How Transformers Implement Induction Heads: Approximation and Optimization Analysis

Anonymous authors
Paper under double-blind review

Abstract

Transformers have demonstrated exceptional in-context learning capabilities, yet the theoretical understanding of the underlying mechanisms remains limited. A recent work (Elhage et al., 2021) identified a "rich" in-context mechanism known as induction head, contrasting with "lazy" n-gram models that overlook long-range dependencies. In this work, we provide both approximation and dynamics analyses of how transformers implement induction heads. In the approximation analysis, we formalize both standard and generalized induction head mechanisms, and examine how transformers can efficiently implement them, with an emphasis on the distinct role of each transformer submodule. For the dynamics analysis, we study the training dynamics on a synthetic mixed target, composed of a 4-gram and an in-context 2-gram component. This controlled setting allows us to precisely characterize the entire training process and uncover an *abrupt transition* from lazy (4-gram) to rich (induction head) mechanisms as training progresses.

1 Introduction

Transformer, introduced by Vaswani et al. (2017), have achieved remarkable success across various domains, including natural language processing, computer vision, and scientific computing. An emergent observation is that transformers, trained on trillions of tokens, can perform (few-shot) in-context learning (ICL), which makes prediction based on the contextual information without needing model retraining (Brown et al., 2020). This ICL ability is widely regarded as crucial for enabling large language models (LLMs) to solve reasoning tasks, representing a key step toward more advanced artificial intelligence.

To understand how transformers implement ICL, Elhage et al. (2021) and Olsson et al. (2022) identified a simple yet powerful mechanism known as **induction head**. Specifically, given an input sequence $[\cdots \texttt{ab} \cdots \texttt{a}]$, an induction head predicts $\texttt{b}$ as the next token by leveraging the prior occurrence of the pattern $\texttt{ab}$ in the context, effectively modeling an in-context bi-gram. In contrast, traditional n-gram model (Shannon, 1948) (with a small n) utilizes only a limited number of recent tokens to predict the next token, which is context-independent and inevitably overlooks long-range dependence. Based on the extent of context utilization, we categorize n-gram model as a *"lazy" mechanism*, whereas the induction head represents a more *"rich" mechanism*.

Practically, induction heads have been demonstrated to play a critical role in enabling LLMs' ICL capabilities (Song et al., 2024; Crosbie and Shutova, 2024), and even used to test new LLM architectures (Gu and Dao, 2023). Theoretically, induction heads also serve as a controllable tool for understanding various aspects of LLMs, such as multi-step reasoning (Sanford et al., 2024b) and inductive biases of different architectures (Jelassi et al., 2024).

In this paper, we aim to provide a theoretical analysis of how transformers can efficiently implement induction heads. The first key problem is to rigorously formalize induction heads and evaluate the efficiency of transformers in representing them. According to Elhage et al. (2021), the original induction head can be implemented using a two-layer, twelve-head transformer without feed-forward networks (FFNs). However, practical scenarios demand more powerful induction heads. Thus, it is crucial to generalize the mechanism behind and explore how different transformer submodules, such as varying the number of attention heads or incorporating FFNs, impact the transformer's ability to implement them. This forms our first research objective:

(Approximation). Investigate how two-layer transformers express the induction head mechanism and its potential variants.

The next problem is to investigate the dynamics of transformers in learning induction heads. The pioneering works by Elhage et al. (2021) and Olsson et al. (2022) demonstrated that transformers undergo an abrupt phase transition to learning induction heads. A recent empirical study on synthetic datasets replicate this behavior, further showing that 2-gram is always learned prior to induction heads (Bietti et al., 2024). However, a rigorous theoretical analysis of this learning progression is still lacking. Closing this gap forms our second research objective:

(Optimization). Understand how transformers transition from relying on n-gram patterns to employing the induction head mechanism as training progresses.

Focusing on these two key problems, in this paper, we make the following contributions:

- **Approximation analysis: how transformers express induction heads.** We consider three types of induction heads with varying complexities. First, we show that two-layer, single-head transformers without FFNs can efficiently approximate the vanilla induction head (Elhage et al., 2021). We then introduce two generalized induction heads, which leverage richer in-context n-gram information and incorporate a general similarity function. Our analysis clarifies the distinct roles of multihead attention, positional encoding, dot-product structure, and FFNs in implementing these generalized induction heads.
- **Optimization analysis: how learning undergoes a sharp transition from n-gram to induction head.** We study the learning dynamics of a two-layer transformer without FFNs for a mixed target, composed of a 4-gram and an in-context 2-gram component. This toy setting allows us to capture the entire training process precisely. Specifically, we show that learning progresses through four phases: partial learning of the 4-gram, plateau of induction head learning, emergence of the induction head, and final convergence, showcasing a sharp transition from 4-gram to induction head. Our analysis identifies two key drivers of the transition: 1) time-scale separation due to low- and high-order parameter dependencies in self-attention, and 2) speed differences caused by the relative proportions of the two components in the mixed target.

2 Related Works

Empirical observations of induction head. The induction head mechanism was first identified by Elhage et al. (2021) in studying how two-layer transformers perform language modeling. Subsequently, Olsson et al. (2022) conducted a more systematic investigation, revealing two key findings: 1) induction head emerges abruptly during training, and 2) induction head plays a critical role in the development of in-context learning capabilities. To obtain a fine-grained understanding of how induction head emerges during training, recent studies have developed several synthetic settings (Reddy, 2024; Edelman et al., 2024; Bietti et al., 2024). Particularly, Bietti et al. (2024) successfully reproduced the fast learning of (global) bigrams and the slower development of induction head. Despite these efforts, a comprehensive theoretical understanding of how the induction head operates in two-layer transformers and how it is learned during training remains elusive.

Expressiveness of transformers. Theoretically, Dehghani et al. (2019); Pérez et al. (2021); Wei et al. (2022) explored the Turing-completeness of transformers; Yun et al. (2019) established the universal approximation property of transformers. Subsequent studies examined the efficiency of transformers in representing specific functions or tasks, such as sparse functions (Edelman et al., 2022), targets with nonlinear temporal kernels (Jiang and Li, 2023), practical computer programs (Giannou et al., 2023), long but sparse memories (Wang et al., 2024), induction head (Sanford et al., 2024a;b; Rajaraman et al., 2024), and memorization and reasoning (Chen and Zou, 2024). Besides, many studies suggest that transformers achieve in-context learning by approximating gradient-based iterations across various layers (Garg et al., 2022; Akyürek et al., 2022; Von Oswald et al., 2023; Mahankali et al., 2023; Bai et al., 2023; Shen et al., 2023). Besides, several studies explored the limitation of transformer's expressivity, particularly in modeling formal languages or simulating circuits (Hahn, 2020; Weiss et al., 2021; Bhattamishra et al., 2020; Merrill et al., 2022; Merrill and Sabharwal, 2023). Among all these works, the most closely related to ours are Rajaraman et al. (2024), which examined a generalized induction head similar to our Eq. (6). Specifically, they showed that multi-

layer transformers with single-head attention can implement this mechanism. In contrast, we prove that two-layer transformers are sufficient if multihead attention is used.

Training dynamics of transformers. To gain insights into the dynamics of training transformers, several studies have analyzed simplified transformers on toy tasks. These tasks include learning distinct/common tokens (Tian et al., 2023), leaning balance/inblanced features (Huang et al., 2023), linear regression task (Zhang et al., 2023; Ahn et al., 2024), multi-task linear regression (Chen et al., 2024a), binary classification (Li et al., 2024), transformer with diagonal weights (Abbe et al., 2024), learning causal structure (Nichani et al., 2024), sparse token selection task (Wang et al., 2024), and learning n-gram Markov chain (Chen et al., 2024b). Additionally, studies such as those by Ataee Tarzanagh et al. (2023), Tarzanagh et al. (2023) and Vasudeva et al. (2024) have analyzed scenarios where transformers converge to max-margin solutions. Furthermore, Thrampoulidis (2024) has examined the implicit bias of next-token prediction. Among these works, the most closely related to ours are Nichani et al. (2024) and Chen et al. (2024b), which proved that two-layer transformers can converge to induction head solutions. In this work, we explore a setting where the target is a mixture of 4-gram and induction head. We show that two-layer transformers can effectively converge to this mixed target and provide a precise description of the learning process associated with each component. Importantly, we are able to capture the *abrupt transition* from learning 4-gram patterns to mastering the induction head mechanism—a critical phase in the learning of induction heads, as highlighted in the seminal works (Elhage et al., 2021; Olsson et al., 2022).

3 Preliminaries

Notations. For $k \in \mathbb{N}^+$, let $[k] = \{1, 2, \dots, k\}$. For a vector v and $1 \le p \le \infty$, we denote by $\|v\|_p$ the ℓ_p norm of v. For a matrix $A = (a_{i,j})$, we denote by $\|A\|$, $\|A\|_F$ the spectral and Frobenius norms, respectively; let $\|A\|_{1,1} = \sum_{i,j} |a_{i,j}|$. For an event S, we define $\mathbb{I}\{S\} = 1$ if S is true, and 0 otherwise. We use standard big-O notations $\mathcal{O}, \Omega, \Theta$ to hide absolute positive constants, and use $\tilde{\mathcal{O}}, \tilde{\Omega}, \tilde{\Theta}$ to further hide logarithmic constants.

Sequence modeling. Given a sequence of tokens $(x_1, x_2, x_3, \dots)$ with each token lying in $\mathbb{R}^d$, let $X_L = (x_1, x_2, \dots, x_L) \in \mathbb{R}^{d \times L}$ and $X_{m:n} = (x_m^\top, x_{m+1}^\top, \dots, x_n^\top)^\top \in \mathbb{R}^{(n-m+1)d}$. Given $A = (a_1, \cdots a_n) \in \mathbb{R}^{m \times n}$, we denote $(a_s)_{s=i}^j = (a_i, \cdots, a_j) \in \mathbb{R}^{m \times (j-i+1)}$. Then, we consider the next-token prediction task: predict x_{L+1} using $X_L = (x_1, x_2, \dots, x_L)$.

In a **n-gram model** (Shannon, 1948), the conditional probability of predicting the next token is given by $p(x_{L+1}|X_L) = p(x_{L+1}|X_{L-n+2:L})$, meaning that the prediction depends only on the most recent $n-1$ tokens. In practice, the value of n is typically small (e.g., 2, 3, or 4), as the computational cost of n-gram models grows exponentially with n. However, n-gram models with small n cannot capture long-range interactions, leading to inferior performance in sequence modeling.

Transformer is designed to more efficiently capture long-range dependencies in sequence modeling (Vaswani et al., 2017). Specifically, given an L-token input sequence $X = (x_1, \cdots, x_L) \in \mathbb{R}^{d \times L}$, an U-layer transformer TF processes it as follows. First, each input token is embedded into a higher-dimensional space through an *embedding layer*:

$$x_s^{(0)} = W_E x_s + b_E, \ s \in [L], \quad \text{with } W_E \in \mathbb{R}^{D \times d}, b_E \in \mathbb{R}^D.$$

Next, the U-layer attention blocks process the embedded sequence $X^{(0)} = (x_1^{(0)}, \cdots, x_L^{(0)})$ as follows, and the output of the final layer is taken as the output sequence $\mathsf{TF}(X) = X^{(L)} \in \mathbb{R}^{D \times L}$:

$$\begin{aligned} X^{(u-\frac{1}{2})} &= X^{(u-1)} + \mathsf{SA}^{(u)}(X^{(u-1)}), \quad u \in [U]; \\ X^{(u)} &= X^{(u-\frac{1}{2})} + \mathsf{FFN}^{(u)}(X^{(u-\frac{1}{2})}), \quad u \in [U]. \end{aligned} \tag{1}$$

Here, $\mathsf{FFN}^{(u)}$ denotes a (token-wise) two-layer FFN of width M, and $\mathsf{SA}^{(u)}$ represents the multi-head self-attention operation. Specifically, when applied to a sequence $Z = (z_1, \cdots, z_L) \in \mathbb{R}^{D \times L}$, $\mathsf{SA}^{(l)}$ operates it as follows:

$$\begin{aligned} \mathsf{SA}^{(u)}(Z) =& W_O^{(u)} \sum_{h=1}^{H_u} \mathsf{SA}^{(u,h)}(Z), \\ \mathsf{SA}^{(u,h)}(Z) =& \left(W_V^{(u,h)} Z\right) \mathrm{softmax}\left(\left\langle W_Q^{(u,h)} Z, W_K^{(u,h)} Z \right\rangle + R^{(u,h)}\right), \end{aligned} \tag{2}$$

where $W_Q^{(u,h)}, W_K^{(u,h)}, W_V^{(u,h)}, W_O^{(u)} \in \mathbb{R}^{D\times D}$ correspond to the query, key, value and output matrices of the (u,h)-th head, respectively. $\mathrm{softmax}$ represents taking softmax normalization across columns. $\left\langle W_Q^{(u,h)}X, W_K^{(u,h)}X \right\rangle$ is called the dot-product (DP) structure. Furthermore, $R^{(u,h)} = (R_{i,j}^{(u,h)}) \in \mathbb{R}^{L\times L}$ denotes the additive relative positional encoding matrix, which satisfies $R_{i,j}^{(u,h)} = -\infty$ if $i \le j$ for the next-token prediction task.

Relative positional encoding (RPE). Throughout this paper, we focus on the Alibi RPE (Press et al., 2022), where $R_{ij}^{(u,h)}$ follows a Toeplitz structure, i.e., $R_{ij}^{(u,h)} = \phi(i-j; p^{(u,h)})$ for $i,j \in [L]$. Here, $p^{(u,h)}$'s are learnable parameters and we consider $\phi(\cdot; p)$ of the following form:

$$\phi(z;p) = \begin{cases} -p\cdot(z-1) & \text{if } z \ge 1 \\ -\infty & \text{otherwise} \end{cases}. \tag{3}$$

Note that we adopt the Alibi RPE only for simplicity and our results can be extended to other additive RPEs, such as T5 (Raffel et al., 2020). However, extending our analysis to the popular rotary RPE (Su et al., 2024) may be nontrivial, and we leave this for future work.

4 Formulation and Approximation of Induction Head

In this section, we formalize three types of induction head mechanisms with varying levels of complexity. We then theoretically investigate how two-layer single- or multi-head transformers, with or without FFNs, can efficiently implement these mechanisms, highlighting the distinct roles of different transformer submodules

4.1 Vanilla Induction Heads

The original induction head, proposed in Elhage et al. (2021) and Olsson et al. (2022), is regarded as one of the key mechanisms to implement ICL and reasoning. This induction head suggests that two-layer multi-head transformers without FFNs can execute a simple in-context algorithm to predict the next token `b` from a context `[···ab···a]` through retrieval, copying, and pasting, based on in-context bi-gram pairs, as illustrated in Figure 1.

