} \|\hat{f}(\boldsymbol{D})_m - \boldsymbol{b}_{m+1}\|^2 = \frac{1}{2B} \sum_{m=n_t}^{n_{t+1}-1} \|\Psi_t^\top \mathcal{T}_{\boldsymbol{\theta}}(\boldsymbol{D})_m - \boldsymbol{b}_{m+1}\|^2 \tag{1}$$

Training and Evaluation. We consider single-pass SGD with batch size B. For the parameter $\boldsymbol{\theta}$, we write $\boldsymbol{\theta}_{\setminus \boldsymbol{W}}$ to denote all parameters except the attention matrix $\boldsymbol{W}$. The training algorithm can be found in Algorithm 1. At test time, we randomly generate test inputs $\boldsymbol{D}_{\mathsf{test}} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_T] \in \mathbb{R}^{B' \times T}$ and corresponding label $(\boldsymbol{y}_{\mathsf{test}})_i = \prod_{j \in \mathcal{S}} b_j^i$ for all $i \in [B']$.

3.2 Key Challenges

There are two primary challenges in training multi-layer transformers. We discuss them in turn.

Representation Collapse. First, the input states collapse in later layers and lead to exponentially large noise that confuses the gradient. This is formalized in the following lemma:

Lemma 1 (Representation collapse of input states)**.** *For $B = \mathsf{poly}(n)$, with probability $1 - \exp(-n^{\epsilon/16})$ over random sampling of the input data $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_T$, it holds that, for all $n^{\epsilon/8} \leq m \leq T$:*

$$\|\phi(\frac{1}{m} \sum_{\alpha \in [m]} \boldsymbol{b}_\alpha) + \mathbf{1}_B\|_\infty \leq O(n^{-\epsilon/16}).$$

In particular, when $\boldsymbol{W}$ is initialized to $\mathbf{0}_T$, for all $n^{\epsilon/8} \leq m \leq T$, we have

$$\|\phi(\sum_{j=1}^{m} \sigma_j(\boldsymbol{w}_m)\boldsymbol{b}_j) + \mathbf{1}_B\|_\infty = \|\phi(\frac{1}{m} \sum_{j \in [m]} \boldsymbol{b}_j) + \mathbf{1}_B\|_\infty \leq O(n^{-\epsilon/16})$$

The proof is given in Section A.1, which simply adopts a concentration bound argument as the random input data has mean 0 and is bounded. To resolve this problem, we incorporate hyper-connections to replace the residual connections, inspired by Zhu et al. (2024a) which was proposed to mitigate representation collapse in pre-training.

In the original setup, the hyper-connections can either be static or learnable weights. We consider a design that adaptively adjust the gating by the representation. Specifically, for positions in the CoT steps ($m \geq n$), we set the connection weight to 0 if all data in the batch collapse to -1 by adding the following filter after the feedforward layer ϕ such that the representation in the previous layer is

carried to the next layer.

$$g^{(\ell)}(\boldsymbol{X}^{(\ell-1)})_m = \begin{cases} 0 & \|\Psi_{\ell-1}^\top \hat{\boldsymbol{z}}_m^{(\ell)} + \mathbf{1}_B\|_\infty < \varepsilon_0 \\ 1 & \text{o.w.} \end{cases}, \forall\, m \geq n$$

where $\hat{\boldsymbol{z}}_m^{(\ell)} = \sum_{j=1}^m \sigma_j(\boldsymbol{w}_m^{(\ell)})\boldsymbol{x}_j^{(\ell-1)}$. For input positions $m \leq n-1$, we simply set it to $g^{(\ell)}(\boldsymbol{X}^{(\ell-1)})_m = 0$. This effectively prevents representation collapse across layers. As we will show later, by Lemma 1, once we set $\varepsilon_0 > O(n^{-\epsilon/16})$, we have $g_m^{(\ell)} = \begin{cases} 0 & m \leq n_\ell - 1 \\ 1 & \text{o.w.} \end{cases}$ if the attention weights yield uniform attention after one-step gradient update.

Error propagation. As we adopt a multi-stage training curriculum, incorrect predictions from last stage can propagate to later layers, which exponentially amplifies the errors and drowns out the gradient signals. To address this, we introduce two modifications to the vanilla transformer model, following Kim & Suzuki (2024).

We revise the causal attention mask such that each intermediate state $\boldsymbol{x}_m$ only depends on tokens $\boldsymbol{x}_j$ up to the previous level. An illustration of the mask can be found in Figure 2, with the formulation specified below:

$$[\boldsymbol{C}]_{j,m} = \begin{cases} -\infty, & m \leq n \text{ and } j > m \\ -\infty, & m > n \text{ and } j > n_{h[m]-1} \\ 0, & \text{o.w.} \end{cases}$$

After every gradient update, we quantize the attention weights by rounding each entry of $\{\boldsymbol{W}^{(\ell)}\}_{\ell\in[L]}$ to the nearest integer: $\boldsymbol{W}^{(\ell)}(t+1) = q\left(\boldsymbol{W}^{(\ell)}(t) - \eta\nabla_{\boldsymbol{W}^{(\ell)}}\mathcal{L}^{(t)}(\boldsymbol{\theta}(t))\right)$ where $q : \mathbb{R} \to \mathbb{Z}$ is the nearest-integer operator. Integer-based quantization methods are widely used in practice to improve training and inference efficiency (Jacob et al., 2018; Wu et al., 2020), and has also been used in theory to control error propagation (Kim & Suzuki, 2024). In our setting, quantization additionally "locks" previously trained layers $\{\boldsymbol{W}^{(\ell)}\}_{\ell\in[t-1]}$. Once their weights are trained, small subsequent gradients do not change them after rounding, allowing us to focus on analyzing $\boldsymbol{W}^{(t)}$ at stage t.

3.3 Main Theorem

The two modifications allow us to prevent uncontrolled error accumulation and make the stagewise optimization amenable to analysis. We are therefore ready to state our main theoretical result.

Theorem 1 (Log-ICoT). *Consider an $L = \log_2 k$-layer transformer with customized attention mask in Figure 2. Suppose $B = \Omega(n^{2+\epsilon})$ for some constant $\epsilon > 0$ where the input sequence length n is sufficiently large. Let $\tilde{\nabla}$ be any $O(n^{-2-\epsilon/8})$-approximate gradient oracle. Following Algorithm 1 with learning rate $\eta = \Theta(n^{2+\epsilon/16})$. Then it holds with probability $1 - \exp(-T^{\epsilon/2})$ over random sampling of dataset $\mathcal{D}$ that after L-step update w.r.t Eq. (1), we have*

$$\|\hat{f}(\boldsymbol{D}_{\text{test}}) - \boldsymbol{y}_{\text{test}}\|_\infty \leq \exp(-\Omega(n^{\epsilon/16}))$$

The proof is given in Section A.2.

4 Experiments

We train a 4-layer transformer with $n = 30, k = 16$ and report the training loss in Figure 3. In Figure 4, we show the attention patterns of the trained model. As explicit CoT steps are removed, it forces the model to internalize all the reasoning tokens inside the model's hidden states, which verifies the result in Theorem 1.

5 Conclusion

In this paper, we provide a rigorous theoretical foundation for Implicit Chain-of-Thought (ICoT), demonstrating that multi-layer transformers can internalize complex reasoning processes within

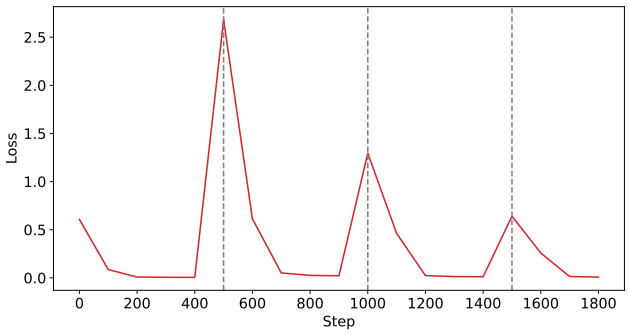
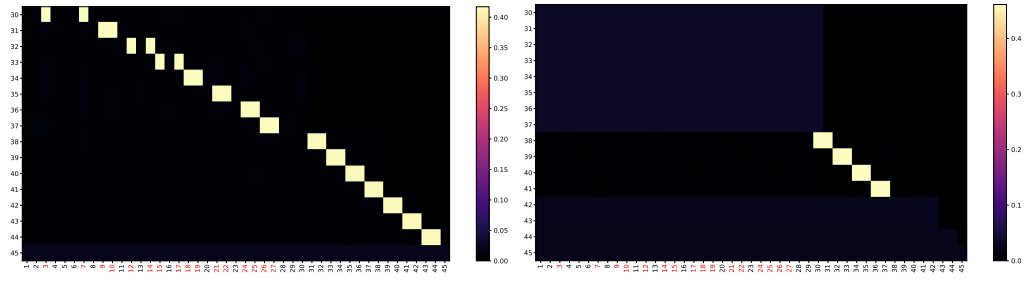
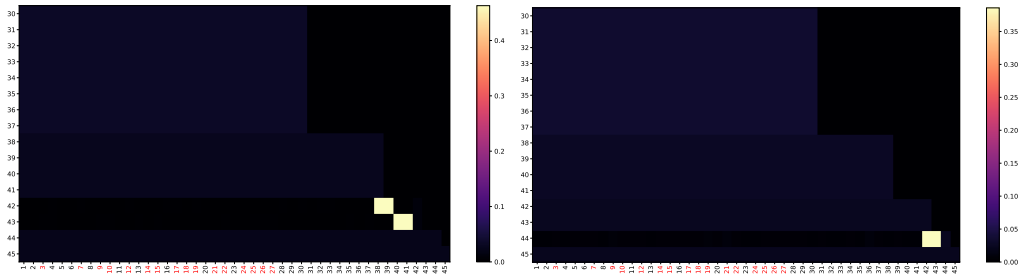


Figure 3: **Training loss of a 4-layer transformer on parity task with $k = 16$.** Dashed vertical lines indicate different training stages of Log-ICoT.



(a) Layer 1

(b) Layer 2



(c) Layer 3

(d) Layer 4

Figure 4: **Visualization of the internalization process via layer-wise attention maps.** The panels illustrate the layer-wise shift in attention patterns of a trained 4-layer transformer as explicit CoT steps are removed. The progression demonstrates how reasoning logic is progressively internalized from early-layer explicit dependencies into the hidden states of deeper layers.

their hidden states without sacrificing the sample efficiency gains of explicit CoT. By analyzing the parity learning task, we prove that transformers can achieve polynomial sample complexity while significantly reducing inference latency by eliminating the need for explicit token generation.

A core contribution of this paper is the introduction of Log-ICoT, a novel training curriculum that accelerates the internalization process. While standard ICoT training scales linearly with the length of the reasoning chain, we prove that Log-ICoT reduces this dependency to a logarithmic one, offering a provable exponential speedup in training efficiency.

REFERENCES

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: 1560–1588, 2023.

Enric Boix-Adsera, Etai Littwin, Emmanuel Abbe, Samy Bengio, and Joshua Susskind. Transformers learn through gradual rank increase. *Advances in Neural Information Processing Systems*, 36: 24519–24551, 2023.

Lei Chen, Joan Bruna, and Alberto Bietti. Distributional associations vs in-context reasoning: A study of feed-forward and attention layers. *arXiv preprint arXiv:2406.03068*, 2024.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*, 2024.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36:70757–70798, 2023.

Hengyu Fu, Tianyu Guo, Yu Bai, and Song Mei. What can a single attention layer learn? a study through the random features lens. *Advances in Neural Information Processing Systems*, 36: 11912–11951, 2023.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*, 2023.

Halil Alperen Gozeten, M Emrullah Ildiz, Xuechen Zhang, Hrayr Harutyunyan, Ankit Singh Rawat, and Samet Oymak. Continuous chain of thought enables parallel exploration and reasoning. *arXiv preprint arXiv:2505.23648*, 2025.

Tianyu Guo, Druv Pai, Yu Bai, Jiantao Jiao, Michael I Jordan, and Song Mei. Active-dormant attention heads: Mechanistically demystifying extreme-token phenomena in llms. *arXiv preprint arXiv:2410.13835*, 2024.

Tianyu Guo, Hanlin Zhu, Ruiqi Zhang, Jiantao Jiao, Song Mei, Michael I Jordan, and Stuart Russell. How do llms perform two-hop reasoning in context? *arXiv preprint arXiv:2502.13913*, 2025.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Jianhao Huang, Zixuan Wang, and Jason D Lee. Transformers learn to implement multi-step gradient descent with chain of thought. *arXiv preprint arXiv:2502.21212*, 2025a.

Yixiao Huang, Hanlin Zhu, Tianyu Guo, Jiantao Jiao, Somayeh Sojoudi, Michael I Jordan, Stuart Russell, and Song Mei. Generalization or hallucination? understanding out-of-context reasoning in transformers. *arXiv preprint arXiv:2506.10887*, 2025b.

Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 19660–19722, 2024.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2704–2713, 2018.

Samy Jelassi, Michael Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. *Advances in Neural Information Processing Systems*, 35:37822–37836, 2022.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*, 2022.

Juno Kim and Taiji Suzuki. Transformers provably solve parity efficiently with chain of thought. *arXiv preprint arXiv:2410.08633*, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Yingcong Li, Yixiao Huang, Muhammed E Ildiz, Ankit Singh Rawat, and Samet Oymak. Mechanics of next token prediction with self-attention. In *International Conference on Artificial Intelligence and Statistics*, pp. 685–693. PMLR, 2024a.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. *arXiv preprint arXiv:2402.12875*, 1, 2024b.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.

Charles London and Varun Kanade. Pause tokens strictly increase the expressivity of constant-depth transformers. *arXiv preprint arXiv:2505.21024*, 2025.

Xutao Ma, Yixiao Huang, Hanlin Zhu, and Somayeh Sojoudi. Breaking the reversal curse in autoregressive language models via identity bridge. *arXiv preprint arXiv:2602.02470*, 2026.

Arvind Mahankali, Tatsunori B Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. *arXiv preprint arXiv:2307.03576*, 2023.

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

Eshaan Nichani, Alex Damian, and Jason D Lee. How transformers learn causal structure with gradient descent. In *International Conference on Machine Learning*, pp. 38018–38070. PMLR, 2024.

Jacob Pfau, William Merrill, and Samuel R Bowman. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*, 2024.

Yunwei Ren, Zixuan Wang, and Jason D Lee. Learning and transferring sparse contextual bigrams with linear transformers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *International Conference on Machine Learning*, pp. 3067–3075. PMLR, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

DiJia Su, Hanlin Zhu, Yingchen Xu, Jiantao Jiao, Yuandong Tian, and Qinqing Zheng. Token assorted: Mixing latent and text tokens for improved language model reasoning. *arXiv preprint arXiv:2502.03275*, 2025.

Yuandong Tian, Yiping Wang, Beidi Chen, and Simon S Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. *Advances in neural information processing systems*, 36:71911–71947, 2023a.

Yuandong Tian, Yiping Wang, Zhenyu Zhang, Beidi Chen, and Simon Du. Joma: Demystifying multilayer transformers via joint dynamics of mlp and attention. *arXiv preprint arXiv:2310.00535*, 2023b.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023a.

Xinyi Wang, Lucas Caccia, Oleksiy Ostapenko, Xingdi Yuan, William Yang Wang, and Alessandro Sordoni. Guiding language model reasoning with planning tokens. *arXiv preprint arXiv:2310.05707*, 2023b.

Zixuan Wang, Stanley Wei, Daniel Hsu, and Jason D. Lee. Transformers provably learn sparse token selection while fully-connected nets cannot. In *ICML*, 2024.

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. *arXiv preprint arXiv:2505.23683*, 2025.

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, 2022.

Kaiyue Wen, Huaqing Zhang, Hongzhou Lin, and Jingzhao Zhang. From sparse dependence to sparse attention: unveiling how chain-of-thought enhances transformer sample efficiency. *arXiv preprint arXiv:2410.05459*, 2024.

Noam Wies, Yoav Levine, and Amnon Shashua. Sub-task decomposition enables learning in sequence to sequence tasks. *arXiv preprint arXiv:2204.02892*, 2022.

Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*, 2020.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. *arXiv preprint arXiv:2409.19606*, 2024a.

Hanlin Zhu, Baihe Huang, Shaolun Zhang, Michael Jordan, Jiantao Jiao, Yuandong Tian, and Stuart J Russell. Towards a theoretical understanding of the'reversal curse'via training dynamics. *Advances in Neural Information Processing Systems*, 37:90473–90513, 2024b.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Emergence of superposition: Unveiling the training dynamics of chain of continuous thought. *arXiv preprint arXiv:2509.23365*, 2025a.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. *arXiv preprint arXiv:2505.12514*, 2025b.

A Proof of Section 3

A.1 Proof of Lemma 1

Proof. Fix any $n^{\epsilon/8} \le m \le T$. Since $\boldsymbol{b}_j \sim \mathsf{Unif}(\{\pm 1\}^B)$ for $j \in [m]$. By Hoeffding's inequality, we have that:

$$Pr(\|\frac{1}{m}\sum_{\alpha\in[m]} \boldsymbol{b}_\alpha\|_\infty \ge t) \le 2B\exp(-\frac{mt^2}{2}).$$

Thus with probability $1-p_1$, we have:

$$\|\frac{1}{m}\sum_{\alpha\in[m]} \boldsymbol{b}_\alpha\|_\infty \le \sqrt{\frac{2}{m}\log(2B/p_1)}.$$

Since ϕ behaves like a quadratic near $0, \pm 1$, there exists some constant $C_2 > 0$ that depends only on ϕ such that

$$\begin{aligned}\|\phi(\frac{1}{m}\sum_{\alpha\in[m]} \boldsymbol{b}_\alpha) + \mathbf{1}_B\|_\infty = \|\phi(\frac{1}{m}\sum_{\alpha\in[m]} \boldsymbol{b}_\alpha) - \phi(\mathbf{0}_B)\|_\infty &\le C_2\left(\sqrt{\frac{2}{m}\log(2B/p_1)}\right)^2 \\ &\le C_2\frac{2}{m}\log(2B/p_1).\end{aligned}$$

Finally, assume the failure probability for each m be $\delta_m = p_1/T$. With probability $1-p_1$, for all $n^{\epsilon/8} \le m \le T < 2n$, we have

$$\|\phi(\frac{1}{m}\sum_{\alpha\in[m]} \boldsymbol{b}_\alpha) + \mathbf{1}_B\|_\infty \le C_2\frac{2}{m}\log(2TB/p_1).$$

Recall $B = \mathsf{poly}(n)$ and set $p_1 = \exp(-n^{\epsilon/16})$, we have

$$\|\phi(\frac{1}{m}\sum_{\alpha\in[m]} \boldsymbol{b}_\alpha) + \mathbf{1}_B\|_\infty \le O(n^{-\epsilon/16}).$$

□

A.2 Proof of Theorem 1

Notations. Recall that $\Psi_{t'}^\top \boldsymbol{W}_{\mathsf{O}}^{(t)} = \mathbb{1}(t=t')\left[\mathbf{0}_{B\times T}, \boldsymbol{e}_{L+1,t}^\top \otimes \boldsymbol{I}_B\right] \in \mathbb{R}^{B\times d}$. Let $\boldsymbol{S}_t = \boldsymbol{e}_{L+1,t}^\top \otimes \boldsymbol{I}_B \in \mathbb{R}^{B\times B_L}$, we denote $\boldsymbol{P}_t = \Psi_t^\top \boldsymbol{W}_{\mathsf{O}}^{(t)} = [\mathbf{0}_{B\times T}, \boldsymbol{S}_t]$, which extracts the t-th B-dimensional block. For any $\boldsymbol{x} \in \mathbb{R}^d$, we define the t-th B-block (after the first T coordinates) by

$$\boldsymbol{x}_{[t]} := S_t \boldsymbol{x}_{T:} \in \mathbb{R}^B.$$

Equivalently, we have $(\boldsymbol{x}_{[t]})_i = x_{T+B(t-1)+i}$. For a tuple of indices $(j_1, \ldots, j_r) \in [T]^r$, we consider $x_{j_1}\cdots x_{j_r}$ to be trivial if it always equals 1. E.g., the parity $x_1x_2x_9$ is trivial in Figure 1. We define the set of non-trivial tuples of length r over the first m indices as:

$$I_{r,m} = \{(j_1, \ldots, j_r) \in [m]^r, x_{j_1}\cdots x_{j_r} \not\equiv 1\}.$$

For $r = 1$, every index is non-trivial so $I_{1,m} = [m]$.

Lemma 2 (Kim & Suzuki (2024), Lemma 9)**.** *Assume $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m \in \{\pm 1\}^B$ be vectors where each bit is sampled i.i.d. from the uniform distribution. For any $p > 0$, with probability at least $1-p$, the following holds for all $r \le 4$:*

$$\max_{(j_1,\ldots,j_r)\in I_{r,m}} \frac{|\langle \boldsymbol{b}_{j_1}, \ldots, \boldsymbol{b}_{j_r}\rangle|}{B} \le \kappa := \sqrt{\frac{2}{B}\log\frac{32n^4}{p}}.$$

In particular, we set $B = \Omega(n^{2+\epsilon})$ and $p = \exp(-n^{\epsilon/2})$ such that $\kappa = O(n^{-1-\epsilon/4})$.

We now proceed to the main proof of Theorem 1.

Lemma 3 (Stage 1). *Assume the assumptions in Theorem 1 holds and let $\boldsymbol{W}^{(\ell')} = \mathbf{0}_{T\times T}$ for $\ell' \geq 1$. After one-step gradient descent on the first stage $\mathcal{L}^{(1)}$ with B training samples and learning rate η with $B = \Omega(n^{2+\epsilon}), \eta = \Theta(n^{2+\epsilon/16})$, it holds for all $m \in [n_1, n_2 - 1]$ that*

$$\frac{1 - n\exp(-Cn^{\epsilon/16})}{2} \leq \sigma_{c_1[m+1]}(\boldsymbol{w}_m^{(1)}), \sigma_{c_2[m+1]}(\boldsymbol{w}_m^{(1)}) \leq \frac{1}{2}.$$

and

$$\|\boldsymbol{x}_{m,[2]}^{(1)} - \boldsymbol{b}_{m+1}\|_\infty \leq \exp(-\Omega(n^{\epsilon/16}))$$

Proof. We first compute the gradient. Recall that at stage t, the only relevant output is from layer t.

$$\begin{aligned}
\mathcal{L}^{(t)}(\boldsymbol{\theta}) &= \frac{1}{2B}\sum_{m=n_t}^{n_{t+1}-1} \|\Psi_t^\top \mathcal{T}_{\boldsymbol{\theta}}(\boldsymbol{X})_m - \boldsymbol{b}_{m+1}\|^2 \\
&= \frac{1}{2B}\sum_{m=n_t}^{n_{t+1}-1} \|\Psi_t^\top \boldsymbol{W}_{\mathsf{O}}^{(t)} \left(g_m^{(t)}\phi(\sum_{j=1}^{m}\sigma_j(\boldsymbol{w}_m^{(t)})\boldsymbol{x}_j^{(t-1)}) + (1-g_m^{(t)})\boldsymbol{x}_m^{(t-1)} \right) - \boldsymbol{b}_{m+1}\|^2.
\end{aligned}$$