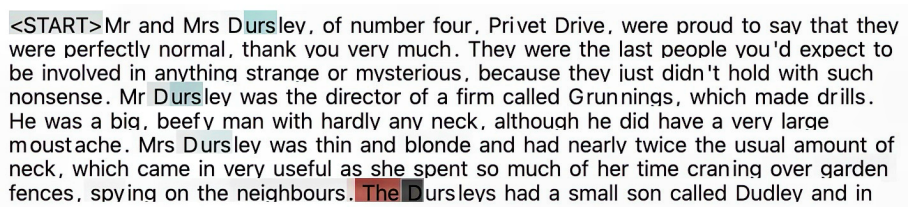

Figure 1: An illustration of the original induction head (taken from Elhage et al. (2021)). The induction head proceeds the context [$\cdots$ The D] by retrieving the preceding information most relevant to the current token (`D`), then copying and pasting the subsequent token (the green `urs`) as the current prediction. Notably, the two self-attention layers focus on the highlighted red and green tokens respectively. For further details, refer to the description below Theorem 4.1.

Formulation of IH_2. Based on the phenomenon illustrated in Figure 1, we define the vanilla induction head $\mathsf{IH}_2 : \cup_{L\in\mathbb{N}^+}\mathbb{R}^{d\times L} \mapsto \mathbb{R}^d$ as follows:

$$\mathsf{IH}_2(X_L) = (x_s)_{s=2}^{L-1}\,\mathrm{softmax}\Big(\big(x_L^\top W^\star x_{s-1}\big)_{s=2}^{L-1}\Big)^\top \tag{4}$$

Specifically, IH_2 retrieves in-context information based on the similarities of in-context bi-gram pairs $\{(x_s, x_L)\}_{s=1}^{L-2}$. Note that the magnitude of matrix $W^\star$ controls the sparsity of retrieval, since increasing $\|W^\star\|$ causes the softmax output to concentrate as a delta measure over the preceding tokens. Additionally, IH_2 can handle input sequences of arbitrary length.

This model retrieves previous tokens x_{s-1}'s that are similar to the current token x_L based on a dot-product similarity, and then copies and pastes x_{s-1}'s subsequent token x_s as the current prediction

x_{L+1}. For example, in Figure 1, the current token x_L is `D`, and the model retrieves previous tokens similar to `D`, copying and pasting its subsequent token `urs` as the prediction.

Comparison with previous formulations. As shown in Figure 1, the current token `D` appears multiple times in the preceding context, and the induction head detects all occurrences of `D`. Our formulation (4) captures this behavior, as the softmax scores for all preceding `D` are identical. In contrast, previous formulations, such as Sanford et al. (2024a) and Sanford et al. (2024b), focus solely on the most recent occurrence of `D`, neglecting this multi-occurrence aspect.

Measure of approximation. Consider a target function $\mathsf{H} : \cup_{L\in\mathbb{N}^+}\mathbb{R}^{d\times L} \mapsto \mathbb{R}^d$, where d is the token dimension and L denotes the sequence length. Given an input sequence $X \in \mathbb{R}^{d\times L}$, transformer TF approximates $\mathsf{H}(X)$ using its last output token, i.e., $\mathsf{TF}_{-1}(X) \in \mathbb{R}^d$. To quantify the approximation error, we define the following metric: for $1 \le p \le +\infty$,

$$\|\mathsf{H} - \mathsf{TF}\|_{L,p} := \left(\mathbb{E}_{X_L}[\|\mathsf{H}(X_L) - \mathsf{TF}_{-1}(X_L)\|_\infty^p]\right)^{1/p}. \tag{5}$$

The next theorem shows that a two-layer *single-head* transformer *without FFNs* suffices to implement vanilla induction heads.

Theorem 4.1 (two-layer single-head TF w/o FFNs)**.** *Let* IH_2 *satisfy Eq.* (4)*. Then exists a constant* $C > 0$ *and a two-layer single-head transformer* TF *(without FFNs), with* $D = 2d$*,* $W_K^{(1,1)} = W_Q^{(1,1)} = 0$*,* $p^{(2,1)} = 0$*, and* $\|W_K^{(2,1)}\|, \|W_Q^{(2,1)}\| \le \mathcal{O}(1, \|W^\star\|_F)$*, such that*

$$\sup_{L\in\mathbb{N}^+} \|\mathsf{IH}_2 - \mathsf{TF}\|_{L,\infty} \le \frac{C}{e^{p^{(1,1)}}}.$$

This theorem shows that single head suffices to approximate the vanilla induction head and moreover, the approximation efficiency is independent of the sequence length. The proof is provided in Appendix A.1, offering the following insights into how two-layer single-head transformers without FFNs implement vanilla induction heads:

- **The first layer** aggregates local tokens and outputs $(z_s = [x_{s-1}, x_s])_{2\le s\le L}$ for the s-th token. This is achieved by using SA with only RPE (no DP). Specifically, RPE allows SA to capture the *preceding token* via $x_{s-1} = \sum_{j\ge 1} x_{s-j}\rho(j)$ for each token x_s, where $\rho(\cdot) = \mathbb{I}\{\cdot = 1\}$. Hence, DP in this layer is not essential and can be omitted.
- **The second layer** extracts the relevant tokens using DP similarity. First, DP computes the similarity $\langle W_Q z_L, W_K z_s\rangle = x_L^\top W^\star x_{s-1}$, where $z_L = [x_{L-1}, x_L]$ and $z_s = [x_{s-1}, x_s]$ represent the hidden tokens outputted by the first layer. This similarity measure enables SA to identify tokens that match x_L. Subsequently, the value/output component extracts x_s in z_s, effectively copying the subsequent token and using it as the current prediction. In this layer, RPE is not necessary and can be omitted.

Remark 4.2 (Alignment with experimental findings)**.** Our theoretical analysis is consistent with the experimental observations reported in Elhage et al. (2021). Specifically, the experiments there demonstrate that SA in the first layer attends to adjacent tokens, while SA in the second layer retrieves information related to the current token. Our analysis identifies components responsible for these two operations, and reveals that *single-head* transformers suffice to perform them efficiently.

4.2 Generalized Induction Heads: In-context n-gram and Generic Similarity

Although the standard induction head defined in Eq. (4) is intuitive, it exhibits notable limitations: **1)** it retrieves only a *single token*, potentially missing *complete local information* and leading to false retrievals; **2)** it relies solely on the *dot-product* to measure the similarity between two tokens, which is not sufficiently general.

Formulation of IH_n. Motivated by the limitation **1)** above, we define a generalized induction head:

$$\mathsf{IH}_n(X_L) = (x_s)_{s=n}^{L-1}\ \mathrm{softmax}\Big(\big(X_{L-n+2:L}^\top W^\star X_{s-n+1:s-1}\big)_{s=n}^{L-1}\Big)^\top, \tag{6}$$

where the patch $X_{s-n+1:s-1}$ incorporates richer local information near x_s and $X_{L-n+2:L}$ denotes the current patches. This formulation is more general than Eq. (4), which only focuses on x_{s-1}. This

induction head operates based on the similarity between the n-gram pairs: $(X_{s-n+1:s-1}; X_{L-n+2:L})$ for $s = n, \ldots, L-1$.

Integrating richer local information facilitates more accurate information retrieval. The model (6) retrieves previous $(n-1)$-token patch that are similar to the current $(n-1)$-token patch, thereby generalizing the vanilla induction head (4), which considers only single-token retrieval. For example, as depicted in Figure 1, if the current local information is `The D` (comprising two tokens), and prior local information such as `Mr D` and `Mrs D` is identified as similar to `The D`, transformer would copy and paste their subsequent token, `urs`, as the prediction.

Theorem 4.3 (two-layer multi-head TF w/o FFNs)**.** *Let* IH_n *satisfy Eq.* (6)*. Then, for any* $q \in \mathbb{N}^+$ *, there exists an absolute constant* $C > 0$ *and a two-layer* H*-head transformer* $\mathsf{TF}(\cdot)$ *(without FFNs), with* $D = nd$*, such that:*

$$\sup_{L \in \mathbb{N}^+} \left\| \mathsf{IH}_n - \mathsf{TF} \right\|_{L,\infty} \leq C \left(\frac{n e^{1+0.01n}}{H} \right)^q .$$

This theorem demonstrates that two-layer multi-head transformers, even without FFNs, can *efficiently* implement the generalized induction head (6). Notably, the approximation error scales as $\mathcal{O}(H^{-q})$, where q can be arbitrarily large, and $H \gtrsim n e^{1+0.01n}$ is sufficient to ensure a good approximation. Furthermore, n is typically small when extracting local semantics. For example, in the vanilla induction head, $n = 2$. The proof of this theorem is provided in Appendix A.2.

The role of multiple heads. In Theorem 4.3, multiple heads are employed in the first layer to approximate the n-gram interaction, represented by the $n-1$ memory kernels $\{\rho_j := \mathbb{I}\{\cdot = j\}\}_{j=1}^{n-1}$. Thus, TF can capture $n-1$ *preceding tokens* via $x_{s-j} = \sum_{k \geq 1} x_{s-k} \rho_j(k)$ for $j \in [n-1]$. Intuitively, as n increases, more memory kernels are required for accurate approximation, necessitating more attention heads. In contrast, Theorem 4.1 only requires approximating a single memory kernel $\mathbb{I}\{\cdot = 1\}$, which can be efficiently achieved using a single attention head.

Recently, Rajaraman et al. (2024) explored a generalized induction head similar to Eq. (6) and showed that multi-layer single-head transformers can implement it. In contrast, our Theorem 4.3 demonstrates that two layers suffice if multi-head self-attention is adopted.

Formulation of GIH_n. Building on the formulation (6), and motivated by the limitation **2)** above, we further consider the following generalized induction head:

$$\mathsf{GIH}_n(X_L) = (x_s)_{s=n}^{L-1} \ \mathrm{softmax}\Big(\big(g\big(X_{L-n+2:L}; X_{s-n+1:s-1}\big) \big)_{s=n}^{L-1} \Big)^\top, \tag{7}$$

where $g : \mathbb{R}^{D \times (n-1)} \times \mathbb{R}^{D \times (n-1)} \to \mathbb{R}$ denotes a generic function measuring the similarity between two $(n-1)$-length patches.

This model retrieves previous relevant multi-token patch $X_{s-n+1:s-1}$ that is similar to the current multi-token patch $X_{L-n+2:L}$, utilizing the generalized similarity function $g(\cdot, \cdot)$. This mechanism is more general than Eq. (6), which is limited to dot-product similarities. For instance, the use of general similarity g enables the model to recognize not only synonymous but also antonymic semantics, thereby improving both the accuracy and diversity of in-context retrievals.

Theorem 4.4 (two-layer multi-head TF with FFNs)**.** *Let* GIH_n *satisfy Eq.* (7)*. Suppose the similarity function* g *is* α*-well-behaved (see Definition A.7). Then, for any* $q \in \mathbb{N}^+$*, there exist constants* $A_{g,q,n}, B_{g,\alpha} > 0$ *and a two-layer* H*-head transformer* $\mathsf{TF}(\cdot)$ *with FFNs of width* M*, such that*

$$\left\| \mathsf{GIH}_n - \mathsf{TF} \right\|_{L,2} \leq A_{g,q,n} H^{-q} + B_{g,\alpha} L^{1/(1+2\alpha)} M^{-\alpha/(1+3\alpha)}.$$

This theorem establishes that if the similarity function g is well-behaved, two-layer multi-head transformers with FFNs can efficiently implement the generalized induction head (7).

The role of FFNs. In contrast to Theorem 4.3, transformer models in Theorem 4.4 include FFNs. These FFN layers are used to approximate the similarity function g. Specifically, we consider the proper orthogonal decomposition (POD) of g, which can be viewed as an extension of the matrix singular value decomposition (SVD) applied to functions of two variables. For $g : \mathcal{I} \times \mathcal{I} \to \mathbb{R}$, its POD is $g(u, v) = \sum_{k=1}^{\infty} \sigma_k \phi_k(u) \psi_k(v)$, where ϕ_k, ψ_k are orthonormal bases for $L^2(\mathcal{I})$ (see Appendix D for details). Intuitively, the FFN in the first layer is used to efficiently approximate K bases (ϕ_i's and ψ_i's). Then, in the second layer, DP in SA can approximately reconstruct g by using the truncated sum $g(u, v) \approx \sum_{k=1}^{K} \sigma_k \phi_k(u) \psi_k(v)$. The complete proof is deferred to Appendix A.3.

5 THE TRANSITION FROM LAZY TO RICH MECHANISMS IN LEARNING INDUCTION HEADS

In this section, we investigate the dynamics of learning induction heads using a transformer, particularly focusing on how this differs from n-gram learning. To facilitate the analysis, we consider a mixed target function that comprises a 4-gram component and a vanilla induction head component as defined in Eq. (4). Specifically, we study the gradient flow dynamics of a two-layer multi-head transformer without FFNs on this task.

5.1 SETUPS

5.1.1 MIXED TARGET FUNCTION

Mixed target function. Let the input sequence be $X = (x_1, \cdots, x_L) \in \mathbb{R}^{1\times L}$. Our mixed target function $f^\star$ contains both a 4-gram component $f^\star_{\mathsf{G}_4}$ and an in-context 2-gram component $f^\star_{\mathsf{IH}_2}$:

$$f^\star(X) := \left(\frac{\alpha^\star}{1+\alpha^\star} f^\star_{\mathsf{G}_4}(X), \frac{1}{1+\alpha^\star} f^\star_{\mathsf{IH}_2}(X)\right)^\top \in \mathbb{R}^2, \tag{8}$$

where $\alpha^\star > 0$ represents the relative weight between the two components: $f^\star_{\mathsf{G}_4}(X)$ and $f^\star_{\mathsf{IH}_2}(X)$. Here, $f^\star_{\mathsf{G}_4}$ represents a 4-gram component and $f^\star_{\mathsf{IH}_2}$ is given by the vanilla induction head (4) to represent a type of in-context 2-gram information:

$$f^\star_{\mathsf{G}_4}(X) := x_{L-2}, \quad f^\star_{\mathsf{IH}_2}(X) := (x_s)_{s=2}^{L-1} \operatorname{softmax}\Big((x_L w^{\star 2} x_{s-1})_{s=2}^{L-1}\Big)^\top.$$

Note that $f^\star_{\mathsf{G}_4}$ denotes a "simplest" 4-gram target, where the next token is predicted according to the conditional probability $p(z|X) = p(z|x_L, x_{L-1}, x_{L-2}) = \mathbb{I}\{z = x_{L-2}\}$.