When $t = 1$, we have $g_m^{(1)} = 1$ for $m \geq n_1$ as $\Psi_0^\top \boldsymbol{x}_m^{(0)} = \boldsymbol{b}_m \not\equiv \mathbf{1}_B$. Thus we can simplify the loss to

$$\begin{aligned}
\mathcal{L}^{(1)}(\boldsymbol{\theta}) &= \frac{1}{2B}\sum_{m=n_1}^{n_2-1} \|\Psi_1^\top \boldsymbol{W}_{\mathsf{O}}^{(1)} \left(g_m^{(1)}\phi(\sum_{j=1}^{m}\sigma_j(\boldsymbol{w}_m^{(1)})\boldsymbol{x}_j^{(0)}) + (1-g_m^{(1)})\boldsymbol{x}_m^{(0)} \right) - \boldsymbol{b}_{m+1}\|^2 \\
&= \frac{1}{2B}\sum_{m=n_1}^{n_2-1} \|\Psi_1^\top \boldsymbol{W}_{\mathsf{O}}^{(1)} \phi\left(\sum_{j=1}^{m}\sigma_j(\boldsymbol{w}_m^{(1)})\,\boldsymbol{x}_j^{(0)} \right) - \boldsymbol{b}_{m+1}\|^2.
\end{aligned}$$

Fix any $m \in [n_1, n_2 - 1]$. Let $\hat{\boldsymbol{z}}_m^{(1)} = \sum_{j=1}^{m}\sigma_j(\boldsymbol{w}_m^{(1)})\boldsymbol{x}_j^{(0)}$. For any $1 \leq \alpha < m$, we have

$$\frac{\partial \sigma_\alpha(\boldsymbol{w}_m^{(1)})}{\partial w_{j,m}^{(1)}} = (\mathbb{1}(j=\alpha) - \sigma_\alpha(\boldsymbol{w}_m^{(1)}))\sigma_j(\boldsymbol{w}_m^{(1)}) = (\mathbb{1}(j=\alpha) - \sigma_j(\boldsymbol{w}_m^{(1)}))\sigma_\alpha(\boldsymbol{w}_m^{(1)}),$$

and by linearity,

$$\frac{\partial \hat{\boldsymbol{z}}_m^{(1)}}{\partial w_{j,m}^{(1)}} = \sum_{\alpha=1}^{m-1}(\mathbb{1}(j=\alpha) - \sigma_j(\boldsymbol{w}_m^{(1)}))\sigma_\alpha(\boldsymbol{w}_m^{(1)})\boldsymbol{x}_\alpha^{(0)} = \sigma_j(\boldsymbol{w}_m^{(1)})\big(\boldsymbol{x}_j^{(0)} - \hat{\boldsymbol{z}}_m^{(1)}\big).$$

Then the gradient of $\mathcal{L}^{(1)}$ w.r.1 $w_{j,m}^{(1)}$ is given by

$$\begin{aligned}
\frac{\partial \mathcal{L}^{(1)}}{\partial w_{j,m}^{(1)}} =& \frac{1}{B}\big(\Psi_1^\top \boldsymbol{W}_{\mathsf{O}}^{(1)}\phi(\hat{\boldsymbol{z}}_m^{(1)}) - \boldsymbol{b}_{m+1}\big)^\top \left(\Psi_1^\top \boldsymbol{W}_{\mathsf{O}}^{(1)}\right)^\top \frac{\partial \phi(\hat{\boldsymbol{z}}_m^{(1)})}{\partial w_{j,m}^{(1)}} \\
=& \frac{1}{B}\left(\boldsymbol{P}_1^\top\left(\boldsymbol{P}_1\phi(\hat{\boldsymbol{z}}_m^{(1)}) - \boldsymbol{b}_{m+1}\right)\right)^\top \frac{\partial \phi(\hat{\boldsymbol{z}}_m^{(1)})}{\partial w_{j,m}^{(1)}} \\
=& \frac{\sigma_j(\boldsymbol{w}_m^{(1)})}{B}\,\langle \boldsymbol{P}_1^\top(\boldsymbol{P}_1\phi(\hat{\boldsymbol{z}}_m^{(1)}) - \boldsymbol{b}_{m+1}), \phi'(\hat{\boldsymbol{z}}_m^{(1)}), \boldsymbol{x}_j^{(0)} - \hat{\boldsymbol{z}}_m^{(1)}\rangle.
\end{aligned}$$

In the following analysis, we ignore the superscript of $t = 1$ for the states. At initialization, we have $\sigma_j(\boldsymbol{w}_m) = \frac{1}{n}$ for all $j \in [n]$. Thus we get

$$\begin{aligned}\frac{\partial \mathcal{L}^{(1)}}{\partial w_{j,m}^{(1)}} =& \frac{\sigma_j(\boldsymbol{w}_m)}{B} \langle \boldsymbol{P}_1^\top (\boldsymbol{P}_1 \phi(\hat{\boldsymbol{z}}_m) - \boldsymbol{b}_{m+1}), \phi'(\hat{\boldsymbol{z}}_m), \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m \rangle \\ =& -\frac{1}{nB} \langle \boldsymbol{P}_1^\top \boldsymbol{b}_{m+1}, 2c\hat{\boldsymbol{z}}_m, \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m \rangle & (2) \\ &+ \frac{1}{nB} \langle \boldsymbol{P}_1^\top \boldsymbol{P}_1(-\mathbf{1}_d + c\hat{\boldsymbol{z}}_m^2), 2c\hat{\boldsymbol{z}}_m, \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m \rangle & (3) \\ &+ \frac{1}{nB} \langle O(\boldsymbol{P}_1^\top \boldsymbol{P}_1 |\hat{z}_m|^4), 2c\hat{\boldsymbol{z}}_m, \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m \rangle & (4) \\ &+ \frac{1}{nB} \langle \boldsymbol{P}_1^\top (\boldsymbol{P}_1 \phi(\hat{\boldsymbol{z}}_m) - \boldsymbol{b}_{m+1}), O(|\hat{\boldsymbol{z}}_m|^3), \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m \rangle & (5)\end{aligned}$$

For Eq. (2), we substitute $\hat{\boldsymbol{z}}_m = \frac{1}{n} \sum_{\alpha \in [n]} \boldsymbol{x}_\alpha$ at initialization to expand

$$\frac{1}{B} \langle \boldsymbol{P}_1^\top \boldsymbol{b}_{m+1}, \hat{\boldsymbol{z}}_m, \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m \rangle = \frac{1}{nB} \sum_{\alpha \in [n]} \langle \boldsymbol{P}_1^\top \boldsymbol{b}_{m+1}, \boldsymbol{x}_\alpha, \boldsymbol{x}_j \rangle - \frac{1}{n^2 B} \sum_{\alpha, \beta \in [n]} \langle \boldsymbol{P}_1^\top \boldsymbol{b}_{m+1}, \boldsymbol{x}_\alpha, \boldsymbol{x}_\beta \rangle$$

Recall $\boldsymbol{S}_t = \boldsymbol{e}_{L+1,t}^\top \otimes \boldsymbol{I}_B \in \mathbb{R}^{B \times B_L}$. Since $\boldsymbol{P}_1 = \Psi_1^\top \boldsymbol{W}_{\mathsf{O}}^{(1)} = [\mathbf{0}_{B \times T}, \boldsymbol{S}_1]$, we have $\boldsymbol{P}_1^\top \boldsymbol{b}_{m+1}$ be supported on the block 1 only. Consider any fixed α, β:

$$\langle \boldsymbol{P}_1^\top \boldsymbol{b}_{m+1}, \boldsymbol{x}_\alpha, \boldsymbol{x}_\beta \rangle = \sum_{i=1}^{d} (\boldsymbol{P}_1^\top \boldsymbol{b}_{m+1})_i x_{\alpha,i} x_{\beta,i} = \langle \boldsymbol{b}_{m+1}, \boldsymbol{x}_{\alpha,[1]}, \boldsymbol{x}_{\beta,[1]} \rangle = \langle \boldsymbol{b}_{m+1}, \boldsymbol{b}_\alpha, \boldsymbol{b}_\beta \rangle$$

When $t = 1$, we have $n \le m \le n_2 - 1$ and $\alpha, \beta \in [n]$ thus $h[m+1] = 2$ and $h[\alpha], h[\beta] = 1$. Note that $b_{m+1} b_\alpha b_\beta$ will be trivial iff $\{\alpha, \beta\} = \{c[m+1], c_2[m+1]\}$, where $\langle \boldsymbol{b}_{m+1}, \boldsymbol{b}_\alpha, \boldsymbol{b}_\beta \rangle = B$. Thus we have that

$$\frac{1}{B} \sum_{\alpha, \beta \in [n]} \langle \boldsymbol{b}_{m+1}, \boldsymbol{b}_\alpha, \boldsymbol{b}_\beta \rangle = 2 + \frac{1}{B} \sum_{(m+1, \alpha, \beta) \in I_{3,m+1}} \langle \boldsymbol{b}_{m+1}, \boldsymbol{b}_\alpha, \boldsymbol{b}_\beta \rangle = 2 + O(n^2 \kappa)$$

Similarly, the contraction $\langle \boldsymbol{b}_{m+1}, \boldsymbol{b}_\alpha, \boldsymbol{b}_j \rangle$ is nontrivial only when $p[j] = m+1$ and α is the other child node of b_{m+1}. As a result,

$$\frac{1}{B} \sum_{\alpha \in [n]} \langle \boldsymbol{b}_{m+1}, \boldsymbol{b}_\alpha, \boldsymbol{b}_j \rangle = \begin{cases} 1 + O(n\kappa) & p[j] = m+1, \\ O(n\kappa) & o.w. \end{cases}$$

where $\kappa = O(n^{-1-\epsilon/4})$. Combining these, we get

$$\begin{aligned}& -\frac{1}{nB} \langle \boldsymbol{P}_t^\top \boldsymbol{b}_{m+1}, 2c\hat{\boldsymbol{z}}_m, \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m \rangle \\ &= -\frac{2c}{n^2} (\mathbb{1}(p[j] = m+1) + O(n\kappa)) + \frac{2c}{n^3} (2 + O(n^2 \kappa)) \\ &= -\frac{2c}{n^2} \mathbb{1}(p[j] = m+1) + O(n^{-2-\epsilon/4}).\end{aligned}$$

Next for Eq. (3), we expand

$$\begin{aligned}& \frac{1}{B} \langle \boldsymbol{P}_1^\top \boldsymbol{P}_1(-\mathbf{1}_d + c\hat{\boldsymbol{z}}_m^2), 2c\hat{\boldsymbol{z}}_m, \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m \rangle \\ &= -\frac{2c}{B} \langle \boldsymbol{P}_1^\top \boldsymbol{P}_1 \hat{\boldsymbol{z}}_m, \boldsymbol{x}_j \rangle + \frac{2c}{B} \langle \boldsymbol{P}_1^\top \boldsymbol{P}_1 \hat{\boldsymbol{z}}_m^2 \rangle + \frac{2c^2}{B} \langle \boldsymbol{P}_1^\top \boldsymbol{P}_1 \hat{\boldsymbol{z}}_m^3, \boldsymbol{x}_j \rangle - \frac{2c^2}{B} \langle \boldsymbol{P}_1^\top \boldsymbol{P}_1 \hat{\boldsymbol{z}}_m^4 \rangle \\ &= -\frac{2c}{B} \langle \hat{\boldsymbol{z}}_{m,[1]}, \boldsymbol{x}_{j,[1]} \rangle + \frac{2c}{B} \langle \hat{\boldsymbol{z}}_{m,[1]}^2 \rangle + \frac{2c^2}{B} \langle \hat{\boldsymbol{z}}_{m,[1]}^3, \boldsymbol{x}_{j,[1]} \rangle - \frac{2c^2}{B} \langle \hat{\boldsymbol{z}}_{m,[1]}^4 \rangle\end{aligned}$$

For the second-order terms, we have:

$$\frac{1}{B} \langle \hat{\boldsymbol{z}}_{m,[1]}, \boldsymbol{x}_{j,[1]} \rangle = \frac{1}{nB} \left(\langle \boldsymbol{b}_j, \boldsymbol{b}_j \rangle + \sum_{\alpha \neq j} \langle \boldsymbol{b}_\alpha, \boldsymbol{b}_j \rangle \right) = \frac{1}{n} + O(\kappa)$$

$$\frac{1}{B} \langle \hat{\boldsymbol{z}}_{m,[1]}^2 \rangle = \frac{1}{n^2 B} \left(\sum_{\alpha} \langle \boldsymbol{b}_\alpha, \boldsymbol{b}_\alpha \rangle + \sum_{\alpha \neq \beta} \langle \boldsymbol{b}_\alpha, \boldsymbol{b}_\beta \rangle \right) = \frac{1}{n} + O(\kappa)$$

For the fourth-order interaction terms, we dicsuss when $(\alpha, \beta, \gamma, \delta) \notin I_{4,m}$. Without loss of generality, assume $h[\alpha] \le h[\beta] \le h[\gamma] \le h[\delta]$.

1. If $h[\beta] < h[\gamma] < h[\delta]$, to cancel out b_δ, it must hold that b_γ is a child of b_δ and b_α, b_β are the two children of the other child. This pattern is fully determined by choosing δ and which child is γ. Thus there are only $O(n)$ such trivial 4-tuples.
2. If $h[\beta] = h[\gamma] < h[\delta]$, it must also hold that $h[\gamma] = h[\delta] - 1$. Then b_β and b_γ must both be the children nodes of b_δ. Then we have either $b_\beta = b_\gamma$ or $b_\delta = b_\beta b_\gamma$, where both cases cannot be trivial.
3. If $h[\beta] < h[\gamma] = h[\delta]$, we have $\gamma = \delta$ and $\alpha = \beta$ such that $b_\gamma b_\delta \equiv 1$ and $b_\alpha b_\beta \equiv 1$. There are $O(n^2)$ different choices.
4. If $h[\beta] = h[\gamma] = h[\delta]$, again we have two of the indices to be the same and so are the remaining two. Thus there are $O(n^2)$ trivial 4-tuples.

Therefore,

$$\begin{aligned}
\frac{1}{B}\langle \hat{\boldsymbol{z}}^4_{m,[1]} \rangle &= \frac{1}{n^4 B} \sum_{\alpha,\beta,\gamma,\delta} \langle \boldsymbol{b}_\alpha, \boldsymbol{b}_\beta, \boldsymbol{b}_\gamma, \boldsymbol{b}_\delta \rangle \\
&= \frac{1}{n^4 B} \left(\sum_{\alpha,\beta,\gamma,\delta \in I_{4,m}} \langle \boldsymbol{b}_\alpha, \boldsymbol{b}_\beta, \boldsymbol{b}_\gamma, \boldsymbol{b}_\delta \rangle + \sum_{\alpha,\beta,\gamma,\delta \notin I_{4,m}} \langle \boldsymbol{b}_\alpha, \boldsymbol{b}_\beta, \boldsymbol{b}_\gamma, \boldsymbol{b}_\delta \rangle \right) \\
&= \frac{1}{n^4 B} \left(\sum_{\alpha,\beta,\gamma,\delta \in I_{4,m}} O(B\kappa) + \sum_{\alpha,\beta,\gamma,\delta \notin I_{4,m}} B \right) \\
&\le \frac{1}{n^4} \left(m^4 O(\kappa) + O(n^2) \right) \\
&\le O(n^{-2} + \kappa)
\end{aligned}$$

Now for $\langle \hat{\boldsymbol{z}}^3_{m,[1]}, \boldsymbol{x}_{j,[1]} \rangle$, assuming index j is contained in $(\alpha, \beta, \gamma, \delta)$. Then for case 1 we only have $O(1)$ trivial tuples and both case 3 and 4 are reduced to $O(n)$ as there's only one free index to be determined.

$$\frac{1}{B}\langle \hat{\boldsymbol{z}}^3_{m,[1]}, \boldsymbol{x}_{j,[1]} \rangle = \frac{1}{n^3 B} \sum_{\alpha,\beta,\gamma} \langle \boldsymbol{b}_\alpha, \boldsymbol{b}_\beta, \boldsymbol{b}_\gamma, \boldsymbol{b}_j \rangle \le O(\kappa) + \frac{O(n)}{n^3} = O(n^{-2} + \kappa)$$

Thus

$$\frac{1}{nB}\langle \boldsymbol{P}_1^\top \boldsymbol{P}_1(-\mathbf{1}_d + c\hat{\boldsymbol{z}}_m^2), 2c\hat{\boldsymbol{z}}_m, \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m \rangle = -\frac{2c}{n}(\frac{1}{n} - \frac{1}{n}) + \frac{O(\kappa)}{n} = O(n^{-2-\epsilon/4})$$

For Eq. (4), note that

$$\frac{1}{B}\langle \boldsymbol{P}_t^\top \boldsymbol{P}_t |\hat{z}_m|^4 \rangle = \frac{1}{B}\langle |\hat{\boldsymbol{z}}^4_{m,[1]}| \rangle = \frac{1}{B}\langle \hat{\boldsymbol{z}}^4_{m,[1]} \rangle = O(n^{-2} + \kappa)$$

and since $2c\hat{\boldsymbol{z}}_m, \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m$ are contained in $[-1, 1]$ and $[-2, 2]$ respectively. We have

$$\frac{1}{nB}\langle O(\boldsymbol{P}_t^\top \boldsymbol{P}_t |\hat{z}_m|^4), 2c\hat{\boldsymbol{z}}_m, \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m \rangle = \frac{4c}{m} O(n^{-2} + \kappa) = O(n^{-2-\epsilon/4})$$

Finally for Eq. (5), using Cauchy-Schwarz we have

$$\begin{aligned}
\frac{1}{B}\langle \boldsymbol{P}_t^\top \boldsymbol{P}_t |\hat{z}_m|^3 \rangle = \frac{1}{B}\langle |\hat{\boldsymbol{z}}^3_{m,[1]}| \rangle &= \frac{1}{B} \sum_{i=1}^{B} |\hat{\boldsymbol{z}}^3_{m,[1],i}| \\
&\le \frac{1}{B} \left(\sum_{i=1}^{B} \hat{\boldsymbol{z}}^2_{m,[1],i} \right)^{1/2} \left(\sum_{i=1}^{B} \hat{\boldsymbol{z}}^4_{m,[1],i} \right)^{1/2} \\
&= O(n^{-1-\epsilon/8})
\end{aligned}$$

By the definition of $\boldsymbol{P}_t$, we get

$$\begin{aligned}
&\frac{1}{nB}\langle \boldsymbol{P}_t^\top(\boldsymbol{P}_t\phi(\hat{\boldsymbol{z}}_m) - \boldsymbol{b}_{m+1}), O(|\hat{\boldsymbol{z}}_m|^3), \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m\rangle \\
&= \frac{1}{nB}\langle \phi(\hat{\boldsymbol{z}}_m) - \boldsymbol{P}_t^\top \boldsymbol{b}_{m+1}), O(\boldsymbol{P}_t^\top \boldsymbol{P}_t|\hat{\boldsymbol{z}}_m|^3), \boldsymbol{x}_j - \hat{\boldsymbol{z}}_m\rangle \\
&= \frac{4}{nB}\langle O(\boldsymbol{P}_t^\top \boldsymbol{P}_t|\hat{\boldsymbol{z}}_m|^3)\rangle \\
&= O(n^{-2-\epsilon/8})
\end{aligned}$$

Combining all terms from (2) to (5), we get

$$\frac{\partial \mathcal{L}^{(1)}}{\partial w_{j,m}^{(1)}} = -\frac{2c}{n^2}\mathbb{1}(p[j] = m+1) + O(n^{-2-\epsilon/8}).$$

Note that this applies to the approximate gradient $\tilde{\nabla}_{w_{j,m}^{(1)}}\mathcal{L}^{(1)}$ too since each component of the bnoise is bounded by $O(n^{-2-\epsilon/8})$.

Concentration of attention scores. Taking $\eta = n^{2+\epsilon/16}\eta_0$, the updated weight becomes

$$w_{j,m}^{(1)}(1) = -\eta\tilde{\nabla}_{w_{j,m}^{(1)}}\mathcal{L}^{(1)} = 2c\eta_0\frac{n^{2+\epsilon/16}}{n^2}\mathbb{1}(p[j] = m+1) + O(n^{-\epsilon/16}).$$

We choose η_0 such that the leading terms are not half-integers. For sufficiently large n, the nearest-integer operator $q(\cdot)$ absorbs the noise terms. Specifically, for any $j \notin \{c_1[m+1], c_2[m+1]\}$, we have

$$w_{j,m}^{(1)} = q\left(O(n^{-\epsilon/16})\right) = 0.$$

Conversely, for the indices corresponding to the signal, the weight concentrate as

$$w_{c_1[m+1],m}^{(1)} = w_{c_2[m+1],m}^{(1)} = q\left(2c\eta_0 n^{\epsilon/16}\right).$$

The softmax scores for the noise terms can thus be upper bounded by

$$\sigma_j(\boldsymbol{w}_m^{(1)}) \le \frac{1}{\exp\left(w_{c_1[m+1],m}^{(1)}\right) + \exp\left(w_{c_2[m+1],m}^{(1)}\right) + n - 2} \le \exp(-Cn^{\epsilon/16}).$$

Since the softmax scores must sum to 1, it holds that

$$\frac{1 - n\exp(-Cn^{\epsilon/16})}{2} \le \sigma_{c_1[m+1]}(\boldsymbol{w}_m^{(1)}), \sigma_{c_2[m+1]}(\boldsymbol{w}_m^{(1)}) \le \frac{1}{2}.$$

Evaluating the forward pass. For any $m \in [n, n+n_1-1]$, we have

$$\begin{aligned}
&\|\hat{\boldsymbol{z}}_{m,[1]}^{(1)} - \frac{\boldsymbol{b}_{c_1[m+1]} + \boldsymbol{b}_{c_2[m+1]}}{2}\|_\infty = \left\|\sum_j \sigma_j(\boldsymbol{w}_m^{(1)})\boldsymbol{b}_j - \frac{\boldsymbol{b}_{c_1[m+1]} + \boldsymbol{b}_{c_2[m+1]}}{2}\right\|_\infty \\
&\le \sum_{p[j]\neq m+1}\sigma_j(\boldsymbol{w}_m^{(1)}) + \left|\sigma_{c_1[m+1]}(\boldsymbol{w}_m^{(1)}) - \frac{1}{2}\right| + \left|\sigma_{c_2[m+1]}(\boldsymbol{w}_m^{(1)}) - \frac{1}{2}\right| \\
&\le 2(n-1)\exp(-Cn^{\epsilon/16})
\end{aligned}$$

Following the Taylor expansion of ϕ, we get:

$$\begin{aligned}
\left\|\boldsymbol{x}_{[2]}^{(1)} - \boldsymbol{b}_{m+1}\right\|_\infty = \left\|\Psi_1^\top \boldsymbol{x}^{(1)} - \boldsymbol{b}_{m+1}\right\|_\infty &= \left\|\boldsymbol{P}_1^\top \boldsymbol{P}_1\phi(\hat{\boldsymbol{z}}_m^{(1)}) - \phi\left(\frac{\boldsymbol{b}_{c_1[m+1]} + \boldsymbol{b}_{c_2[m+1]}}{2}\right)\right\|_\infty \\
&= \left\|\phi(\hat{\boldsymbol{z}}_{m,[1]}^{(1)}) - \phi\left(\frac{\boldsymbol{b}_{c_1[m+1]} + \boldsymbol{b}_{c_2[m+1]}}{2}\right)\right\|_\infty \\
&\le C_2\left(2(n-1)\exp(-Cn^{\epsilon/16})\right)^2 \\
&\le \exp(-\Omega(n^{\epsilon/16})) \qquad (6)
\end{aligned}$$

□

At each stage $t \geq 1$, we replace the first $k(1-2^{-(t-1)})$ thinking tokens with padding token. This ensures that at stage $v := \log_2 k$ stages, we are only relying on the input sequence to predict. The input sequence looks like

$$\boldsymbol{b}^{(t)} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \underbrace{\mathbf{0}_B, \ldots, \mathbf{0}_B}_{n_t}, \boldsymbol{b}_{n_t}, \boldsymbol{b}_{n_t+1}, \ldots, \boldsymbol{b}_T)$$

The loss function is only computed on position $n_t \leq m < n_{t+1}$, i.e.,

$$\mathcal{L}^{(t)}(\boldsymbol{\theta}) = \frac{1}{2B} \sum_{m=n_t}^{n_{t+1}-1} \|\hat{f}(\boldsymbol{D})_m - \boldsymbol{b}_{m+1}\|^2$$

Similar to the first stage, we can learn the t-th layer by one step of gradient descent. Before proceeding, we first state a lemma that characterizes how the hidden states progress over layers.