Remark 5.1 (The reason for considering 4-gram)**.** Note that our target includes a 4-gram component rather than simpler 2- or 3-gram components. As suggested by the experimental results in Elhage et al. (2021), for a learned two-layer transformer that implements vanilla induction head IH_2, the first layer has extracted both x_L and x_{L-1}, which can be outputted using the residual block. Thus, the 2- and 3-gram targets: $p(z|X) = \mathbb{I}\{z = x_L\}$ and $p(z|X) = \mathbb{I}\{z = x_{L-1}\}$ must be learned prior to the induction head. Hence we focus on the more challenging 4-gram target to avoid trivializing the learning process, though our analysis extends straightforwardly to the 2- or 3-gram scenarios.

Remark 5.2 (Extension)**.** Since the transformer studied in this section does not have FFNs, its expressive power is limited. Consequently, we only consider the simple but representative mixed target (8). However, (8) can be generalized to $f^\star(X) = F(f^\star_{\mathsf{G}_4}(X); f^\star_{\mathsf{IH}_2}(X))$, where F is general nonlinear function. Such a form can be efficiently approximated by transformers with FFNs. We leave the optimization analysis under this general setting for future work.

5.1.2 TWO-LAYER MULTI-HEAD TRANSFORMER WITH REPARAMETERIZATION

Two-layer multi-head transformer w/o FFNs. We consider a simple two-layer multi-head transformer TF, where the first layer contains a single head $\mathsf{SA}^{(1,1)}$, and the second layer contain two heads $\mathsf{SA}^{(2,1)}, \mathsf{SA}^{(2,2)}$. Given an input sequence $X = (x_1, \cdots, x_L) \in \mathbb{R}^{1\times L}$, it is first embedded as $X^{(0)} := (X^\top, 0^\top) \in \mathbb{R}^{2\times L}$. The model then processes the sequence as follows:

$$X^{(1)} = X^{(0)} + \mathsf{SA}^{(1,1)}(X^{(0)}),$$
$$\mathsf{TF}(X) = \mathsf{SA}^{(2,1)}(X^{(1)}) + \mathsf{SA}^{(2,2)}(X^{(1)}).$$

Reparameterization. Despite the simplification, the transformer above is still too complicated for dynamics analysis. To overcome this challenge, we adopt the reparametrization trick used in previous works (Tian et al., 2023; Huang et al., 2023; Chen et al., 2024b). Specifically, by Theorem 4.1 and its proof, *the first layer does not require DP, and the second layer does not require RPE*. Moreover, to express the 4-gram component $f^\star_{\mathsf{G}_4}$, we only need an additional head without DP in the second layer. Therefore, we can reparameterize the model as follows:

- **The first layer.** This layer has only one trainable parameter $p^{(1,1)}$. In the unique head $\mathsf{SA}^{(1,1)}$, DP is removed by setting $W_Q^{(1,1)} = W_K^{(1,1)} = 0$, and we let $W_V^{(1,1)} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$. The output

sequence of this layer given by $X^{(1)} = X^{(0)} + \mathsf{SA}^{(1,1)}(X^{(0)}) = \begin{pmatrix} x_1, \cdots, x_L \\ y_1, \cdots, y_L \end{pmatrix}$, where

$$y_s = (x_\tau)_{\tau=1}^{s-1} \operatorname{softmax}\left(\left(-p^{(1,1)}(s-1-\tau) \right)_{\tau=1}^{s-1} \right)^\top, \ s \in [L], \tag{9}$$

where $p^{(1,1)}$, used in RPE (3), is the unique trainable parameter in this layer.

- **The second layer.** This layer has 5 trainable parameters: $w_V^{(2,1)}, w_V^{(2,2)}, p^{(2,1)}, w_K^{(2,2)}, w_Q^{(2,2)}$ for parametrizing the two heads. The first head $\mathsf{SA}^{(2,1)}$ without DP is responsible to fit $f^\star_{\mathsf{G}_4}$, while the second head $\mathsf{SA}^{(2,2)}$ without RPE is responsible to fit $f^\star_{\mathsf{IH}_2}$. Specifically,

$$W_Q^{(2,1)} = W_K^{(2,1)} = 0, W_V^{(2,1)} = \begin{pmatrix} 0 & w_V^{(2,1)} \\ 0 & 0 \end{pmatrix}, p^{(2,2)} = 0, W_V^{(2,2)} = \begin{pmatrix} w_V^{(2,2)} & 0 \\ 0 & 0 \end{pmatrix}.$$

Then the second layer processes $X^{(1)}$ and outputs the last token:

$$\mathsf{TF}_{-1}(X;\theta) = \begin{pmatrix} (w_V^{(2,1)} y_s)_{s=2}^{L-2} \operatorname{softmax}\left(\left(-p^{(2,1)}(L-1-s) \right)_{s=2}^{L-2} \right)^\top \\ (w_V^{(2,2)} x_s)_{s=2}^{L-2} \operatorname{softmax}\left(\left(x_L w_Q^{(2,2)} w_K^{(2,2)} x_{s-1} \right)_{s=2}^{L-2} \right)^\top \end{pmatrix}, \tag{10}$$

where y_s is given by (9). $p^{(2,1)}, w_V^{(2,1)}$ are trainable parameters in $\mathsf{SA}^{(2,1)}$, while $w_Q^{(2,2)}, w_K^{(2,2)}, w_V^{(2,2)}$ are trainable parameters in $\mathsf{SA}^{(2,2)}$.

The set of all six trainable parameters across both layers is denoted by θ.

5.1.3 Gradient Flow on Square Loss

We consider the Gaussian input and square loss, both of which are commonly used in analyzing transformer dynamics and ICL (Akyürek et al., 2022; Huang et al., 2023; Wang et al., 2024). The loss is defined as:

$$\mathcal{L}(\theta) = \frac{1}{2} \mathbb{E}_{X \sim \mathcal{N}(0, I_{L \times L})} \left[\| \mathsf{TF}_{-1}(X;\theta) - f^\star(X) \|_2^2 \right], \tag{11}$$

To characterize the learning of G_4 and IH_2, we introduce the following two partial losses:

$$\mathcal{L}_{\mathsf{G}_4}(\theta) = \frac{1}{2} \mathbb{E}_X \left(\mathsf{TF}_{-1,1}(X;\theta) - f_1^\star(X) \right)^2, \ \mathcal{L}_{\mathsf{IH}_2}(\theta) = \frac{1}{2} \mathbb{E}_X \left(\mathsf{TF}_{-1,2}(X;\theta) - f_2^\star(X) \right)^2,$$

which correspond to the two dimensions in $\mathsf{TF}_{-1}(X;\theta) - f^\star(X) \in \mathbb{R}^2$, respectively. It follows that $\mathcal{L}(\theta) = \mathcal{L}_{\mathsf{G}_4}(\theta) + \mathcal{L}_{\mathsf{IH}_2}(\theta)$.

Gradient flow (GF). We analyze the GF for minimizing the objective (11):

$$\frac{\mathrm{d}\theta(t)}{\mathrm{d}t} = -\nabla \mathcal{L}(\theta(t)), \text{ starting with } \theta(0) = (\sigma_{\text{init}}, \cdots, \sigma_{\text{init}})^\top, \tag{12}$$

where $0 < \sigma_{\text{init}} \ll 1$ is sufficiently small. Note that $\sigma_{\text{init}} \neq 0$ prevents $\nabla \mathcal{L}(\theta(0)) = 0$.

Layerwise training paradigm. We consider a layerwise training paradigm in which, during each stage, only one layer is trained by GF. Specifically,

- **Training Stage I:** In this phase, only the parameter in the first layer, i.e., $p^{(1,1)}$, is trained.
- **Training Stage II:** In this phase,the first layer parameter $p^{(1,1)}$ keeps fixed and only parameters in the second layer are trained: $w_V^{(2,1)}, w_V^{(2,2)}, p^{(2,1)}, w_Q^{(2,2)}, w_K^{(2,2)}$.

This type of layerwise training has been widely used to study the training dynamics of neural networks, including FFN networks (Safran and Lee, 2022; Bietti et al., 2023; Wang et al., 2023) and transformers (Tian et al., 2023; Nichani et al., 2024; Chen et al., 2024b).

Lemma 5.3 (Training Stage I). *For the Training Stage I,* $\lim_{t \to +\infty} p^{(1,1)}(t) = +\infty$.

According to (9), this lemma implies that, at the end of Training Stage I, the first layer captures the preceding token x_{s-1} for each token x_s, i.e., $y_s = x_{s-1}$. This property is crucial for transformers to implement induction heads and aligns with our approximation result in Theorem 4.1. The proof of Lemma 5.3 is deferred to Appendix B.

5.2 Training Stage II: Transition from 4-gram to Induction Head

In this section, we analyze the dynamics in Training Stage II. We start from the following lemma:

Lemma 5.4 (Parameter balance). *In Training Stage II, it holds that* $|w_Q^{(2,2)}(t)|^2 \equiv |w_K^{(2,2)}(t)|^2$.

Lemma 5.4 is similar to the balance result for homogeneous networks (Du et al., 2018), and its proof can be found at the start of Appendix C. By this lemma, we can define $w_{KQ}^{(2,2)} := w_Q \equiv w_K$. Additionally, Lemma 5.3 ensures that $p^{(1,1)} = +\infty$ holds during Stage II. For simplicity, we denote $w_{V_1} := w_V^{(2,1)}, w_{V_2} := w_V^{(2,2)}, p := p^{(2,1)}, w_{KQ} := w_{KQ}^{(2,2)}$. Consequently, the training dynamics are reduced to four parameters

$$\theta = (w_{V_1}, w_{V_2}, p, w_{KQ}),$$

where we still denote the set of parameters as θ without introducing ambiguity. It is important to note that the problem remains **highly non-convex** due to the joint optimization of both inner parameters (p, w_{KQ}) and outer parameters (w_{V_1}, w_{V_2}) in the two heads. At this training stage, GF has a **unique fixed point**:

$$w_{V_1} = \frac{\alpha^\star}{1+\alpha^\star},\ w_{V_2} = \frac{1}{1+\alpha^\star},\ p = +\infty,\ w_{KQ} = w^\star,$$

which corresponds to a global minimizer of the objective (11).

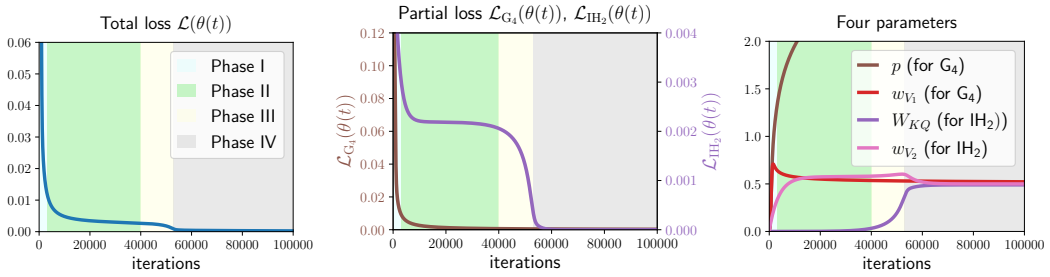

Figure 2: Visualize the dynamical behavior of Training Stage II with total loss, partial loss, and the parameter evolution. Here, $\alpha^\star = 1, w^\star = 0.49, \sigma_{\text{init}} = 0.01, L = 40$. The is clearly shown that transformer learns the 4-gram component first and then, starts to learn the induction head mechanism. Notably, the entire dynamics unfold in four distinct phases, consistent with our theoretical results (Theorem 5.5). For more experimental details, we refer to Appendix E.1.

As shown in Figure 2, a learning transition from the 4-gram mechanism to the induction head mechanism does occur in our setting. Moreover, the learning process exhibits a four-phase dynamics. The next theorem provides a precise characterization of the four phases, whose proof can be found in Appendix C.

Theorem 5.5 (Learning transition and 4-phase dynamics). *Let* $\alpha^\star = \Omega(1)$ *and* $w^\star = \mathcal{O}(1)$, *and we consider the regime of small initialization* $(0 < \sigma_{\text{init}} \ll 1)$ *and long input sequences* $(L \gg 1)$. *Then we have the following results:*

- **Phase I (partial learning).** *In this phase, most of the 4-gram component in the mixed target is learned, while a considerable number of induction head component have not yet been learned. Specifically, let* $T_{\text{I}} = \mathcal{O}(1)$, *then we have the following estimates:*

$$\mathcal{L}_{G_4}(\theta(T_{\text{I}})) \le 0.01 \cdot \mathcal{L}_{G_4}(\theta(0)), \quad \mathcal{L}_{IH_2}(\theta(T_{\text{I}})) \ge 0.99 \cdot \mathcal{L}_{IH_2}(\theta(0)).$$

- **Phase II (plateau) + Phase III (emergence).** *In these two phases, the learning of the induction head first gets stuck in a plateau for* T_{II} *time, then is learned suddenly. Specifically, denoted by an observation time* $T_o = \Theta(L)$, *we have the following tight estimate of the duration:*

$$T_{\text{II}} := \inf\left\{t > T_o : \mathcal{L}_{IH_2}(\theta(t)) \le 0.99 \cdot \mathcal{L}_{IH_2}(\theta(T_o))\right\} = \Theta\left((\alpha^\star+1)^2 L \log(1/\sigma_{\text{init}})/w^{\star 2}\right);$$

$$T_{\text{III}} := \inf\left\{t > T_o : \mathcal{L}_{IH_2}(\theta(t)) \le 0.01 \cdot \mathcal{L}_{IH_2}(\theta(T_o))\right\} = \Theta\left((\alpha^\star+1)^2 L \log(1/\sigma_{\text{init}})/w^{\star 2}\right).$$

During these phases, the parameter w_{KQ} (for learning $w^\star$ in IH_2) increases exponentially:

$$w_{KQ}(t) = \sigma_{\text{init}} \cdot \exp\left(\Theta\left(\frac{w^{\star 2} t}{(1+\alpha^\star)^2 L}\right)\right),\ t < T_{\text{III}}.$$

- **Phase IV (convergence).** *In this phase, the loss converges toward zero. Specifically, the following convergence rates hold for all $t > T_{\text{III}}$:*

$$\mathcal{L}_{\mathsf{G}_4}(\theta(t)) = \mathcal{O}\left(\frac{1}{t}\right), \quad \mathcal{L}_{\mathsf{IH}_2}(\theta(t)) = \mathcal{O}\left(\exp\left(-\Omega\left(\frac{w^{\star 2} t}{(1+\alpha^\star)^2 L}\right)\right)\right),$$

and $\mathcal{L}(\theta(t)) = \mathcal{L}_{\mathsf{G}_4}(\theta(t)) + \mathcal{L}_{\mathsf{IH}_2}(\theta(t))$.