Lemma 4 (Characterization of Hidden States)**.** *Suppose that for all layers $\tau \in \{1, \ldots, \ell\}$, the states $\mathbf{x}^{(\tau)}$ follow the recursive form:*

$$\boldsymbol{x}^{(\tau)}_{m,[\tau+1]} = \begin{cases} \boldsymbol{b}_{m+1} + \exp(-Cn^{\epsilon/16}) & n_\tau \leq m < n_{\tau+1} \\ \boldsymbol{x}^{(\tau-1)}_{m,[\tau]} & m < n_\tau \end{cases} \tag{7}$$

Then, the state at layer ℓ is characterized by:

$$\boldsymbol{x}^{(\ell)}_{m,[\ell+1]} = \begin{cases} \boldsymbol{b}_{m+1} + \exp(-Cn^{\epsilon/16}) & n \leq m < n_{\ell+1} \\ \boldsymbol{b}_m & m < n \end{cases} \tag{8}$$

Proof. We prove this by induction on the layer index ℓ. Recall that $n_\ell = n + k(1-2^{-(\ell-1)})$, where $n_1 = n$. When $\ell = 1$, we have $n_\ell = n$. Moreover, for the input region $m < n$, by Eq. (8) we have $\boldsymbol{x}^{(\ell-1)}_{m,[\ell]} = \boldsymbol{x}^{(0)}_{m,[1]} \overset{(a)}{=} \boldsymbol{b}_m$ where (a) follows the definition of the embedding layer. Now suppose the hypothesis is true for layer ℓ, we want to show the state at $\ell+1$ follows Eq. (8) as well.

- For the newly included nodes $n_{\ell+1} \leq m < n_{\ell+2}$, by Eq. (7), we have $\boldsymbol{x}^{(\ell+1)}_{m,[\ell+2]} = \boldsymbol{b}_{m+1} + \exp(-Cn^{\epsilon/16})$.
- For $n \leq m < n_{\ell+1}$, Eq. (7) indicates the state is inherited from the previous layer: $\boldsymbol{x}^{(\ell+1)}_{m,[\ell+2]} = \boldsymbol{x}^{(\ell)}_{m,[\ell+1]}$. by the inductive hypothesis, this value is $\boldsymbol{b}_{m+1} + \exp(-Cn^{\epsilon/16})$.
- For $m < n$, the state is again inherited layer-by-layer: $\boldsymbol{x}^{(\ell+1)}_{m,[\ell+2]} = \boldsymbol{x}^{(\ell)}_{m,[\ell+1]}$. By the inductive hypothesis, $\boldsymbol{x}^{(\ell)}_{m,[\ell+1]} = \boldsymbol{b}_m$.

Combining these, we can get the proposed result. □

Lemma 5 (Stage t)**.** *Assume $\boldsymbol{W}^{(\ell')} = \mathbf{0}_{T\times T}$ for $\ell' \geq t+1$. After one-step gradient descent on the $(t+1)$-th stage loss $\mathcal{L}^{(t+1)}$ with B training samples and learning rate η with $B = \Omega(n^{2+\epsilon}), \eta = \Theta(n^{2+\epsilon/16})$. Then for all $m \in [n_{t+1}, n_{t+2}-1]$, we have*

$$\frac{1 - n\exp(-Cn^{\epsilon/16})}{2} \leq \sigma_{c_1[m+1]}(\boldsymbol{w}_m^{(t+1)}), \sigma_{c_2[m+1]}(\boldsymbol{w}_m^{(t+1)}) \leq \frac{1}{2}$$

and

$$\|\boldsymbol{x}^{(t+1)}_{m,[t+2]} - \boldsymbol{b}_{m+1}\|_\infty \leq \exp(-\Omega(n^{\epsilon/16}))$$

Proof. Following Lemma 3, we have $\Psi_0^\top \hat{\boldsymbol{z}}_m^{(1)} = \hat{\boldsymbol{z}}^{(1)}_{m,[1]} = \frac{1}{2}(\boldsymbol{b}_{c_1[m+1]} + \boldsymbol{b}_{c_2[m+1]}) + \exp(-Cn^{\epsilon/16})$, which passes the filter. thus we have

$$g_m^{(1)} = \begin{cases} 0 & m \leq n-1 \\ 1 & n \leq m < n_2 \end{cases}$$

and Eq. (7) holds when $t = 1$ by Eq. (6) in Lemma 3. Assume the connection weights are given by

$$g_m^{(t)} = \begin{cases} 0 & m \le n_t - 1 \\ 1 & n_t \le m < n_{t+1} \end{cases},$$

and that the recursive in Eq. (7) holds for all layers up to $t \ge 1$, our goal is to prove both the connection-weight condition and the recursion continue to hold for layer $t+1$ after training on $\mathcal{L}^{(t+1)}$. To simplify the notation, let

$$\bar{\boldsymbol{x}}_{\tau,[t+1]}^{(t)} = \begin{cases} \boldsymbol{b}_{\tau+1} & n \le \tau < n_{t+1} \\ \boldsymbol{b}_\tau & \tau < n \end{cases}$$

By Lemma 4, we have $\boldsymbol{x}_{\tau,[t+1]}^{(t)} = \bar{\boldsymbol{x}}_{\tau,[t+1]}^{(t)} + \boldsymbol{\xi}_{\tau,[t+1]}^{(t)}$ where

$$\|\boldsymbol{\xi}_{\tau,[t+1]}^{(t)}\|_\infty \le \delta_\tau = \begin{cases} \delta := \exp(-Cn^{\epsilon/16}) & n \le \tau < n_{t+1} \\ 0 & \tau < n \end{cases}.$$

Consider any $n_{t+1} \le m < n_{t+2}$. Based on the attention mask in Figure 2, we have $\sigma_j(\boldsymbol{w}_m) = \frac{1}{n_{t+1}-1}$ for $j \le n_{t+1} - 1$.

$$\begin{aligned}
\frac{\partial \mathcal{L}^{(t+1)}}{\partial w_{j,m}^{(t+1)}} =& \frac{\sigma_j(\boldsymbol{w}_m^{(t+1)})}{B} \langle \boldsymbol{P}_{t+1}^\top (\boldsymbol{P}_{t+1}\phi(\hat{\boldsymbol{z}}_m^{(t+1)}) - \boldsymbol{b}_{m+1}), \phi'(\hat{\boldsymbol{z}}_m^{(t+1)}), \boldsymbol{x}_j^{(t)} - \hat{\boldsymbol{z}}_m^{(t+1)} \rangle \\
=& -\frac{1}{(n_{t+1}-1)B} \langle \boldsymbol{P}_{t+1}^\top \boldsymbol{b}_{m+1}, 2c\hat{\boldsymbol{z}}_m^{(t+1)}, \boldsymbol{x}_j^{(t)} - \hat{\boldsymbol{z}}_m^{(t+1)} \rangle && (9) \\
&+ \frac{1}{(n_{t+1}-1)B} \langle \boldsymbol{P}_{t+1}^\top \boldsymbol{P}_{t+1}(-\mathbf{1}_d + c(\hat{\boldsymbol{z}}_m^{(t+1)})^2), 2c\hat{\boldsymbol{z}}_m^{(t+1)}, \boldsymbol{x}_j^{(t)} - \hat{\boldsymbol{z}}_m^{(t+1)} \rangle && (10) \\
&+ \frac{1}{(n_{t+1}-1)B} \langle O(\boldsymbol{P}_{t+1}^\top \boldsymbol{P}_{t+1} |\hat{\boldsymbol{z}}_m^{(t+1)}|^4), 2c\hat{\boldsymbol{z}}_m^{(t+1)}, \boldsymbol{x}_j^{(t)} - \hat{\boldsymbol{z}}_m^{(t+1)} \rangle && (11) \\
&+ \frac{1}{(n_{t+1}-1)B} \langle \boldsymbol{P}_{t+1}^\top (\boldsymbol{P}_{t+1}\phi(\hat{\boldsymbol{z}}_m^{(t+1)}) - \boldsymbol{b}_{m+1}), O(|\hat{\boldsymbol{z}}_m^{(t+1)}|^3), \boldsymbol{x}_j^{(t)} - \hat{\boldsymbol{z}}_m^{(t+1)} \rangle. && (12)
\end{aligned}$$

For Equation (9), we substitute $\hat{\boldsymbol{z}}_m^{(t+1)} = \frac{1}{n_{t+1}-1} \sum_{\alpha \in [n_{t+1}-1]} \boldsymbol{x}_\alpha^{(t)}$ to expand

$$\begin{aligned}
&\frac{1}{B} \langle \boldsymbol{P}_{t+1}^\top \boldsymbol{b}_{m+1}, \hat{\boldsymbol{z}}_m^{(t+1)}, \boldsymbol{x}_j^{(t)} - \hat{\boldsymbol{z}}_m^{(t+1)} \rangle \\
&= \frac{1}{(n_{t+1}-1)B} \left(\sum_\alpha \langle \boldsymbol{P}_{t+1}^\top \boldsymbol{b}_{m+1}, \boldsymbol{x}_\alpha^{(t)}, \boldsymbol{x}_j^{(t)} \rangle - \frac{1}{B} \sum_{\alpha,\beta} \langle \boldsymbol{P}_{t+1}^\top \boldsymbol{b}_{m+1}, \boldsymbol{x}_\alpha^{(t)}, \boldsymbol{x}_\beta^{(t)} \rangle \right) \\
&= \frac{1}{(n_{t+1}-1)B} \left(\sum_\alpha \langle \boldsymbol{b}_{m+1}, \boldsymbol{x}_{\alpha,[t+1]}^{(t)}, \boldsymbol{x}_{j,[t+1]}^{(t)} \rangle - \frac{1}{B} \sum_{\alpha,\beta} \langle \boldsymbol{b}_{m+1}, \boldsymbol{x}_{\alpha,[t+1]}^{(t)}, \boldsymbol{x}_{\beta,[t+1]}^{(t)} \rangle \right).
\end{aligned}$$