By this theorem, the 4-gram mechanism is first learned, taking time T_{I}. Then, the learning of the induction head mechanism enters a plateau, taking time T_{II}, followed by a sudden emergence of learning, taking time $T_{\text{III}} - T_{\text{II}}$. Finally, the loss for both components converges to zero.

The clear learning transition. When any one of $L, \alpha^\star, 1/\sigma_{\text{init}}, 1/w^\star$ is sufficiently large, Phase II lasts for $T_{\text{II}} \gg 1$. During this phase, the 4-gram component has been learned well but the induction head component remains underdeveloped, demonstrating a distinct learning transition. Moreover, Theorem 5.5 and its proof reveal two key factors that drive this transition:

- **Time-scale separation due to high- and low-order parameter dependence in self attention**. The learning of DP and RPE components differ in their parameter dependencies. DP component exhibits a quadratic dependence on the parameter w_{KQ}, while RPE component shows linear dependence on the parameter p. With small initialization $\sigma_{\text{init}} \ll 1$, a clear time-scale separation emerges: $|\dot{w}_{KQ}| \sim w_{KQ} \ll 1$ (DP, slow dynamics) and $|\dot{p}| \sim 1$ (RPE, fast dynamics). Consequently, the induction head (fitted by DP) is learned much slower than the 4-gram component (fitted by RPE). This time-scale separation accounts for the term $\log(1/\epsilon_{\text{init}})$ in the plateau time T_{II}.
- **Speed difference due to component proportions in the mixed target.** The 4-gram target component and the induction-head component have differing proportions in the mixed target. A simple calculation shows: $\mathcal{L}_{\mathsf{G}_4}(0) \sim \alpha^{\star 2}/(1+\alpha^\star)^2$; If $w^\star = \mathcal{O}(1)$, then $\mathcal{L}_{\mathsf{IH}_2}(0) \sim 1/[(1+\alpha^\star)^2 L]$. Notably, $\mathcal{L}_{\mathsf{IH}_2}(0)$ is significantly smaller than $\mathcal{L}_{\mathsf{G}_4}(0)$. This proportion disparity accounts for the $(1+\alpha^\star)^2 L$ term in the plateau time T_{II}.

Proof idea. We highlight that our fine-grained analysis of entire learning process is guided by two key observations: 1) the dynamics of the two heads can be decoupled; 2) there exist a distinct transition point in the dynamics of each head, as shown in Figure 2 (right). These insights lead us to divide the analysis of each head into two phases: a monotonic phase and a convergence phase.

6 Experimental Validation

To further support both our approximation results and optimization dynamics, we conduct a series of experiments ranging from simple toy models to real-world natural language training tasks. Due to space constraints, the detailed experimental setups and results are presented in Appendix E.

7 Conclusion

In this work, we present a comprehensive theoretical analysis of how transformers implement induction heads, examining both the approximation and optimization aspects. From the approximation standpoint, we identify the distinct roles of each transformer component in implementing induction heads of varying complexity. On the optimization side, we analyze a toy setting, where we clearly characterize how learning transitions from n-grams to induction heads. Looking forward, an important direction for future research is to investigate the dynamics of learning general induction heads, which are crucial for realizing stronger ICL capabilities.

REFERENCES

Emmanuel Abbe, Samy Bengio, Enric Boix-Adsera, Etai Littwin, and Joshua Susskind. Transformers learn through gradual rank increase. *Advances in Neural Information Processing Systems*, 36, 2024. 3

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36, 2024. 3

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022. 2, 8

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016. 42, 43

Davoud Ataee Tarzanagh, Yingcong Li, Xuechen Zhang, and Samet Oymak. Max-margin token selection in attention mechanism. *Advances in Neural Information Processing Systems*, 36:48314–48362, 2023. 3

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *arXiv preprint arXiv:2306.04637*, 2023. 2, 32

Andrew R Barron. Neural net approximation. In *Proc. 7th Yale Workshop on Adaptive and Learning Systems*, volume 1, pages 69–72, 1992. 19

Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993. 19

Andrew R Barron. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14(1):115–133, 1994. 19

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the ability and limitations of transformers to recognize formal languages. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. 2

Alberto Bietti, Joan Bruna, and Loucas Pillaud-Vivien. On learning gaussian multi-index models with gradient flow. *arXiv preprint arXiv:2310.19793*, 2023. 8

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. 2, 43

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. 1

Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. *arXiv preprint arXiv:2402.19442*, 2024a. 3

Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Unveiling induction heads: Provable training dynamics and feature learning in transformers. *arXiv preprint arXiv:2409.10559*, 2024b. 3, 7, 8

Xingwu Chen and Difan Zou. What can transformer learn with varying depth? case studies on sequence learning tasks. *arXiv preprint arXiv:2404.01601*, 2024. 2

Joy Crosbie and Ekaterina Shutova. Induction heads as an essential mechanism for pattern matching in in-context learning. *arXiv preprint arXiv:2407.07011*, 2024. 1

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *International Conference on Learning Representations*, 2019. 2

Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *Advances in neural information processing systems*, 31, 2018. 9

Weinan E, Chao Ma, and Lei Wu. A priori estimates of the population risk for two-layer neural networks. *Communications in Mathematical Sciences*, 17(5):1407–1425, 2019. 19

Weinan E, Chao Ma, and Lei Wu. The barron space and the flow-induced function spaces for neural network models. *Constructive Approximation*, pages 1–38, 2021. 19

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pages 5793–5831. PMLR, 2022. 2, 39

Benjamin L Edelman, Ezra Edelman, Surbhi Goel, Eran Malach, and Nikolaos Tsilivis. The evolution of statistical induction heads: In-context learning markov chains. *arXiv preprint arXiv:2402.11004*, 2024. 2, 43, 44

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. 1, 2, 3, 4, 5, 7

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. 2

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. *International Conference on Machine Learning*, 2023. 2

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 1

Michael Hahn. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171, 2020. 2

Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. *arXiv preprint arXiv:2310.05249*, 2023. 3, 7, 8

Samy Jelassi, David Brandfonbrener, Sham M Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. *arXiv preprint arXiv:2402.01032*, 2024. 1

Haotian Jiang and Qianxiao Li. Approximation theory of transformer networks for sequence modeling. *arXiv preprint arXiv:2305.18475*, 2023. 2, 19

Hongkang Li, Meng Wang, Songtao Lu, Xiaodong Cui, and Pin-Yu Chen. Training nonlinear transformers for efficient in-context learning: A theoretical learning and generalization analysis. *arXiv preprint arXiv:2402.15607*, 2024. 3

Chao Ma, Stephan Wojtowytsch, Lei Wu, and Weinan E. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. *arXiv preprint arXiv:2009.10713*, 2020. 19

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. 2

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. 40, 41

William Merrill and Ashish Sabharwal. The expresssive power of transformers with chain of thought. *arXiv preprint arXiv:2310.07923*, 2023. 2

William Merrill, Ashish Sabharwal, and Noah A Smith. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10:843–856, 2022. 2

Eshaan Nichani, Alex Damian, and Jason D Lee. How transformers learn causal structure with gradient descent. *arXiv preprint arXiv:2402.14735*, 2024. 3, 8

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022. 1, 2, 3, 4

Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is turing complete. *The Journal of Machine Learning Research*, 22(1):3463–3497, 2021. 2

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *International Conference on Learning Representations*, 2022. 4

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 4

Nived Rajaraman, Marco Bondaschi, Kannan Ramchandran, Michael Gastpar, and Ashok Vardhan Makkuva. Transformers on markov data: Constant depth suffices. *arXiv preprint arXiv:2407.17686*, 2024. 2, 6

Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. *International Conference on Learning Representations*, 2024. 2

Michael Reed and Barry Simon. *Methods of modern mathematical physics: Functional analysis*, volume 1. Gulf Professional Publishing, 1980. 19

Itay Safran and Jason Lee. Optimization-based separations for neural networks. In *Conference on Learning Theory*, pages 3–64. PMLR, 2022. 8

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. One-layer transformers fail to solve the induction heads task. *arXiv preprint arXiv:2408.14332*, 2024a. 2, 5

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. *arXiv preprint arXiv:2402.09268*, 2024b. 1, 2, 5

Johannes Schmidt-Hieber et al. Nonparametric regression using deep neural networks with ReLU activation function. *Annals of Statistics*, 48(4):1875–1897, 2020. 20

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948. 1, 3

Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. Do pretrained transformers really learn in-context by gradient descent? *arXiv preprint arXiv:2310.08540*, 2023. 2

Jonathan W Siegel and Jinchao Xu. Approximation rates for neural networks with general activation functions. *Neural Networks*, 128:313–321, 2020. 19

Jiajun Song, Zhuoyan Xu, and Yiqiao Zhong. Out-of-distribution generalization via composition: a lens through induction heads in transformers. *arXiv preprint arXiv:2408.09503*, 2024. 1

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 4

Davoud Ataee Tarzanagh, Yingcong Li, Christos Thrampoulidis, and Samet Oymak. Transformers as support vector machines. *arXiv preprint arXiv:2308.16898*, 2023. 3

Christos Thrampoulidis. Implicit bias of next-token prediction. *arXiv preprint arXiv:2402.18551*, 2024. 3

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, 2023. 3, 7, 8

Bhavya Vasudeva, Puneesh Deora, and Christos Thrampoulidis. Implicit bias and fast convergence rates for self-attention. *arXiv preprint arXiv:2402.05738*, 2024. 3

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 3

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023. 2

Mingze Wang and Weinan E. Understanding the expressive power and mechanisms of transformer for sequence modeling. *Advances in Neural Information Processing Systems*, 2024. 39

Zihao Wang, Eshaan Nichani, and Jason D Lee. Learning hierarchical polynomials with three-layer neural networks. *arXiv preprint arXiv:2311.13774*, 2023. 8

Zixuan Wang, Stanley Wei, Daniel Hsu, and Jason D Lee. Transformers provably learn sparse token selection while fully-connected nets cannot. *arXiv preprint arXiv:2406.06893*, 2024. 2, 3, 8

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, 2022. 2

Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking like transformers. In *International Conference on Machine Learning*, pages 11080–11090. PMLR, 2021. 2

Norman Yarvin and Vladimir Rokhlin. Generalized gaussian quadratures and singular value decompositions of integral operators. *SIAM Journal on Scientific Computing*, 20(2):699–718, 1998. 19

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. 2

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023. 3

Appendix

A Proofs in Section 4

A.1 Proof of Theorem 4.1

$$\mathsf{IH}_2(X_L) = (x_s)_{s=2}^{L-1}\ \mathrm{softmax}\Big(\big(x_L^\top W^\star x_{s-1}\big)_{s=2}^{L-1}\Big)^\top, \tag{13}$$

Theorem A.1 (Restatement of Theorem 4.1)**.** *Let* IH_2 *satisfy Eq. (13). Then, there exists a constant* $C > 0$ *and a two-layer single-head transformer* TF *(without FFNs), with* $D = 2d$*,* $W_K^{(1,1)} = W_Q^{(1,1)} = 0$*,* $p^{(2,1)} = 0$*, and* $\|W_K^{(2,1)}\|, \|W_Q^{(2,1)}\| \le \mathcal{O}(1, \|W^\star\|_F)$*, such that*

$$\sup_{L\in\mathbb{N}^+} \|\!|\mathsf{IH}_2 - \mathsf{TF}|\!\|_{L,\infty} \le \frac{C}{e^{p^{(1,1)}}}.$$

Proof. We consider two-layer single-head transformer without FFN, where the first layer has the residual block, while the second layer does not have the residual block.

We first embed each token into $\mathbb{R}^D$ as $\begin{pmatrix} x_s \\ 0 \end{pmatrix}$ and take $W_V^{(1)} = \begin{pmatrix} 0 & 0 \\ I_{d\times d} & 0 \end{pmatrix}$, then the s-th output token of the first layer is

$$\begin{pmatrix} x_s \\ y_s \end{pmatrix} = \begin{pmatrix} x_s \\ (x_\tau)_{\tau=1}^{s-1}\ \mathrm{softmax}\Big(\big(-p^{(1,1)}(s-1-\tau)\big)_{\tau=1}^{s-1}\Big) \end{pmatrix}.$$

Then for the second layer, we choose $p^{(2,1)} = 0$,

$$W_Q^{(2,1)} = \begin{pmatrix} 0 & 0 \\ I_{d\times d} & 0 \end{pmatrix}, \; W_K^{(2,1)} = \begin{pmatrix} 0 & 0 \\ 0 & W^\star \end{pmatrix}, \; W_V^{(2,1)} = \begin{pmatrix} I_{d\times d} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{D\times D},$$

and the projection $W_O^{(2)} = (I_{d\times d} \quad 0_{d\times d}) \in \mathbb{R}^{d\times D}$.

Then the last output token of the second layer is

$$(x_s)_{s=2}^{L-1} \operatorname{softmax}\Big(\big(x_L^\top W^\star y_s\big)_{s=2}^{L-1}\Big)^\top.$$

By Lemma D.1 , for any $L \in \mathbb{N}^+$

$$\begin{aligned}
&\|\!|\mathsf{IH}_2 - \mathsf{TF}|\!\|_{L,\infty} \\
=&\sup_{X_L} \|\mathsf{IH}(X_L) - \mathsf{TF}_{-1}(X_L)\|_\infty \\
=&\left\| (x_s)_{s=2}^{L-1} \operatorname{softmax}\left(\big(x_L^\top W^\star y_s\big)_{s=2}^{L-1}\right)^\top - (x_s)_{s=2}^{L-1} \operatorname{softmax}\left(\big(x_L^\top W^\star x_{s-1}\big)_{s=2}^{L-1}\right)^\top \right\|_\infty \\
\le&\|(x_s)_{s=2}^{L-1}\|_{\infty,\infty} \left\| \operatorname{softmax}\left(\big(x_L^\top W^\star y_s\big)_{s=2}^{L-1}\right) - \operatorname{softmax}\left(\big(x_L^\top W^\star x_{s-1}\big)_{s=2}^{L-1}\right)\right\|_1 \\
\le&2 \sup_{2\le s\le L-1} |x_L^\top W^\star y_s - x_L^\top W^\star x_{s-1}| \\
\le&2\|x_L^\top W^\star\|_1 \sup_s \|y_s - x_{s-1}\|_\infty \\
\le&2\sum_{i,j} |W^\star_{i,j}| \sup_s \left\| (x_\tau)_{\tau=1}^{s-1} \operatorname{softmax}\left(\Big(-p^{(1,1)}(s-1-\tau)\Big)_{\tau=1}^{s-1}\right)^\top - x_{s-1}\right\|_\infty \\
\le&2\|W^\star\|_{1,1} \sup_s \left\| \operatorname{softmax}\left(\Big(-p^{(1,1)}(s-1-\tau)\Big)_{\tau=1}^{s-1}\right) - \boldsymbol{e}_{s-1}\right\|_1 \\
\le&4\|W^\star\|_{1,1} \frac{e^{-p^{(1,1)}}}{1-e^{-p^{(1,1)}}} \le \mathcal{O}\left(e^{-p^{(1,1)}}\right).
\end{aligned}$$

□

A.2 Proof of Theorem 4.3

$$\mathsf{IH}_n(X_L) = (x_s)_{s=n}^{L-1} \operatorname{softmax}\Big(\big(X_{L-n+2:L}^\top W^\star X_{s-n+1:s-1}\big)_{s=n}^{L-1}\Big)^\top, \tag{14}$$

Theorem A.2 (Restatement of Theorem 4.3)**.** *Let IH_n satisfy Eq. (14). Then for any $q \in \mathbb{N}^+$, there exists a constant $C_{q,n} > 0$ and a two-layer H-head transformer $\mathsf{TF}(\cdot)$ (without FFNs), with $D = nd$, such that:*

$$\sup_{L\in\mathbb{N}^+} \|\!|\mathsf{IH}_n - \mathsf{TF}|\!\|_{L,\infty} \le C\left(\frac{ne^{1+0.01n}}{H}\right)^q.$$

Proof. We consider two-layer multi-head transformer without FFN, where the first layer has the residual block, while the second layer does not have the residual block.