Since the query token only sees token prior to its level and suppose $\ell = h[m+1]$, we have $h[\alpha], h[\beta] \le \ell - 1$. Similar to Lemma 3, $b_{m+1} b_{\alpha+1} b_{\beta+1}$ will be trivial iff $h[\alpha+1] = h[\beta+1] = \ell - 1$ and $\{\alpha+1, \beta+1\} = \{c[m+1], c_2[m+1]\}$. This implies $\alpha, \beta \ge n$ as $m \ge n_{t+1}$. Since $\bar{\boldsymbol{x}}_{\tau,[t+1]}^{(t)} = \boldsymbol{b}_{\tau+1}$ for $\tau \ge n$, we get

$$\begin{aligned}
\langle \boldsymbol{b}_{m+1}, \boldsymbol{x}_{\alpha,[t+1]}^{(t)}, \boldsymbol{x}_{\beta,[t+1]}^{(t)} \rangle &= |\langle \boldsymbol{b}_{m+1}, \bar{\boldsymbol{x}}_{\alpha,[t+1]}^{(t)}, \bar{\boldsymbol{x}}_{\beta,[t+1]}^{(t)} \rangle| + 2\delta |\langle \boldsymbol{b}_{m+1}, \bar{\boldsymbol{x}}_{\alpha,[t+1]}^{(t)} \rangle| + \delta^2 |\langle \boldsymbol{b}_{m+1} \rangle| \\
&\le |\langle \boldsymbol{b}_{m+1}, \boldsymbol{b}_{\alpha+1}, \boldsymbol{b}_{\beta+1} \rangle| + 2\delta |\langle \boldsymbol{b}_{m+1}, \boldsymbol{b}_{\alpha+1} \rangle| + \delta^2 |\langle \boldsymbol{b}_{m+1} \rangle| \\
&= O\left(B(1+\delta\kappa)\right).
\end{aligned}$$

Otherwise, we have

$$\begin{aligned}
|\langle \boldsymbol{b}_{m+1}, \boldsymbol{x}_{\alpha,[t+1]}^{(t)}, \boldsymbol{x}_{\beta,[t+1]}^{(t)} \rangle| &\le |\langle \boldsymbol{b}_{m+1}, \boldsymbol{b}_{\alpha+1}, \boldsymbol{b}_{\beta+1} \rangle| + 2\delta |\langle \boldsymbol{b}_{m+1}, \boldsymbol{b}_{\alpha+1} \rangle| + \delta^2 |\langle \boldsymbol{b}_{m+1} \rangle| \\
&\le B(1 + 2\delta + \delta^2)\kappa \\
&= O(B\kappa).
\end{aligned}$$

Since $n \le n_{t+1} \le 2n$, put these together, we get

$$\frac{1}{B}\sum_{\alpha,\beta\in[n_{t+1}-1]}\langle \boldsymbol{b}_{m+1}, \boldsymbol{x}^{(t)}_{\alpha,[t+1]}, \boldsymbol{x}^{(t)}_{\beta,[t+1]}\rangle = 2 + O\left(\exp(-Cn^{\epsilon/16})\kappa\right) + O(n^2\kappa)$$
$$= 2 + O(n^2\kappa).$$

Similarly, the contraction $\langle \boldsymbol{b}_{m+1}, \boldsymbol{x}^{(t)}_{\alpha,[t+1]}, \boldsymbol{x}^{(t)}_{j,[t+1]}\rangle$ is nontrivial only when $p[j+1] = m+1$ and $\alpha+1$ is the other child node of b_{m+1}. As a result,

$$\frac{1}{B}\sum_{\alpha\in[n_{t+1}-1]}\langle \boldsymbol{b}_{m+1}, \boldsymbol{x}^{(t)}_{\alpha,[t+1]}, \boldsymbol{x}^{(t)}_{j,[t+1]}\rangle = \begin{cases} 1+O(n\kappa) & p[j+1]=m+1 \\ O(n\kappa) & o.w. \end{cases}.$$

Recall that $\kappa = O(n^{-1-\epsilon/4})$. Combinig these, we get

$$\begin{aligned}
&-\frac{1}{(n_{t+1}-1)B}\langle \boldsymbol{P}_{t+1}^\top \boldsymbol{b}_{m+1}, 2c\hat{\boldsymbol{z}}^{(t+1)}_m, \boldsymbol{x}^{(t)}_j - \hat{\boldsymbol{z}}^{(t+1)}_m\rangle \\
&= -\frac{2c}{(n_{t+1}-1)^2}(\mathbb{1}(p[j+1]=m+1) + O(n\kappa)) + \frac{2c}{(n_{t+1}-1)^3}(2+O(n^2\kappa)) \\
&= -\frac{2c}{(n_{t+1}-1)^2}\mathbb{1}(p[j+1]=m+1) + O(n^{-2-\epsilon/4}).
\end{aligned}$$

Next for Equation (10), we expand

$$\begin{aligned}
&\frac{1}{B}\langle \boldsymbol{P}_{t+1}^\top \boldsymbol{P}_{t+1}(-\mathbf{1}_d + c(\hat{\boldsymbol{z}}^{(t+1)}_m)^2), 2c\hat{\boldsymbol{z}}^{(t+1)}_m, \boldsymbol{x}^{(t)}_j - \hat{\boldsymbol{z}}^{(t+1)}_m\rangle \\
=& -\frac{2c}{B}\langle \hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]}, \boldsymbol{x}^{(t)}_{j,[t+1]}\rangle + \frac{2c}{B}\langle (\hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]})^2\rangle + \frac{2c^2}{B}\langle (\hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]})^3, \boldsymbol{x}^{(t)}_{j,[t+1]}\rangle - \frac{2c^2}{B}\langle (\hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]})^4\rangle.
\end{aligned}$$

For the second-order terms, we have:

$$\begin{aligned}
&\frac{1}{B}\langle \hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]}, \boldsymbol{x}^{(t)}_{j,[t+1]}\rangle \\
&= \frac{1}{(n_{t+1}-1)B}\left(\langle \boldsymbol{x}^{(t)}_{j,[t+1]}, \boldsymbol{x}^{(t)}_{j,[t+1]}\rangle + \sum_{\alpha\neq j}\langle \boldsymbol{x}^{(t)}_{\alpha,[t+1]}, \boldsymbol{x}^{(t)}_{j,[t+1]}\rangle\right) \\
&= \frac{1}{(n_{t+1}-1)B}\left(\langle \bar{\boldsymbol{x}}^{(t)}_{j,[t+1]}, \bar{\boldsymbol{x}}^{(t)}_{j,[t+1]}\rangle + \sum_{\alpha\neq j}\langle \bar{\boldsymbol{x}}^{(t)}_{\alpha,[t+1]}, \bar{\boldsymbol{x}}^{(t)}_{j,[t+1]}\rangle\right) + \frac{n_{t+1}-n-1}{n_{t+1}-1}(\delta\kappa+\delta^2) \\
&= \frac{1}{n_{t+1}-1}(1+(n_{t+1}-2)\kappa) + \frac{O(k)}{n_{t+1}-1}(\delta\kappa+\delta^2) \\
&= \frac{1}{n_{t+1}-1} + \frac{O(k)}{n_{t+1}-1}\delta^2 + O(\kappa).
\end{aligned}$$

Since $\delta = \exp(-Cn^{\epsilon/16})$ and $k \le n$, we observe that

$$\frac{O(k)}{n_{t+1}-1}\delta^2 \le \exp(-O(n^{\epsilon/16})),$$

which is negligible compared to $\kappa = O(n^{-1-\epsilon/4})$. Consequently, we get

$$\frac{1}{B}\langle \hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]}, \boldsymbol{x}^{(t)}_{j,[t+1]}\rangle = \frac{1}{n_{t+1}-1} + O(\kappa).$$

Similarly

$$\begin{aligned}\frac{1}{B}\langle(\hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]})^2\rangle &= \frac{1}{(n_{t+1}-1)^2B}\left(\sum_{\alpha}\langle \boldsymbol{x}^{(t)}_{\alpha,[t+1]}, \boldsymbol{x}^{(t)}_{\alpha,[t+1]}\rangle + \sum_{\alpha\neq\beta}\langle \boldsymbol{x}^{(t)}_{\alpha,[t+1]}, \boldsymbol{x}^{(t)}_{\beta,[t+1]}\rangle\right)\\ &= \frac{O(k)}{n_{t+1}-1}\delta\kappa + \frac{O(k^2)}{(n_{t+1}-1)^2}\delta^2\\ &\quad + \frac{1}{(n_{t+1}-1)^2B}\left(\sum_{\alpha}\langle \bar{\boldsymbol{x}}^{(t)}_{\alpha,[t+1]}, \bar{\boldsymbol{x}}^{(t)}_{\alpha,[t+1]}\rangle + \sum_{\alpha\neq\beta}\langle \bar{\boldsymbol{x}}^{(t)}_{\alpha,[t+1]}, \bar{\boldsymbol{x}}^{(t)}_{\beta,[t+1]}\rangle\right)\\ &= \frac{1}{n_{t+1}-1} + O(\kappa).\end{aligned}$$

For the fourth-order interaction terms, we discuss when $(\alpha,\beta,\gamma,\delta)\notin I_{4,m}$. Without loss of generality, assume $h[\alpha]\le h[\beta]\le h[\gamma]\le h[\delta]$.

1. If $h[\beta]<h[\gamma]<h[\delta]$, to cancel out b_δ, it must hold that b_γ is a child of b_δ and b_α, b_β are the two children of the other child. This pattern is fully determined by choosing δ and which child is γ. Thus there are only $O(n)$ such trivial 4-tuples.
2. If $h[\beta]=h[\gamma]<h[\delta]$, it must also hold that $h[\gamma]=h[\delta]-1$. Then b_β and b_γ must both be the children nodes of b_δ. Then we have either $b_\beta=b_\gamma$ or $b_\delta=b_\beta b_\gamma$, where both cases cannot be trivial.
3. If $h[\beta]<h[\gamma]=h[\delta]$, we have $\gamma=\delta$ and $\alpha=\beta$ such that $b_\gamma b_\delta\equiv 1$ and$b_\alpha b_\beta\equiv 1$. There are $O(n^2)$ different choices.
4. If $h[\beta]=h[\gamma]=h[\delta]$, again we have two of the indices to be the same and so are the remaining two. Thus there are $O(n^2)$ trivial 4-tuples.

$$\begin{aligned}\frac{1}{B}\langle(\hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]})^4\rangle &= \frac{1}{(n_{t+1}-1)^4B}\sum_{\alpha,\beta,\gamma,\delta}\langle \boldsymbol{x}^{(t)}_{\alpha,[t+1]}, \boldsymbol{x}^{(t)}_{\beta,[t+1]}, \boldsymbol{x}^{(t)}_{\gamma,[t+1]}, \boldsymbol{x}^{(t)}_{\delta,[t+1]}\rangle\\ &= \frac{1}{(n_{t+1}-1)^4B}\sum_{\alpha,\beta,\gamma,\delta}\langle \bar{\boldsymbol{x}}^{(t)}_{\alpha,[t+1]}, \bar{\boldsymbol{x}}^{(t)}_{\beta,[t+1]}, \bar{\boldsymbol{x}}^{(t)}_{\gamma,[t+1]}, \bar{\boldsymbol{x}}^{(t)}_{\delta,[t+1]}\rangle\\ &\quad + \sum_{j=1}^{3}\frac{O(k^j)}{(n_{t+1}-1)^j}\delta^j\kappa + \frac{O(k^4)}{(n_{t+1}-1)^4}\delta^4\\ &\le \frac{1}{(n_{t+1}-1)^4B}\left(\sum_{\alpha,\beta,\gamma,\delta\in I_{4,m}}O(B\kappa) + \sum_{\alpha,\beta,\gamma,\delta\notin I_{4,m}}B\right) + O(\delta\kappa)\\ &\le \frac{1}{(n_{t+1}-1)^4}(O(n^2)+(n_{t+1}-1)^4\kappa) + O(\delta\kappa)\\ &= O(n^{-2}+\kappa).\end{aligned}$$