First, we choose the embedding dimension $D = nd$, and parameters in the embedding map

$$W_E = \begin{pmatrix} I_{d\times d} \\ 0_{(D-d)\times d} \end{pmatrix} \in \mathbb{R}^{D\times d}, \quad b_E = 0 \in \mathbb{R}^D,$$

then each token $x_s^{(0)}$ after embedding is

$$x_s^{(0)} = W_E x_s + b_E = \begin{pmatrix} x_s \\ 0 \end{pmatrix} \in \mathbb{R}^D.$$

This proof can be summarized as the following process for TF_{-1}:

$$(x_s)_{s=n}^{L-1} \xrightarrow{\text{softmax}} \left((X_{L-n+2:L}^\top W^\star X_{s-n+1:s-1})_{s=n}^{L-1}\right)^\top$$
$$\xrightarrow{\text{Step II. 2-st Attn}} X_{L-n+2:L}$$
$$\xrightarrow{\text{Step I. 1-st Attn}} (x_L^\top, 0_{D-d}^\top)^\top$$

Step I. The first layer. We use 1-st Attn with residual to copy the previous tokens $(x_{s-n+1}, \cdots, x_{s-1})$ of each token x_s. We use $H = \sum_{i=1}^{n-1} H_i$ attention heads to realize this step, and the following projection matrices are needed:

$$P_i := (0_{d\times id}, I_{d\times d}, 0_{d\times(D-(i+1)d)}) \in \mathbb{R}^{d\times D}, \quad i = 1, \ldots, n-1.$$

By lemma D.2, there exist a constant $C > 0$ such that: for any rate $q \in \mathbb{N}^+$, there exists a function

$$\phi_i^{\exp}(t) = \sum_{h=1}^{H_i} \alpha_{h,i} e^{-\beta_{h,i}(t-1)}$$

such that $\beta_h > 0$ and

$$\|\mathbb{I}\{\cdot = i\} - \phi_i^{\exp}(\cdot)\|_{\ell_1(\mathbb{N})} = \sum_{s=i}^{+\infty} |\mathbb{I}\{s = 1\} - \phi^{\exp}(s)| \le \frac{Ce^{q+0.01(q+1)i}}{H_i^q}$$

For $h = \sum_{j=1}^{i-1} H_j, 1 + \sum_{j=1}^{i-1} H_j, \ldots, \sum_{j=1}^{i} H_j$, we choose parameters as follows

$$p^{(1,h)} = \beta_{h,i}, \quad W_V^{(1,h)} = \alpha_{h,i} \left(\sum_{j=0}^{H_i} \exp(-\beta_{h,i}(j-1))\right) S_i,$$
$$W_K^{(1,h)} = W_Q^{(1,h)} = 0, \quad W_O^{(1)} = I_{D\times D}$$

where $S_i \in \mathbb{R}^{D\times D}$ is a shift matrix that takes out the first d elements of a vector and shifts it backward to the $(id+1)$-th to $(i+1)d$-th elements. Then

$$\left(P_i \sum_{h=\sum_{j=1}^{i-1} H_j}^{\sum_{j=1}^{i} H_j} \mathsf{SA}^{(1,h)}(X_L^{(0)})\right)_{-1} = \sum_{h=\sum_{j=1}^{i-1} H_j}^{\sum_{j=1}^{i} H_j} \alpha_{h,i} \sum_{s=1}^{L-1} e^{-\beta_{h,i}(s-1)} x_{L-s}.$$

We denote $x_L^{(1)} := \mathsf{SA}^{(1)}(X_L^{(0)})_{-1}$, then the approximation error of this step is

$$\varepsilon_{\mathsf{SA}}^{(1)} := \sup_s \left\| x_s^{(1)} - \begin{pmatrix} x_s \\ \vdots \\ x_{s-n+1} \end{pmatrix} \right\|_\infty \le \sup_s \sum_{i=1}^{n-1} \left\| P_i x_s^{(1)} - x_{s-i} \right\|_\infty$$
$$\le \sup_s \sum_{i=1}^{n-1} \|\mathbb{I}\{\cdot = i\} - \phi_i^{\exp}(\cdot)\|_{\ell_1(\mathbb{N})} \le Ce^q \sum_{i=1}^{n-1} \frac{e^{0.01(q+1)i}}{H_i^q}.$$

Consequently, one detail is to assign the head number $\{H_i\}_{i=1}^n$ such that the error's sum $\sum_{i=1}^{n-1} \frac{e^{0.01(q+1)i}}{H_i^q}$ is as small as possible. Our way is solving the minimization problem

$$\min : \sum_{i=1}^{n-1} \frac{e^{0.01(q+1)i}}{H_i^q}$$
$$\text{s.t.} \sum_{i=1}^{n-1} H_i = H,$$

which suggests that we should choose the head number:

$$H_i = \frac{e^{0.01i}}{\sum_{j=1}^{n-1} e^{0.01j}}, i \in [n-1].$$

Thus, we obtain the bound

$$\varepsilon_{\mathsf{SA}}^{(1)} \leq \frac{Ce^q}{H^q}\left(\sum_{i=1}^{n-1} e^{0.01i}\right)^q \leq C\left(\frac{ene^{0.01n}}{H}\right)^q.$$

Additionally, we denote the output of this step as

$$x_s^{(1)} := \begin{pmatrix} x_s \\ \hat{x}_{s-1} \\ \vdots \\ \hat{x}_{s-n+1} \end{pmatrix} := \hat{X}_{s-n+1:s-1}.$$

We choose H large enough so that $x_s^{(1)} \in [-2,2]^D$.

Step II. The second layer. For the second Attn, we only need use the first head (by setting $W_V^{(2,h)} = 0$ for $h \geq 1$). Specifically, we choose $p^{(2,1)} = 0$,

$$W_Q^{(2,1)} = \begin{pmatrix} 0 & 0 \\ I_{(D-d)\times(D-d)} & 0 \end{pmatrix}, W_K^{(2,1)} = \begin{pmatrix} 0 & 0 \\ 0 & W^\star \end{pmatrix}, W_V^{(2)} = \begin{pmatrix} I_{d\times d} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{D\times D},$$

and the projection $W_O^{(2)} = \begin{pmatrix} I_{d\times d} & 0_{(D-d)\times d} \end{pmatrix} \in \mathbb{R}^{d\times D}$.

Then the output of this layer is

$$x_L^{(2)} = (x_s)_{s=n}^{L-1}\ \mathrm{softmax}\Big(\big(\hat{X}_{L-n+2:L}^\top W^\star \hat{X}_{s-n+1:s-1}\big)_{s=n}^{L-1}\Big)^\top$$

According to Lemma D.1,

$$\begin{aligned}
&\left\| x_L^{(2)} - (x_s)_{s=n}^{L-1}\ \mathrm{softmax}\Big(\big(X_{L-n+2:L}^\top W^\star X_{s-n+1:s-1}\big)_{s=n}^{L-1}\Big)^\top \right\|_\infty \\
\leq &\sum_{s=n}^{L-1} \Big|\mathrm{softmax}\Big(\big(\hat{X}_{L-n+2:L}^\top W^\star \hat{X}_{s-n+1:s-1}\big)_{s=n}^{L-1}\Big)^\top \\
&\qquad - \mathrm{softmax}\Big(\big(X_{L-n+2:L}^\top W^\star X_{s-n+1:s-1}\big)_{s=n}^{L-1}\Big)^\top\Big)\Big| \\
\leq &2\max_s |\hat{X}_{L-n+2:L}^\top W^\star \hat{X}_{s-n+1:s-1} - X_{L-n+2:L}^\top W^\star X_{s-n+1:s-1}| \\
\leq &2\|W^\star\|_{(1,1)} \cdot \varepsilon_{\mathsf{SA}}^{(1)}.
\end{aligned}$$

Since the above inequality holds for any L and X_L, we have:

$$\sup_{L\in\mathbb{N}^+} \|\mathsf{IH}_n - \mathsf{TF}\|_{L,\infty} \leq C\left(\frac{ne^{1+0.01n}}{H}\right)^q.$$

Additionally, our proof primarily focuses on the case of $H \geq n$. For the case of $H < n$, the approximation error can be trivially bounded by:

$$\sup_{L\in\mathbb{N}^+} \|\mathsf{IH}_n - \mathsf{TF}\|_{L,\infty} \leq \sup_{L\in\mathbb{N}^+} \|\mathsf{IH}_n - 0\|_{L,\infty} \leq 1 \leq C\left(\frac{ne^{1+0.01n}}{H}\right)^q.$$

Then, the two cases can be unified.

□

A.3 Proof of Theorem 4.4

A.3.1 Approximation Results for FFNs

Since the setting in this subsection includes FFNs, we introduce the following preliminary results about the approximation of FFNs.

The well-known universal approximation result for two-layer FNNs asserts that two-layer FNNs can approximate any continuous function (Barron, 1992; 1993; 1994). Nonetheless, this result lacks a characterization of the approximation efficiency, i.e., how many neurons are needed to achieve a certain approximation accuracy? Extensive pre-existing studies aimed to address this gap by establishing approximation rates for two-layer FFNs. A representative result is the Barron theory (E et al., 2019; 2021; Ma et al., 2020): any function f in Barron space $\mathcal{B}$ can be approximated by a two-layer FFN with M hidden neurons can approximate f efficiently, at a rate of $\mathcal{O}(\|f\|_{\mathcal{B}}/\sqrt{M})$. This rate is remarkably independent of the input dimension, thus avoiding the Curse of Dimensionality. Specifically, Barron space is defined in as follows:

Definition A.3 (Barron space (E et al., 2019; 2021; Ma et al., 2020))**.** Consider functions $f : X \to \mathbb{R}$ that admit the following representation: $f(x) = \int_\Omega a\sigma(b^\top x + c)\rho(\mathrm{d}a, \mathrm{d}b, \mathrm{d}c),\ x \in X$. For any $p \in [1, +\infty]$, we define the Barron norm as $\|f\|_{\mathcal{B}_p} := \inf_\rho \left(\mathbb{E}_\rho \left[|a|^p (\|b\|_1 + |c|)^p\right]\right)^{1/p}$. Then the Barron space are defined as: $\mathcal{B}_p := \{f \in \mathcal{C} : \|f\|_{\mathcal{B}_p} < +\infty\}$.

Proposition A.4 (E et al. (2019))**.** *For any $p \in [1, +\infty]$, $\mathcal{B}_p = \mathcal{B}_\infty$ and $\|f\|_{\mathcal{B}_p} = \|f\|_{\mathcal{B}_\infty}$.*

Remark A.5. From the Proposition above, the Barron spaces $\mathcal{B}_p$ are equivalent for any $p \in [1, +\infty]$. Consequently, in this paper, we use $\mathcal{B}$ and $\|\cdot\|_{\mathcal{B}}$ to denote the Barron space and Barron norm.

The next lemma illustrates the approximation rate of two-layer FFNs for Barron functions.

Lemma A.6 (Ma et al. (2020))**.** *For any $f \in \mathcal{B}$, there exists a two-layer ReLU neural network $\mathsf{FFN}(x) = \sum_{k=1}^{M} a_w \sigma(b_k^\top x + c_k)$ with M neurons such that*

$$\|f - \mathsf{FFN}\|_{L^\infty([0,1]^d)} \leq \tilde{\mathcal{O}}\left(\frac{\|f\|_{\mathcal{B}}}{\sqrt{M}}\right).$$

A.3.2 Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) can be viewed as an extension of the matrix singular value decomposition (SVD) applied to functions of two variables. Specifically, for a square integrable function $g : \mathcal{I} \times \mathcal{I} \to \mathbb{R}$, it has the following decomposition (Theorem 3.4 in Yarvin and Rokhlin (1998), Theorem VI.17 in Reed and Simon (1980)):

$$g(u, v) = \sum_{k=1}^{\infty} \sigma_k \phi_k(u) \psi_k(v). \tag{15}$$

Here, ϕ_k, ψ_k are orthonormal bases for $L^2(\mathcal{I})$, and $\sigma_k \geq 0$ are the singular values, arranged in descending order.

Recently, Jiang and Li (2023) also used POD to study the approximation rate of single-layer single-head Transformer for the targets with nonlinear temporal kernels.

Given that two-layer FFNs can efficiently approximate Barron functions (Ma et al., 2020), which is dense in $L^2([0,1]^d)$ (Siegel and Xu, 2020), we introduce the following technical definition regarding the well-behavior POD, which is used for our theoretical analysis.

Definition A.7 (Well-behaved POD)**.** Let the POD of $g : [-2, 2]^D \times [-2, 2]^D \mapsto \mathbb{R}$ be $g(u, v) = \sum_{k=1}^{\infty} \sigma_k \phi_k(u) \psi_k(v)$. We call the function g has α-well-behaved POD ($\alpha > 0$) if:

- The decay rate of singular values satisfies $\sigma_k = \mathcal{O}(1/k^{1+\alpha})$;
- The L^∞ norms, Barron norms, and Lipschitz norms of the POD bases are all uniformly bounded: $\sup_k \left(\|\phi_k\|_{L^\infty} \vee \|\psi_k\|_{L^\infty} \vee \|\phi_k\|_{\mathcal{B}} \vee \|\psi_k\|_{\mathcal{B}} \vee \|\phi_k\|_{\mathrm{Lip}} \vee \|\psi_k\|_{\mathrm{Lip}}\right) < \infty$.