Now for $\langle(\hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]})^3, \boldsymbol{x}^{(t)}_{j,[t+1]}\rangle$, assuming index j is contained in $(\alpha,\beta,\gamma,\delta)$. Then for case 1 we only have $O(1)$ trivial tuples and both case 3 and 4 are reduced to $O(n)$ as there's only one free index to be determined.

$$\begin{aligned}&\frac{1}{B}\langle(\hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]})^3, \boldsymbol{x}^{(t)}_{j,[t+1]}\rangle\\ &\le \frac{1}{(n_{t+1}-1)^3B}\sum_{\alpha,\beta,\gamma}\langle \bar{\boldsymbol{x}}^{(t)}_{\alpha,[t+1]}, \bar{\boldsymbol{x}}^{(t)}_{\beta,[t+1]}, \bar{\boldsymbol{x}}^{(t)}_{\gamma,[t+1]}, \bar{\boldsymbol{x}}^{(t)}_{j,[t+1]}\rangle + O(n^{-1-\epsilon/4})\\ &\le \frac{1}{(n_{t+1}-1)^3}(O(n)+(n_{t+1}-1)^3\kappa) + O(\delta\kappa)\\ &= O(n^{-2}+\kappa).\end{aligned}$$

Thus

$$\begin{aligned}
&\frac{1}{(n_{t+1}-1)B}\langle \boldsymbol{P}_{t+1}^\top \boldsymbol{P}_{t+1}(-\mathbf{1}_d + c(\hat{\boldsymbol{z}}_m^{(t+1)})^2), 2c\hat{\boldsymbol{z}}_m^{(t+1)}, \boldsymbol{x}_j^{(t)} - \hat{\boldsymbol{z}}_m^{(t+1)}\rangle \\
&= -\frac{2c}{n_{t+1}-1} + \frac{2c}{n_{t+1}-1} + \frac{O(\kappa)}{n_{t+1}-1} \\
&= O(n^{-2-\epsilon/4}).
\end{aligned}$$

For Eq. (11), note that

$$\frac{1}{B}\langle \boldsymbol{P}_{t+1}^\top \boldsymbol{P}_{t+1}|\hat{\boldsymbol{z}}_m^{(t+1)}|^4\rangle = \frac{1}{B}\langle |\hat{\boldsymbol{z}}_{m,[t+1]}^{(t+1)}|^4\rangle = \frac{1}{B}\langle (\hat{\boldsymbol{z}}_{m,[t+1]}^{(t+1)})^4\rangle = O(n^{-2}+\kappa).$$

and since $2c\hat{\boldsymbol{z}}_m^{(t+1)}, \boldsymbol{x}_j^{(t)} - \hat{\boldsymbol{z}}_m^{(t+1)}$ are contained in $[-1,1]$ and $[-2,2]$ respectively. We have

$$\frac{1}{(n_{t+1}-1)B}\langle \boldsymbol{P}_{t+1}^\top \boldsymbol{P}_{t+1}|\hat{\boldsymbol{z}}_m^{(t+1)}|^4, 2c\hat{\boldsymbol{z}}_m^{(t+1)}, \boldsymbol{x}_j^{(t)} - \hat{\boldsymbol{z}}_m^{(t+1)}\rangle = \frac{4cO(n^{-1-\epsilon/8})}{n_{t+1}-1} = O(n^{-2-\epsilon/4}).$$

Finally for Eq. (12), using Cauchy-Schwarz we have

$$\begin{aligned}
\frac{1}{B}\langle \boldsymbol{P}_{t+1}^\top \boldsymbol{P}_{t+1}|\hat{\boldsymbol{z}}_m^{(t+1)}|^3\rangle = \frac{1}{B}\langle |(\hat{\boldsymbol{z}}_{m,[t+1]}^{(t+1)})^3|\rangle &= \frac{1}{B}\sum_{i=1}^{B}|(\hat{\boldsymbol{z}}_{m,[t+1],i}^{(t+1)})^3| \\
&\le \frac{1}{B}\left(\sum_{i=1}^{B}(\hat{\boldsymbol{z}}_{m,[t+1],i}^{(t+1)})^2\right)^{1/2}\left(\sum_{i=1}^{B}(\hat{\boldsymbol{z}}_{m,[t+1],i}^{(t+1)})^4\right)^{1/2} \\
&= \frac{1}{B}O(Bn^{-1})^{1/2}O(Bn^{-1-\epsilon/4})^{1/2} \\
&= O(n^{-1-\epsilon/8}).
\end{aligned}$$

By the definition of $\boldsymbol{P}_t$, we get

$$\begin{aligned}
&\frac{1}{(n_{t+1}-1)B}\langle \boldsymbol{P}_{t+1}^\top(\boldsymbol{P}_{t+1}\phi(\hat{\boldsymbol{z}}_m^{(t+1)}) - \boldsymbol{b}_{m+1}), O(|\hat{\boldsymbol{z}}_m^{(t+1)}|^3), \boldsymbol{x}_j^{(t)} - \hat{\boldsymbol{z}}_m^{(t+1)}\rangle \\
&= \frac{1}{(n_{t+1}-1)B}\langle \phi(\hat{\boldsymbol{z}}_m^{(t+1)}) - \boldsymbol{P}_{t+1}^\top \boldsymbol{b}_{m+1}, O(\boldsymbol{P}_{t+1}^\top \boldsymbol{P}_{t+1}|\hat{\boldsymbol{z}}_m^{(t+1)}|^3), \boldsymbol{x}_j^{(t)} - \hat{\boldsymbol{z}}_m^{(t+1)}\rangle \\
&= \frac{4}{(n_{t+1}-1)B}O\left(\langle \boldsymbol{P}_{t+1}^\top \boldsymbol{P}_{t+1}|\hat{\boldsymbol{z}}_m^{(t+1)}|^3\rangle\right) \\
&= O(n^{-2-\epsilon/8}).
\end{aligned}$$

Combining all terms from (9) to (12), for any $j \le n_{t+1}-1$, we gets

$$\frac{\partial \mathcal{L}^{(t+1)}}{\partial w_{j,m}^{(t+1)}} = -\frac{2c}{(n_{t+1}-1)^2}\mathbb{1}(p[j+1] = m+1) + O(n^{-2-\epsilon/8})$$

We can absorb the oracle noise to the second term as in Lemma 3.

Concentration of attention scores. Taking $\eta = n^{2+\epsilon/16}\eta_0$, the updated weight becomes

$$w_{j,m}^{(t+1)}(t+1) = -\eta\tilde{\nabla}_{w_{j,m}^{(t+1)}}\mathcal{L}^{(t+1)} = 2c\eta_0\frac{n^{2+\epsilon/16}}{(n_{t+1}-1)^2}\mathbb{1}(p[j+1] = m+1) + O(n^{-\epsilon/16})$$

Similar to Lemma 3, we choose η_0 such that the leading terms are not half-integers. For any $j \notin \{c_1[m+1], c_2[m+1]\}$, we have

$$w_{j,m}^{(t+1)} = q\left(O(n^{-\epsilon/16})\right) = 0.$$

Conversely, for the indices corresponding to the signal, the coefficient is bounded as $n \le n_{t+1} < 2n$

$$\frac{c\eta_0}{2} \le 2c\eta_0\frac{n^2}{(n_{t+1}-1)^2} \le 2c\eta_0.$$

Let $c_t = 2c\eta_0 \frac{n^2}{(n_{t+1}-1)^2} = \Theta(1)$. The weight concentrates as

$$w^{(t+1)}_{c_1[m+1],m} = w^{(t+1)}_{c_2[m+1],m} = q\left(c_t n^{\epsilon/16}\right).$$

The softmax scores for the noise terms can thus be upper bounded by

$$\sigma_j(\boldsymbol{w}_m^{(t+1)}) \le \frac{1}{\exp\left(w^{(t+1)}_{c_1[m+1],m}\right) + \exp\left(w^{(t+1)}_{c_2[m+1],m}\right) + n - 2} \le \exp(-Cn^{\epsilon/16}).$$

Since the softmax scores must sum to 1, it holds that

$$\frac{1 - n\exp(-Cn^{\epsilon/16})}{2} \le \sigma_{c_1[m+1]}(\boldsymbol{w}_m^{(t+1)}), \sigma_{c_2[m+1]}(\boldsymbol{w}_m^{(t+1)}) \le \frac{1}{2}.$$

Evaluating the forward pass. To bound the prediction loss, we define the following increasing sequence

$$\epsilon_t = \max_{1\le\tau\le t} \max_{m\in[N_\tau+1, N_{\tau+1}]} \|\boldsymbol{x}^{(\tau)}_{m,[\tau+1]} - \boldsymbol{b}_{m+1}\|_\infty, \epsilon_0 = 0.$$

Thus

$$\|\boldsymbol{x}^{(t)}_{c_1[m+1],[t+1]} - \boldsymbol{b}_{c_1[m+1]+1}\|_\infty, \|\boldsymbol{x}^{(t)}_{c_2[m+1],[t+1]} - \boldsymbol{b}_{c_2[m+1]+1}\|_\infty \le \epsilon_t,$$

and for the intermediate state $\hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]}$ we have

$$\begin{aligned}
&\left\|\hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]} - \frac{\boldsymbol{b}_{c_1[m+1]} + \boldsymbol{b}_{c_2[m+1]}}{2}\right\|_\infty \\
&\le \left\|\hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]} - \frac{\boldsymbol{x}^{(t)}_{c_1[m+1]-1,[t+1]} + \boldsymbol{x}^{(t)}_{c_2[m+1]-1,[t+1]}}{2}\right\|_\infty + \epsilon_t \\
&\le \sum_{p[j+1]\neq m+1} \sigma_j(\boldsymbol{w}_m^{(t+1)}) + \left|\sigma_{c_1[m+1]-1}(\boldsymbol{w}_m^{(t+1)}) - \frac{1}{2}\right| + \left|\sigma_{c_2[m+1]-1}(\boldsymbol{w}_m^{(t+1)}) - \frac{1}{2}\right| + \epsilon_t \\
&\le 2(n-1)\exp(-Cn^{\epsilon/16}) + \epsilon_t. \qquad (13)
\end{aligned}$$

We first use this characterization to bound the connection weights, for $n_{t+1} \le m \le n_{t+2} - 1$, we have $\Psi_t^\top \hat{\boldsymbol{z}}_m^{(t+1)} = \hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]} = \frac{1}{2}(\boldsymbol{x}^{(t)}_{c_1[m+1]-1,[t+1]} + \boldsymbol{x}^{(t)}_{c_2[m+1]-1,[t+1]}) + \exp(-Cn^{\epsilon/16})$, which passes the filter. thus we have $\boldsymbol{g}_m^{(t+1)} = 1$ on this range. In contrast, for $n \le m \le n_{t+1} - 1$, the weights $w^{(t+1)}_{j,m}$ remains zero, so the attention is uniform and the corresponding states collapse to $-\mathbf{1}_B$ by Lemma 1. Moreover, by construction $g_m^{(t+1)} = 0$ for $m \le n-1$. Putting these cases together, we obtain

$$g_m^{(t+1)} = \begin{cases} 0 & m \le n_{t+1} - 1 \\ 1 & n_{t+1} \le m \le n_{t+2} - 1 \end{cases}.$$