A.3.3 PROOF OF THEOREM 4.4

$$\mathsf{GIH}_n(X_L) = (x_s)_{s=n}^{L-1}\ \mathrm{softmax}\Big(\big(g(X_{L-n+2:L}; X_{s-n+1:s-1})\big)_{s=n}^{L-1}\Big)^\top, \tag{16}$$

Theorem A.8 (Restatement of Theorem 4.4)**.** *Let* GIH_n *satisfy Eq.* (16)*. Suppose the similarity function* g *is* α*-well-behaved (see Definition A.7). Then, for any* $q > 0$*, there exist constants* $A_{g,q,n}, B_{g,\alpha} > 0$ *and a two-layer* H*-head transformer* $\mathsf{TF}(\cdot)$ *with FFN of width* M*, such that the following approximation rate holds:*

$$\|\mathsf{GIH}_n - \mathsf{TF}\|_{L,2} \le \frac{A_{g,q,n}}{H^n} + \frac{B_{g,\alpha} L^{1/(1+2\alpha)}}{M^{\alpha/(1+3\alpha)}}.$$

Proof. We consider two-layer multi-head transformer with FFN, where the first layer has the residual block.

First, we set an constant $K \in \mathbb{N}^+$, and we will optimize it finally. We choose the embedding dimension $D = nd + 2(n-1)K$, and the flowchart of the theorem proof is as follows:

$$(x_s)_{s=n}^{L-1}\ \mathrm{softmax}\Big(\big(g(X_{L-n+2:L}; X_{s-n+1:s-1})\big)_{s=n}^{L-1}\Big)^\top$$

Step III. 2-st Attn $\uparrow$

$$\Big(x_L^\top, \dots, \hat{x}_{L-n+1}, \hat{\phi}_1(\hat{X}_{L-n+2:L}), \dots, \hat{\phi}_K(\hat{X}_{L-n+2:L}),$$
$$\hat{\psi}_1(\hat{X}_{L-n+1:L-1}), \dots, \hat{\psi}_K(\hat{X}_{L-n+1:L-1})\Big)^\top$$

Step II. 1-st FFN $\uparrow$

$$(x_L^\top, \hat{x}_{L-1}, \dots, \hat{x}_{L-n+1}, 0^\top)^\top$$

Step I. 1-st Attn $\uparrow$

$$(x_t^\top, 0^\top)^\top$$

Recalling Definition A.7, there exists constants $C_g^\infty, C_g^{\mathcal{B}}, C_g^{\mathrm{Lip}} > 0$ such that:

$$\sup_k \left(\|\phi_k\|_\infty \vee \|\psi_k\|_\infty\right) \le C_g^\infty,\ \sup_k \left(\|\phi_k\|_{\mathcal{B}} \vee \|\psi_k\|_{\mathcal{B}}\right) \le C_g^{\mathcal{B}},\ \sup_k \Big(\|\phi_k\|_{\mathrm{Lip}} \vee \|\psi_k\|_{\mathrm{Lip}}\Big) \le C_g^{\mathrm{Lip}}.$$

Additionally, $\sigma_k = \mathcal{O}(1/k^{1+\alpha})$ implies that there exits a $C_\alpha > 0$ such that:

$$\sum_{k=K}^{\infty} \sigma_k < \frac{C_\alpha}{K^\alpha}, \quad \forall K \ge 1.$$

Step I: Error in 1-st Attn layer. This step is essentially the same as Step I in the proof of Theorem 4.3, so we write down the error of the first Attn layer directly:

$$\epsilon_{\mathsf{SA}}^{(1)} \le \frac{C_{q,n}}{H^n}.$$

Moreover, due to $\left\|\hat{X}_{s-n+2:s} - X_{s-n+2:s}\right\| \le \epsilon_{\mathsf{SA}}^{(1)}$, for all s, we have:

$$\hat{X}_{s-n+2:s} \in [-2, 2]^D.$$

Step II: Error in 1-st FFN layer. The 1-st FFN is used to approximate ϕ_k, ψ_k $(k = 1, \dots, K)$. Each function is approximated by a 2-layer neural networks with $\frac{M}{2K}$ neurons defined on $\mathbb{R}^D$, and the FFNs are concatenated together (refer to section 7.1 "Parallelization" in Schmidt-Hieber et al. (2020)) as $\mathrm{FFN}^{(1)}$. We denote them as

$$\hat{\phi}_k(y) = \sum_{m=1}^{\frac{M}{2K}} a_m^k \sigma(b_m^{k\top} y + c_m^k)$$

$$\hat{\psi}_k(y) = \sum_{m=1}^{\frac{M}{2K}} \tilde{a}_m^k \sigma(\tilde{b}_m^{k\top} y + \tilde{c}_m^k)$$

Then according to lemma A.6, such FFNs exist and satisfy the following properties hold for all $1 \le k \le K$:

$$\|\hat{\phi}_k - \phi_k\|_{L^\infty([-2,2]^D)} \le \tilde{\mathcal{O}}\left(\|\phi_k\|_{\mathcal{B}}\sqrt{\frac{K}{M}}\right) \le \epsilon_{\mathsf{FFN}}^{(1)},$$

$$\|\hat{\psi}_k - \psi_k\|_{L^\infty([-2,2]^D)} \le \tilde{\mathcal{O}}\left(\|\psi_k\|_{\mathcal{B}}\sqrt{\frac{K}{M}}\right) \le \epsilon_{\mathsf{FFN}}^{(1)},$$

where

$$\epsilon_{\mathsf{FFN}}^{(1)} := \cdot\tilde{\mathcal{O}}\left(C_g^{\mathcal{B}}\sqrt{\frac{K}{M}}\right).$$

Step III: Error in 2nd Attn layer.

We use matrices in the second layer to take out elements needed

$$W_V^{(2)} = (I_{d\times d}, 0_{d\times D}) \in \mathbb{R}^{d\times D},$$

$$W_K^{(2,1)} = \sum_{i=k}^{K} \sqrt{\sigma_k} e_{k,(n-1)d+k} \in \mathbb{R}^{D\times D},$$

$$W_Q^{(2,1)} = \sum_{k=1}^{K} \sqrt{\sigma_k} e_{k,(n-1)d+K+k} \in \mathbb{R}^{D\times D}.$$

We denote the rank-K truncation of g as

$$g_K := \sum_{k=1}^{K} \sigma_k \phi_k \psi_k,$$

and its approximation as

$$\hat{g}_K := \sum_{k=1}^{K} \sigma_k \hat{\phi}_k \hat{\psi}_k$$

The second FFN is set to be identity map and we denote the final output as

$$x_L^{(2)} := (x_s)_{s=n}^{L-1}\ \mathrm{softmax}\Big(\big(\hat{g}_K\big(\hat{X}_{L-n+2:L}; \hat{X}_{s-n+1:s-1}\big)\big)_{s=n}^{L-1}\Big)^\top.$$

First, we consider the error under the first norm, $\|\cdot\|_\infty$, which can be divided the total error into three components:

$$\begin{aligned}
&\Big\|x_L^{(2)} - (x_s)_{s=n}^{L-1}\ \mathrm{softmax}\Big(\big(g\big(X_{L-n+2:L}; X_{s-n+1:s-1}\big)\big)_{s=n}^{L-1}\Big)^\top\Big\|_\infty \\
\le &\Big\|\mathrm{softmax}\Big(\big(\hat{g}_K\big(\hat{X}_{L-n+2:L}; \hat{X}_{s-n+1:s-1}\big)\big)_{s=n}^{L-1}\Big) - \mathrm{softmax}\Big(\big(g\big(X_{L-n+2:L}; X_{s-n+1:s-1}\big)\big)_{s=n}^{L-1}\Big)\Big\|_\infty \\
\le &\Big\|\mathrm{softmax}\Big(\big(\hat{g}_K\big(\hat{X}_{L-n+2:L}; \hat{X}_{s-n+1:s-1}\big)\big)_{s=n}^{L-1}\Big) - \mathrm{softmax}\Big(\big(g_K\big(\hat{X}_{L-n+2:L}; \hat{X}_{s-n+1:s-1}\big)\big)_{s=n}^{L-1}\Big)\Big\|_\infty \\
&+ \Big\|\mathrm{softmax}\Big(\big(g_K\big(\hat{X}_{L-n+2:L}; \hat{X}_{s-n+1:s-1}\big)\big)_{s=n}^{L-1}\Big) - \mathrm{softmax}\Big(\big(g_K\big(X_{L-n+2:L}; X_{s-n+1:s-1}\big)\big)_{s=n}^{L-1}\Big)\Big\|_\infty \\
&+ \Big\|\mathrm{softmax}\Big(\big(g_K\big(X_{L-n+2:L}; X_{s-n+1:s-1}\big)\big)_{s=n}^{L-1}\Big) - \mathrm{softmax}\Big(\big(g\big(X_{L-n+2:L}; X_{s-n+1:s-1}\big)\big)_{s=n}^{L-1}\Big)\Big\|_\infty \\
\le &\max_s \Big|\hat{g}_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big) - g_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big)\Big| \\
&+ \max_s \Big|g_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big) - g_K\big(X_{L-n+2:L}, X_{s-n+1:s-1}\big)\Big| \\
&+ \sum_{s=n}^{L-1} \big|g_K\big(X_{L-n+2:L}, X_{s-n+1:s-1}\big) - g\big(X_{L-n+2:L}, X_{s-n+1:s-1}\big)\big|
\end{aligned} \tag{17}$$

For the first term in RHS of (17), it holds that:

$$
\begin{aligned}
&\max_s \left|\hat{g}_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big) - g_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big)\right| \\
\leq &\max_s \sum_{k=1}^{K} \sigma_k \left|\hat{\phi}_k(\hat{X}_{L-n+2:L})\hat{\psi}_k(\hat{X}_{s-n+1:s-1}) - \phi_k(\hat{X}_{L-n+2:L})\psi_k(\hat{X}_{s-n+1:s-1})\right| \\
\leq &\sum_{k=1}^{K} \sigma_k \left(\left\|\hat{\phi}_k\right\|_{L^\infty} \left\|\hat{\psi}_k - \psi_k\right\|_{L^\infty} + \|\psi_k\|_{L^\infty} \left\|\hat{\phi}_k - \phi_k\right\|_{L^\infty}\right) \\
\leq &\epsilon^{(1)}_{\mathsf{FFN}} \cdot \sum_{k=1}^{K} \sigma_k \left(\left\|\hat{\phi}_k\right\|_{L^\infty} + \|\psi_k\|_{L^\infty}\right) \\
\leq &\epsilon^{(1)}_{\mathsf{FFN}} \cdot \sum_{k=1}^{K} \sigma_k \left(\|\phi_k\|_{L^\infty} + \left\|\hat{\phi}_k - \phi_k\right\|_{L^\infty} + \|\psi_k\|_{L^\infty}\right) \\
\leq &\epsilon^{(1)}_{\mathsf{FFN}} \cdot (2C_g^\infty + 1) \sum_{k=1}^{K} \sigma_k \leq (2C_g^\infty + 1) C_\alpha \epsilon^{(1)}_{\mathsf{FFN}}.
\end{aligned}
$$

For the second term in RHS of (17), we have:

$$
\begin{aligned}
&\max_s \left|g_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big) - g_K\big(X_{L-n+2:L}, X_{s-n+1:s-1}\big)\right| \\
\leq &\max_s \sum_{k=1}^{K} \sigma_k \Big(\|\phi_k\|_{L^\infty} |\psi_k(\hat{X}_{s-n+1:s-1}) - \hat{\psi}_k(X_{s-n+1:s-1})| \\
&\qquad + \|\psi_k\|_{L^\infty} |\phi_k(\hat{X}_{L-n+1:L-1}) - \phi_k(X_{L-n+1:L-1})|\Big) \\
\leq &\max_s \sum_{k=1}^{K} \sigma_k \Big(\|\phi_k\|_{L^\infty} \|\psi_k\|_{\mathrm{Lip}} \left\|\hat{X}_{s-n+1:s-1} - X_{s-n+1:s-1}\right\| \\
&\qquad + \|\psi_k\|_{L^\infty} \|\phi_k\|_{\mathrm{Lip}} \left\|\hat{X}_{L-n+1:L-1} - X_{L-n+1:L-1}\right\|\Big) \\
\leq &2C_g^\infty C_g^{\mathrm{Lip}} \epsilon^{(1)}_{\mathsf{SA}} \cdot \left(\max_s \sum_{k=1}^{K} \sigma_k\right) \leq 2C_g^\infty C_g^{\mathrm{Lip}} C_\alpha \epsilon^{(1)}_{\mathsf{SA}}.
\end{aligned}
$$

Additioanlly, the third term in RHS of (17), its L^2 holds that:

$$
\begin{aligned}
&\int_{[0,1]^{d\times L}} \left(\sum_{s=n}^{L-1} \left|g_K\big(X_{L-n+2:L}, X_{s-n+1:s-1}\big) - g\big(X_{L-n+2:L}, X_{s-n+1:s-1}\big)\right|\right)^2 dX \\
\leq &(t-1-n) \sum_{s=n}^{L-1} \int_{[0,1]^{D\times L}} \left|g_K\big(X_{-n+2:t}, X_{s-n+1:s-1}\big) - g\big(X_{L-n+2:L}, X_{s-n+1:s-1}\big)\right|^2 dX \\
= &(L-1-n)^2 \int_{[0,1]^D \times [0,1]^D} |g(u,v) - g_K(u;v)|^2 \,\mathrm{d}u\mathrm{d}v \\
= &(L-1-n)^2 \int \left(\sum_{k=K+1}^{+\infty} \sigma_k \phi_k(u)\psi_k(v)\right)^2 du\, dv \\
\leq &\int \left(\sum_{k=K+1}^{+\infty} \sigma_k \phi_k^2(u)\right)\left(\sum_{k=K+1}^{+\infty} \sigma_k \psi_k^2(v)\right) du\, dv \\
\leq &(L-1-n)^2 \left(\sum_{k=K+1}^{\infty} \sigma_k\right)^2 \leq \frac{(L-1-n)^2 C_\alpha^2}{K^{2\alpha}}.
\end{aligned}
$$