Next, we bound ϵ_t . Using the Taylor expansion of ϕ,

$$\begin{aligned}
\epsilon_{t+1} = \|\boldsymbol{x}^{(t+1)}_{m,[t+2]} - \boldsymbol{b}_{m+1}\|_\infty &= \left\|\phi(\hat{\boldsymbol{z}}^{(t+1)}_{m,[t+1]}) - \phi\left(\frac{\boldsymbol{x}^{(t)}_{c_1[m+1],[t+1]} + \boldsymbol{x}^{(t)}_{c_2[m+1],[t+1]}}{2}\right)\right\|_\infty \\
&\le C_2(2(n-1)\exp(-Cn^{\epsilon/16}) + \epsilon_t)^2,
\end{aligned}$$

for some constant C_2 depending on ϕ only. We can inductively show that $\epsilon_t \le \delta = \exp(-Cn^{\epsilon/16})$. When $t = 1$, this is true by Lemma 3. Assume $\epsilon_t \le \delta$, then

$$\epsilon_{t+1} \le C_2(2(n-1)\delta + \delta)^2 \le 4C_2 n^2 \delta^2.$$

To conclude $\epsilon_{t+1} \le \delta$, it suffice to have:

$$4C_2 n^2 \delta^2 \le \delta \;\Leftrightarrow\; 4C_2 n^2 \delta \le 1.$$

Since $\delta = \exp(-Cn^{\epsilon/16})$ decays faster than any polynomial, for any fixed constant C_2, there exists n_0 such that for all $n \ge n_0$,

$$4C_2 n^2 \delta \le 1.$$

Thus we have $\epsilon_t \le \delta$ inductively for all $t \le \log_2 k$. We can conclude that

$$\|\hat{f}(\boldsymbol{D}_{\text{test}}) - \boldsymbol{y}_{\text{test}}\|_\infty \le \exp(-Cn^{\epsilon/16}).$$

□

A.3 Gradient Upper Bound for Trained Layers

Lemma 6. *Suppose the trainig is in stage $t+1$ and for all $\ell' \le t$, $n_{\ell'} \le m \le n_{\ell'+1} - 1$, we have*

$$\frac{1 - n\exp(-Cn^{\epsilon/16})}{2} \le \sigma_{c_1[m+1]}(\boldsymbol{w}_m^{(\ell')}), \sigma_{c_2[m+1]}(\boldsymbol{w}_m^{(\ell')}) \le \frac{1}{2},$$

and for $\ell' \ge t+1$, $\|\boldsymbol{W}_{\mathsf{KQ}}^{(\ell')}\|_2 = 0$. Then for any $\ell' \le t$, we have

$$\left|\frac{\partial \mathcal{L}^{(t+1)}}{\partial w_{j,m}^{(\ell')}}\right| \le O(\exp(-Cn^{\epsilon/16})).$$

Consequently, with step size $\eta = n^{2+\epsilon/16}\eta_0$, the weights in all previously trained layers $\ell' \le t$ remain unchanged

$$w_{j,m}^{(\ell')}(t+1) = w_{j,m}^{(\ell')}(t).$$

Proof. We define the gradient error:

$$\zeta_{j,m,\ell'}^{(\ell)} := \max_{1 \le \alpha \le n_{\ell+1}-1} \left\| \frac{\partial \phi(\hat{\boldsymbol{z}}_{\alpha,[\ell]}^{(\ell)})}{\partial w_{j,m}^{(\ell')}} \right\|_\infty.$$

We will prove by induction that for all $\ell' \le \ell \le t$, $n_{\ell'} \le m \le n_{\ell'+1} - 1$ and $1 \le j \le n_{\ell'} - 1$.

$$\zeta_{j,m,\ell'}^{(\ell)} \le O(\exp(-Cn^{\epsilon/16})).$$

Recall that for all $n_{\ell'} \le m \le n_{\ell'+1} - 1$, we have

$$\left\| \hat{\boldsymbol{z}}_{m,[\ell']}^{(\ell')} - \frac{\boldsymbol{b}_{c_1[m+1]} + \boldsymbol{b}_{c_2[m+1]}}{2} \right\|_\infty \le 2n\exp(-Cn^{\epsilon/16}).$$

Since $\boldsymbol{b}_{c_1[m+1]}, \boldsymbol{b}_{c_2[m+1]} \in \{-1, 1\}$, their average lies in $\{-1, 0, 1\}$. We consider two cases. If $\boldsymbol{b}_{c_1[m+1]} = \boldsymbol{b}_{c_2[m+1]}$, we can consider taylor expansion of ϕ' around ± 1 as $\phi'(\pm 1) = \phi'(0) = 0$ respectively: $\phi'(t) = 2c'(1-t) + O((1-t)^2)$ around 1 and $\phi'(t) = 2c'(-1-t) + O((1+t)^2)$ near -1 for some constant $c' > 0$. Then

$$\|\phi'(\hat{\boldsymbol{z}}_{m,[\ell']}^{(\ell')})\|_\infty = O(\exp(-C'n^{\epsilon/16})),$$

for some constant C'. Similarly, when $\boldsymbol{b}_{c_1[m+1]} = -\boldsymbol{b}_{c_2[m+1]}$, we use the taylor expansion around 0, giving

$$\|\phi'(\hat{\boldsymbol{z}}_{m,[\ell']}^{(\ell')})\|_\infty = O(\exp(-C'n^{\epsilon/16})).$$

Put together, we have

$$\begin{aligned}
\left\| \frac{\partial \phi(\hat{\boldsymbol{z}}_{\alpha,[\ell']}^{(\ell')})}{\partial w_{j,m}^{(\ell')}} \right\|_\infty &\le \|\mathrm{diag}(\phi'(\hat{\boldsymbol{z}}_{m,[\ell']}^{(\ell')}))(\boldsymbol{x}_{j,[\ell']}^{(\ell'-1)} - \hat{\boldsymbol{z}}_{m,[\ell']}^{(\ell')})\|_\infty \sigma_j(w_m^{(\ell')}) \\
&\le 2\|\phi'(\hat{\boldsymbol{z}}_{m,[\ell']}^{(\ell')})\|_\infty \sigma_j(w_m^{(\ell')}) \\
&\le O(\exp(-C'n^{\epsilon/16})).
\end{aligned}$$

Now for any $\ell' < \ell \le t$, we have

$$\begin{aligned}
\left\| \frac{\partial \phi(\hat{\boldsymbol{z}}_{\alpha,[\ell]}^{(\ell)})}{\partial w_{j,m}^{(\ell')}} \right\|_\infty &\le \|\phi'(\hat{\boldsymbol{z}}_{m,[\ell]}^{(\ell)})\|_\infty \sum_{\alpha=1}^{n_\ell - 1} \sigma_\alpha(w_m^{(\ell)}) \left\| \frac{\partial \boldsymbol{x}_{\alpha,[\ell]}^{(\ell-1)}}{\partial w_{j,m}^{(\ell')}} \right\|_\infty \\
&\le O(\exp(-Cn^{\epsilon/16})) \max_{\alpha \in [n_\ell - 1]} \left\| \frac{\partial \boldsymbol{x}_{\alpha,[\ell]}^{(\ell-1)}}{\partial w_{j,m}^{(\ell')}} \right\|_\infty \\
&\overset{(a)}{\le} O(\exp(-Cn^{\epsilon/16}))\zeta_{j,m,\ell'}^{(\ell-1)},
\end{aligned}$$

where (a) follows the fact that for $\alpha \in [n_\ell - 1]$, we have $\boldsymbol{g}_\alpha^{(\ell-1)} = 1$ that leads to $\boldsymbol{x}_{\alpha,[\ell]}^{(\ell-1)} = \phi(\hat{\boldsymbol{z}}_{\alpha,[\ell-1]}^{(\ell-1)})$. Thus, we can inductively show that $\zeta_{j,m,\ell'}^{(t)} = \zeta_{j,m,\ell'}^{(t-1)} = \cdots = \zeta_{j,m,\ell'}^{(\ell')} \leq O(\exp(-Cn^{\epsilon/16}))$. Now when $\ell = t+1$, we have

$$\begin{aligned}
\left\| \frac{\partial \phi(\hat{\boldsymbol{z}}_{\alpha,[t+1]}^{(t+1)})}{\partial w_{j,m}^{(\ell')}} \right\|_\infty &\leq \|\phi'(\hat{\boldsymbol{z}}_{m,[t+1]}^{(t+1)})\|_\infty \sum_{\alpha=1}^{n_{t+1}-1} \sigma_\alpha(w_m^{(t+1)}) \left\| \frac{\partial \boldsymbol{x}_\alpha^{(t)}}{\partial w_{j,m}^{(\ell')}} \right\|_\infty \\
&\overset{(a)}{\leq} O(\|\phi'\|_\infty n) \zeta_{j,m,\ell'}^{(t)} \\
&\leq O(\exp(-Cn^{\epsilon/16})),
\end{aligned}$$

where $\|\phi'\|_\infty$ is the Lipschitz constant for ϕ on $[-1, 1]$. Consequently, for any $\ell' \leq t$, we get

$$\begin{aligned}
\left| \frac{\partial \mathcal{L}^{(t+1)}}{\partial w_{j,m}^{(\ell')}} \right| &= \left| \sum_{\alpha=n_{t+1}}^{n_{t+2}-1} \frac{1}{B} \left(\boldsymbol{P}_{t+1}^\top \left(\boldsymbol{P}_{t+1} \phi(\hat{\boldsymbol{z}}_\alpha) - \boldsymbol{b}_{\alpha+1} \right) \right)^\top \frac{\partial \phi(\hat{\boldsymbol{z}}_\alpha^{(t+1)})}{\partial w_{j,m}^{(\ell')}} \right| \\
&\overset{(a)}{=} \left| \sum_{\alpha=n_{\ell+1}}^{n_{\ell+2}-1} \frac{1}{B} \left(\boldsymbol{P}_{t+1} \phi(\hat{\boldsymbol{z}}_\alpha) - \boldsymbol{b}_{\alpha+1} \right)^\top \frac{\partial \left(\phi(\hat{\boldsymbol{z}}_{\alpha,[t+1]}^{(t+1)}) \right)}{\partial w_{j,m}^{(\ell')}} \right| \\
&\overset{(b)}{\leq} \sum_{\alpha=n_{\ell+1}}^{n_{\ell+2}-1} \frac{1}{B} \left\| \boldsymbol{P}_{t+1} \phi(\hat{\boldsymbol{z}}_\alpha) - \boldsymbol{b}_{\alpha+1} \right\|_1 \left\| \frac{\partial \phi(\hat{\boldsymbol{z}}_{\alpha,[t+1]}^{(t+1)})}{\partial w_{j,m}^{(\ell')}} \right\|_\infty \\
&\overset{(c)}{\leq} 2 \sum_{\alpha=n_{\ell+1}}^{n_{\ell+2}-1} \left\| \frac{\partial \phi(\hat{\boldsymbol{z}}_{\alpha,[t+1]}^{(t+1)})}{\partial w_{j,m}^{(\ell')}} \right\|_\infty \\
&\leq O(\exp(-Cn^{\epsilon/16})),
\end{aligned}$$

where (a) follows the definition of $\boldsymbol{P}_\ell$ as a block-selection matrix, (b) applies Holder's inequality and (c) uses the fact that $\boldsymbol{P}_{t+1}\phi(\hat{\boldsymbol{z}}_\alpha), \boldsymbol{b}_{\alpha+1} \in [-1, 1]^B$. As a result, with $\eta = n^{2+\epsilon/16}\eta_0$, the weights in layer $\ell' \leq t$ stays the same

$$w_{j,m}^{(\ell')}(t+1) = q\left(w_{j,m}^{(\ell')}(t) - \eta O(\exp(-Cn^{\epsilon/16})) \right) = w_{j,m}^{(\ell')}(t)$$

□