Now we combine three error terms together to obtain the total L^2 error for the output of this layer:

$$
\begin{aligned}
&\int_{X\in[0,1]^{d\times L}} \left\| x_L^{(2)} - (x_s)_{s=n}^{L-1} \,\mathrm{softmax}\Big(\big(g\big(X_{L-n+2:L}; X_{s-n+1:s-1}\big)\big)_{s=n}^{L-1} \Big)^\top \right\|_\infty^2 \mathrm{d}X \\
\leq & 3\int_{X\in[0,1]^{d\times L}} \max_s \left| \hat{g}_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big) - g_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big) \right|^2 \mathrm{d}X \\
& + 3\int_{X\in[0,1]^{d\times L}} \max_s \left| g_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big) - g_K\big(X_{L-n+2:L}, X_{s-n+1:s-1}\big) \right|^2 \mathrm{d}X \\
& + 3\int_{X\in[0,1]^{d\times L}} \left(\sum_{s=n}^{L-1} \left| g_K\big(X_{L-n+2:L}, X_{s-n+1:s-1}\big) - g\big(X_{L-n+2:L}, X_{s-n+1:s-1}\big) \right| \right)^2 \mathrm{d}X \\
\leq & 3\max_s \left| \hat{g}_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big) - g_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big) \right|^2 \\
& + 3\max_s \left| g_K\big(\hat{X}_{L-n+2:L}, \hat{X}_{s-n+1:s-1}\big) - g_K\big(X_{L-n+2:L}, X_{s-n+1:s-1}\big) \right|^2 \\
& + 3\left(\frac{(L-1-n)C_\alpha}{K^\alpha} \right)^2 \\
\leq & 3\left((2C_g^\infty + 1)C_\alpha \epsilon_{\mathsf{FFN}}^{(1)} \right)^2 + 3\left(2C_g^\infty C_g^{\mathrm{Lip}} C_\alpha \epsilon_{\mathsf{SA}}^{(1)} \right)^2 + 3\left(\frac{(L-1-n)C_\alpha}{K^\alpha} \right)^2 \\
\leq & 3\left((2C_g^\infty + 1)C_\alpha \epsilon_{\mathsf{FFN}}^{(1)} + 2C_g^\infty C_g^{\mathrm{Lip}} C_\alpha \epsilon_{\mathsf{SA}}^{(1)} + \frac{(L-1-n)C_\alpha}{K^\alpha} \right)^2 .
\end{aligned}
$$

This estimate implies that

$$
\begin{aligned}
& \|\mathsf{GIH}_n - \mathsf{TF}\|_{L,2} \\
\leq & \sqrt{3}\left(2C_g^\infty C_g^{\mathrm{Lip}} C_\alpha \epsilon_{\mathsf{SA}}^{(1)} + (2C_g^\infty + 1)C_\alpha \epsilon_{\mathsf{FFN}}^{(1)} + \frac{(L-1-n)C_\alpha}{K^\alpha} \right) \\
\leq & \mathcal{O}\left(\frac{C_{g,q,n}}{H^n} \right) + \tilde{\mathcal{O}}\left(\frac{C_{g,\alpha}\sqrt{K}}{\sqrt{M}} \right) + \mathcal{O}\left(\frac{tC_\alpha}{K^\alpha} \right)
\end{aligned}
\tag{18}
$$

Step IV. Optimizing K in (18).

Notice that in RHS of (18), only $\tilde{\mathcal{O}}\left(\frac{C_{g,\alpha}\sqrt{K}}{\sqrt{M}}\right)$ and $\mathcal{O}\left(\frac{LC_\alpha}{K^\alpha}\right)$ depend on K.

By Young's inequality, with $p = \frac{\alpha+\frac{1}{2}}{\alpha}$ and $q = 2(\alpha+\frac{1}{2})$, we have:

$$
\begin{aligned}
& \min_K : \frac{\alpha}{\frac{1}{2}+\alpha} \frac{C_{g,\alpha}\sqrt{K}}{\sqrt{M}} + \frac{\frac{1}{2}}{\frac{1}{2}+\alpha} \frac{LC_\alpha}{K^\alpha} \\
= & \min_K : \frac{\alpha}{\frac{1}{2}+\alpha} \left(\left(\frac{C_{g,\alpha}\sqrt{K}}{\sqrt{M}} \right)^{\frac{\alpha}{\frac{1}{2}+\alpha}} \right)^{\frac{\frac{1}{2}+\alpha}{\alpha}} + \frac{\frac{1}{2}}{\frac{1}{2}+\alpha} \left(\left(\frac{LC_\alpha}{K^\alpha} \right)^{\frac{\frac{1}{2}}{\frac{1}{2}+\alpha}} \right)^{2(\frac{1}{2}+\alpha)} \\
= & \frac{C'_{g,\alpha} L^{1/(1+2\alpha)}}{M^{\alpha/(1+2\alpha)}} .
\end{aligned}
$$

Thus, we obtain our final bound:

$$
\|\mathsf{GIH}_n - \mathsf{TF}\|_{L,2} \leq \mathcal{O}\left(\frac{C_{g,q,n}}{H^n} \right) + \left\{ \tilde{\mathcal{O}}\left(\frac{C_{g,\alpha}\sqrt{K}}{\sqrt{M}} \right) + \mathcal{O}\left(\frac{LC_\alpha}{K^\alpha} \right) \right\}_{\min:K}
$$

$$\leq \mathcal{O}\left(\frac{C_{g,q,n}}{H^n}\right) + \tilde{\mathcal{O}}\left(\frac{C'_{g,\alpha}L^{1/(1+2\alpha)}}{M^{\alpha/(1+2\alpha)}}\right) \leq \frac{A_{g,q,n}}{H^n} + \frac{B_{g,\alpha}L^{1/(1+2\alpha)}}{M^{\alpha/(1+3\alpha)}}.$$

□

B Proofs of Optimization Dynamics: Training Stage I

In this subsection we focus on training the first layer of Transformer model to capture the token ahead. For simplicity, we introduce some notations:

$$\tilde{p} := p^{(1,1)}, \quad p := p^{(2,1)}, \quad g := w_V^{(2,1)}, \quad h := w_V^{(2,2)}, \quad w_K := w_K^{(2,2)}, \quad w_Q := w_Q^{(2,2)},$$

and denote the initialization of each parameter as $\tilde{p}(0), p(0), g(0), w_Q(0), w_K(0), h(0)$ respectively.

We initialize $p(0), w_k(0), w_Q(0) = 0$ while the other parameters are all initialized at σ_{init}. In this training stage, we only train $\tilde{p}$. And our goal, **the proof of Lemma 5.3** can be deduced from which, is to prove:

$$\lim_{t\to+\infty} \tilde{p}(t) = +\infty.$$

In this stage, the s-th output token of the first layer is represented as

$$\begin{pmatrix} x_s \\ (x_\tau)_{\tau=1}^{s-1}\ \mathrm{softmax}\left(\left(-\tilde{p}(s-1-\tau)\right)_{\tau=1}^{s-1}\right)^\top \end{pmatrix},$$

and the target function and output of transformer are as follows

$$f^*(X) = \begin{pmatrix} \frac{\alpha^\star}{1+\alpha^\star} x_{L-2} \\ \frac{1}{1+\alpha^\star}(x_s)_{s=2}^{L-1}\ \mathrm{softmax}\left(\left(x_L w^{\star 2} x_{s-1}\right)_{s=2}^{L-1}\right)^\top \end{pmatrix},$$

$$f_\theta(X) = \begin{pmatrix} g(0)\left(\sum_{\tau=1}^{s-1}\mathrm{softmax}_s(-\tilde{p}(s-1-\tau))x_\tau\right)_{s=2}^{L-1}\mathrm{softmax}\left(-p(0)(L-1-s)_{s=2}^{L-1}\right) \\ h(0)(x_s)_{s=2}^{L-2}\cdot \mathrm{softmax}\left(\left(w_K(0)w_Q(0)x_L\cdot(x_\tau)_{\tau=1}^{s-1}\mathrm{softmax}\left(\left(-\tilde{p}(s-1-\tau)\right)_{\tau=1}^{s-1}\right)^\top\right)_{s=2}^{L-2}\right) \end{pmatrix}$$

$$= \begin{pmatrix} g(0)\frac{1}{L-2}\sum_{\tau=1}^{L-2}\left(\sum_{s=\tau+1}^{L-1}\mathrm{softmax}\left(-\tilde{p}(s-1-t)_{t=1}^{s-1}\right)_{t=\tau}\right)x_\tau \\ h(0)\frac{1}{L-2}\sum_{s=2}^{L-2}x_s \end{pmatrix}.$$

Since we only focus on $\tilde{p}$ and the other parameters remain the initialization value, the loss function can be simplified as

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{X\sim\mathbb{N}(0,1)^L}\left[\frac{\alpha^{\star 2}}{(1+\alpha^\star)^2}x_{L-2}^2 + \frac{g(0)^2}{(L-2)^2}\sum_{\tau=1}^{L-2}\left(\sum_{s=\tau+1}^{L-1}\mathrm{softmax}\left(-\tilde{p}(s-1-t)_{t=1}^{s-1}\right)_{t=\tau}\right)^2 x_\tau^2 \right.$$
$$\left. + \frac{2g(0)}{L-2}\frac{\alpha^\star}{1+\alpha^\star}\mathrm{softmax}\left(-p(0)(L-1-s)_{s=2}^{L-1}\right)_{s=L-1}x_{L-2}^2\right] + C(w^\star,\alpha^\star,w(0),h(0))$$

where the second term $C(w^\star,\alpha^\star,w(0),h(0))$ is a constant depends on $w^\star,\alpha^\star,w(0)$ and $h(0)$, produced by calculating the error of the second head, i.e., loss of induction head, while the first term is 4-gram loss.

We first define several functions that will be useful for calculation in this stage and the second one:

Function I. This function is purely defined for the calculation of $\frac{\mathrm{d}p}{\mathrm{d}t}$. Denoted by $q(\tilde{p}) := \sum_{\tau=1}^{L-2}\left(\sum_{s=\tau+1}^{L-1}\frac{e^{\tilde{p}(s-1-\tau)}}{\sum_{k=0}^{s-2}e^{-\tilde{p}k}}\right)^2$, we first prove $q'(\tilde{p}) \le 0$.

$$\begin{aligned}
q(\tilde{p}) &:= \sum_{\tau=1}^{L-2}\left(\sum_{s=\tau+1}^{L-1}\frac{e^{\tilde{p}(s-1-\tau)}}{\sum_{k=0}^{s-2}e^{-\tilde{p}k}}\right)^2 \\
&= \sum_{\tau=1}^{L-2}\left(\sum_{s=\tau+1}^{L-1}\frac{e^{-\tilde{p}(s-1-\tau)}}{1-e^{-\tilde{p}(s-1)}}(1-e^{-\tilde{p}})\right)^2 \\
&= (1-e^{-\tilde{p}})^2\sum_{\tau=1}^{L-2}\left(\sum_{s=\tau+1}^{L-1}\frac{e^{-\tilde{p}(s-1-\tau)}}{1-e^{-\tilde{p}(s-1)}}\right)^2 \\
&= (1-e^{-\tilde{p}})^2\sum_{\tau=1}^{L-2}e^{2\tilde{p}\tau}\left(\sum_{s=\tau+1}^{L-1}\frac{e^{-\tilde{p}(s-1)}}{1-e^{-\tilde{p}(s-1)}}\right)^2 \\
&= (1-e^{-\tilde{p}})^2\sum_{\tau=1}^{L-2}e^{2\tilde{p}\tau}\left(\sum_{s=\tau+1}^{L-1}\frac{1}{e^{\tilde{p}(s-1)}-1}\right)^2
\end{aligned}$$

Then we take its derivative of $\tilde{p}$

$$\begin{aligned}
q'(\tilde{p}) &= 2(1-e^{-\tilde{p}})e^{-\tilde{p}}\sum_{\tau=1}^{L-2}e^{2\tilde{p}\tau}\left(\sum_{s=\tau+1}^{L-1}\frac{1}{e^{\tilde{p}(s-1)}-1}\right)^2 \\
&\quad + (1-e^{-\tilde{p}})^2\sum_{\tau=1}^{L-2}2\tau e^{2\tilde{p}\tau}\left(\sum_{s=\tau+1}^{L-1}\frac{1}{e^{\tilde{p}(s-1)}-1}\right)^2 \\
&\quad + (1-e^{-\tilde{p}})^2\sum_{\tau=1}^{L-2}2e^{2\tilde{p}\tau}\left(\sum_{s=\tau+1}^{L-1}\frac{1}{e^{\tilde{p}(s-1)}-1}\right)\left(\sum_{s=\tau+1}^{L-1}\frac{-(s-1)e^{\tilde{p}(s-1)}}{(e^{\tilde{p}(s-1)}-1)^2}\right) \\
&= 2(1-e^{-\tilde{p}})\sum_{\tau=1}^{L-2}e^{2\tilde{p}\tau}\left(\sum_{s=\tau+1}^{L-1}\frac{1}{e^{\tilde{p}(s-1)}-1}\right)\left(\sum_{s=\tau+1}^{L-1}\frac{e^{-\tilde{p}}+\tau(1-e^{-\tilde{p}})}{e^{\tilde{p}(s-1)}-1}-\frac{(s-1)e^{\tilde{p}(s-1)}}{(e^{\tilde{p}(s-1)}-1)^2}\right)
\end{aligned}$$

$q'(\tilde{p})$'s last factor can be formed as

$$\begin{aligned}
&\frac{\left(\tau-(\tau-1)e^{-\tilde{p}}\right)\left(e^{\tilde{p}(s-1)}-1\right)-(s-1)e^{\tilde{p}(s-1)}}{e^{\tilde{p}(s-1)}-1)^2} \\
=&\frac{(\tau+1-s)t^{s-1}-(\tau-1)t^{s-2}-\tau+\frac{\tau-1}{t}}{e^{\tilde{p}(s-1)}-1)^2}
\end{aligned}$$

where $t = e^{-\tilde{p}} \ge 1$. Since $s \ge \tau+1$, $q'(\tilde{p}) \le 0$.

Function II. For simplicity, we define $M(p)$ and its derivative $m(p)$:

$$M(p) := \sum_{s=2}^{L-1}\exp(-p(L-1-s)) = \sum_{s=0}^{L-3}\exp -ps = \frac{1-e^{-p(L-2)}}{1-e^{-p}},$$

$$m(p) := \sum_{s=1}^{l-3}s\exp(-ps) = \frac{e^{-p}-(L-2)e^{-p(L-2)}+(L-3)e^{-p(L-1)}}{(1-e^{-p})^2}.$$

Function III. The third function is derivative of softmax. By straightfoward calculation, we obtain:

$$\frac{\mathrm{d}}{\mathrm{d}p}\mathrm{softmax}\left(-p(L-1-t)_{t=2}^{L-1}\right)_{t=L-1-s} = \frac{\mathrm{d}}{\mathrm{d}p}\frac{\exp(-ps)}{\sum_{\tau=0}^{L-3}\exp(-p\tau)} = \frac{-s\exp(-ps)M(p)+\exp(-ps)m(p)}{M(p)^2}.$$

Through the quantities and their properties above, we obtain the dynamic of $\tilde{p}$

$$\begin{aligned}\frac{\mathrm{d}\tilde{p}}{\mathrm{d}t} &= -\frac{g(0)^2}{(L-2)^2}q'(\tilde{p}) + \frac{2\alpha^\star g(0)}{(1+\alpha^\star)(L-2)}\frac{m(p)}{M(p)^2} \\ &\geq \frac{2\alpha^\star g(0)}{(1+\alpha^\star)(L-2)}e^{-\tilde{p}},\end{aligned}$$

which implies:

$$\lim_{t\to+\infty}\tilde{p}(t) = +\infty.$$

C Proofs of Optimization Dynamics: Training Stage II

In this training stage, the first layer is already capable of capturing the token ahead i.e. $y_s = x_{s-1}$. And we train the parameters $w_{V_1}, w_{V_2}, p, w_{KQ}$ in the second layer.

We start from proving the parameter balance lemma:

Lemma C.1 (Restate of Lemma 5.4)**.** *In Training Stage II, it holds that* $w_Q^{(2,2)^2}(t) \equiv w_K^{(2,2)^2}(t)$.

Proof. Notice that

$$\begin{aligned}&\frac{\mathrm{d}}{2\mathrm{d}t}\left(w_Q^{(2,2)^2}(t) - w_K^{(2,2)^2}(t)\right) = -w_Q^{(2,2)}\frac{\partial\mathcal{L}}{\partial w_Q^{(2,2)}} + w_K^{(2,2)}\frac{\partial\mathcal{L}}{\partial w_K^{(2,2)}} \\ =& -w_Q^{(2,2)}w_K^{(2,2)}\frac{\partial\mathcal{L}}{\partial\left(w_Q^{(2,2)}w_K^{(2,2)}\right)} + w_K^{(2,2)}w_Q^{(2,2)}\frac{\partial\mathcal{L}}{\partial\left(w_Q^{(2,2)}w_K^{(2,2)}\right)} \equiv 0.\end{aligned}$$

Thus, we have:

$$w_Q^{(2,2)^2}(t) - w_K^{(2,2)^2}(t) \equiv w_Q^{(2,2)^2}(0) - w_K^{(2,2)^2}(0) = 0.$$

□

For simplicity, we still use the following notations:

$$p := p_1, \quad g := w_{V_1}, \quad w := w_{KQ}, \quad h := w_{V_2}.$$

and notations for initialization $p(0), g(0), w(0), h(0)$. Then the target function and output of Transformer can be formed as follows

$$\begin{aligned}f^\star(X) &= \begin{pmatrix}\frac{\alpha^\star}{1+\alpha^\star}x_{L-2} \\ \frac{1}{1+\alpha^\star}(x_s)_{s=2}^{L-1}\cdot\mathrm{softmax}\left(\left(w^{\star 2}x_Lx_{s-1}\right)_{s=2}^{L-1}\right)\end{pmatrix}, \\ \mathsf{TF}(X;\theta) &= \begin{pmatrix}g\cdot(x_{s-1})_{s=2}^{L-2}\cdot\mathrm{softmax}\left((-p(L-1-s))_{s=2}^{L-2}\right) \\ h\cdot(x_s)_{s=2}^{L-2}\cdot\mathrm{softmax}\left(\left(w^2x_Lx_{s-1}\right)_{s=2}^{L-2}\right)\end{pmatrix}.\end{aligned}$$

And the loss function is expressed as:

$$\begin{aligned}&\mathcal{L}(\theta) = \frac{1}{2}\mathop{\mathbb{E}}_{X\sim\mathbb{N}(0,1)^L}\left[\|f^\star(x) - \mathsf{TF}(x;\theta)\|^2\right] \\ &= \frac{1}{2}\mathbb{E}_X\left[\left(\frac{\alpha^\star}{1+\alpha^\star}x_{L-2} - g\cdot(x_{s-1})_{s=2}^{L-2}\cdot\mathrm{softmax}\left((-p(L-1-s))_{s=2}^{L-2}\right)\right)^2\right] \\ &\quad + \frac{1}{2}\mathbb{E}_X\left[\left(\frac{1}{1+\alpha^\star}(x_s)_{s=2}^{L-1}\cdot\mathrm{softmax}\left(\left(w^{\star 2}x_Lx_{s-1}\right)_{s=2}^{L-1}\right) - h\cdot(x_s)_{s=2}^{L-2}\cdot\mathrm{softmax}\left(\left(w^2x_Lx_{s-1}\right)_{s=2}^{L-2}\right)\right)^2\right].\end{aligned}$$

The total loss can naturally be divided into two parts:

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathsf{G}_4}(\theta) + \mathcal{L}_{\mathsf{IH}_2}(\theta),$$

where

$$\begin{aligned}\mathcal{L}_{\mathsf{G}_4}(\theta) &= \mathcal{L}_{\mathsf{G}_4}(p, g) \\ &= \frac{1}{2}\mathbb{E}_X\left[\left(\frac{\alpha^\star}{1+\alpha^\star}x_{L-2} - g\cdot(x_{s-1})_{s=2}^{L-2}\cdot\mathrm{softmax}\Big((-p(L-1-s))_{s=2}^{L-2}\Big)\right)^2\right],\end{aligned}$$

$$\begin{aligned}\mathcal{L}_{\mathsf{IH}_2}(\theta) &= \mathcal{L}_{\mathsf{IH}_2}(w, h) \\ &= \frac{1}{2}\mathbb{E}_X\left[\left(\frac{1}{1+\alpha^\star}(x_s)_{s=2}^{L-1}\cdot\mathrm{softmax}\Big(\left(w^{\star 2}x_L x_{s-1}\right)_{s=2}^{L-1}\Big) - h\cdot(x_s)_{s=2}^{L-2}\cdot\mathrm{softmax}\Big(\left(w^2 x_L x_{s-1}\right)_{s=2}^{L-2}\Big)\right)^2\right].\end{aligned}$$

Notably, the dynamics of (p, g) and (w, h) are **decoupled**, which allows us to analyze them separately.

Additionally, we denote the optimal values of the parameters as:

$$p^\star = +\infty, \quad g^\star = \frac{\alpha^\star}{1+\alpha^\star}, \quad w^\star := w^\star, \quad h^\star = \frac{1}{1+\alpha^\star}.$$

For the initialization scale and the sequence length, we consider the case:

$$\sigma_{\text{init}} = \mathcal{O}(1) \ll 1, \quad L = \Omega(1/\sigma_{\text{init}}) \gg 1.$$

C.1 Dynamics of the parameters for 4-gram

First, we define two useful auxiliary functions:

$$\begin{aligned}M(p) &:= \frac{1-e^{-p(L-2)}}{1-e^{-p}}, \\ m(p) &:= \frac{e^{-p} - (L-2)e^{-p(L-2)} + (L-3)e^{-p(L-1)}}{(1-e^{-p})^2}.\end{aligned}$$

Then, a straightforward calculation, combined with Lemma D.3 and Lemma D.4, yields the explicit formulation of $\mathcal{L}_{\mathsf{G}_4}(\theta)$ and the GF dynamics of p and g:

$$\mathcal{L}_{\mathsf{G}_4}(\theta) = \frac{1}{2}\left(\frac{\alpha^\star}{1+\alpha^\star}\right)^2 + \frac{1}{2}g^2\frac{M(2p)}{M(p)^2} - \frac{\alpha^\star g}{1+\alpha^\star}\frac{1}{M(p)}. \tag{19}$$

$$\begin{aligned}\frac{\mathrm{d}p}{\mathrm{d}t} &= -\frac{\partial\mathcal{L}}{\partial p} = -\frac{\partial\mathcal{L}_{\mathsf{G}_4}}{\partial p} = \frac{m(p)}{M(p)^2}\left[g^2\frac{m(2p)}{m(p)} - g^2\frac{M(2p)}{M(p)} + \frac{\alpha^\star g}{1+\alpha^\star}\right], \\ \frac{\mathrm{d}g}{\mathrm{d}t} &= -\frac{\partial\mathcal{L}}{\partial g} = -\frac{\partial\mathcal{L}_{\mathsf{G}_4}}{\partial g} = \frac{\alpha^\star}{1+\alpha^\star}\frac{1}{M(p)} - g\frac{M(2p)}{M(p)^2},\end{aligned}$$

Equivalently, the dynamics can be written as:

$$\begin{aligned}\frac{\mathrm{d}p}{\mathrm{d}t} &= \frac{m(p)g}{M(p)^2}\left(g^\star - g\frac{M(2p)}{M(p)} + g\frac{m(2p)}{m(p)}\right), \\ \frac{\mathrm{d}g}{\mathrm{d}t} &= \frac{1}{M(p)}\left(g^\star - g\frac{M(2p)}{M(p)}\right).\end{aligned}$$

Notice that at the initialization, it holds that $\frac{\mathrm{d}p}{\mathrm{d}t}|_{t=0} > 0$ and $\frac{\mathrm{d}g}{\mathrm{d}t}|_{t=0} > 0$. Then we first define a hitting time:

$$T_1^g := \inf\{t > 0 : g(t) > g^\star\}.$$

Noticing $g(0) = \sigma_{\text{init}} \ll g^\star$ and the continuity, $T_1^g > 0$.

Our subsequent proof can be divided into **two phases**: a monotonic phase $t < T_1^g$, and a stable convergence phase $t > T_1^g$.

Part I. Analysis for the monotonic phase $t < T_1^g$.

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \frac{m(p)g}{M(p)^2}\left(g^\star - g\frac{M(2p)}{M(p)} + g\frac{m(2p)}{m(p)}\right) = \frac{m(p)g}{M(p)^2}\left(g^\star - g\frac{1+e^{-p(L-2)}}{1+e^{-p}} + g\frac{m(2p)}{m(p)}\right),$$
$$\frac{\mathrm{d}g}{\mathrm{d}t} = \frac{1}{M(p)}\left(g^\star - g\frac{M(2p)}{M(p)}\right) = \frac{1}{M(p)}\left(g^\star - g\frac{1+e^{-p(L-2)}}{1+e^{-p}}\right).$$

It is easy to see that p, g are monotonically increasing for $t < T_1^g$. We can choose sufficiently large

$$L = \Omega(1/p(0)) = \Omega(1/\sigma_{\text{init}})$$

such that:

$$(L-3)e^{-(L-3)p(t)},\ e^{-(L-5)p(t)} < 0.0001, \quad \forall p > \sigma_{\text{init}}.$$

Then we can calculate the following three terms in the dynamics:

$$\frac{m(p)}{M^2(p)} = \frac{e^{-p}\left(1-(L-2)e^{-p(L-3)}+(L-3)e^{-p(L-2)}\right)}{1-e^{-p(L-2)}} = \frac{e^{-p}(1+\xi_1(p))}{1+\xi_2(p)},$$
$$\frac{1}{M(p)} = \frac{1-e^{-p(L-2)}}{1-e^{-p}} = \frac{1+\xi_3(p)}{1-e^{-p}},$$

$$\frac{m(2p)}{m(p)} = \frac{e^{-p}\left(1-(L-2)e^{-2p(L-3)}+(L-3)e^{-2p(L-2)}\right)}{(1+e^{-p})^2\left(1-(L-2)e^{-p(L-3)}+(L-3)e^{-p(L-2)}\right)}$$
$$= \frac{e^{-p}(1+\xi_4(p))}{(1+e^{-p})^2(1+\xi_5(p))},$$

where the error functions satisfy:

$$|\xi_1(p)|, \cdots, |\xi_5(p)| \le 0.0001,\ \forall t > T_1^g.$$

Then the dynamics satisfy:

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \frac{e^{-p}g(1+\xi_1(p))}{1+\xi_2(p)}\left(g^\star - g\frac{1+e^{-p(L-2)}}{1+e^{-p}} + \frac{ge^{-p}(1+\xi_3(t))}{(1+e^{-p})^2(1+\xi_5(t))}\right),$$
$$\frac{\mathrm{d}g}{\mathrm{d}t} = \frac{1+\xi_3(p)}{1-e^{-p}}\left(g^\star - g\frac{1+e^{-p(L-2)}}{1+e^{-p}}\right).$$

When $g < \frac{1}{2}\frac{\alpha^\star}{1+\alpha^\star}$, we have

$$\frac{\mathrm{d}p}{\mathrm{d}g} \le 2\left(e^{-p}-e^{-2p}\right)g.$$

By define $T_{1/2}^g := \inf\{t > 0 : g(t) > g^\star/2\}$ and $\tilde{p} := p(T_{1/2}^g)$, we have

$$\ln(e^{\tilde{p}}-1) \le \frac{1}{4}g^{\star 2} - g(0)^2 + e^{p(0)} - 1 + \ln(e^{p(0)}-1)$$

then $\tilde{p} \le \mathcal{O}(\sqrt{p(0)})$, from which we infer that p barely increases when $t \le T_{1/2}^g$.

For $0 \le t \le T_{1/2}^g$,

$$\frac{\mathrm{d}g}{\mathrm{d}t} \ge \frac{1}{1-e^{-p(0)}}\left[g^\star - \frac{g}{1+e^{-p(0)}}\right]$$

$$g \geq g^\star(1+e^{-p(0)})) + \left[g(0) - g^\star(1+e^{-p(0)})\right] \exp\left(\frac{-t}{1-e^{-2p(0)}}\right)$$

so

$$T_{1/2}^g \leq (1-e^{-2p(0)}) \ln\left(\frac{g^\star(1+e^{-p(0)}) - g(0)}{g^\star\left((1+e^{-p(0)}) - \frac{1}{2}\right)}\right) = \mathcal{O}\left(2p(0)\right)$$

For $T_{1/2}^g \leq t \leq T_1^g$, let $p_1 := p(T_1^g)$,

$$\begin{aligned}\frac{\mathrm{d}p}{\mathrm{d}g} &\leq 1.01e^{-p}(1-e^{-p})g\left(1 + \frac{\frac{g}{1+e^{-p}} - \frac{g}{(1+e^{-p})^2}}{\frac{\alpha^\star}{1+\alpha^\star} - \frac{g}{1+e^{-p}}}\right) \\ &\leq \frac{1.01}{4}\frac{\alpha^\star}{1+\alpha^\star}(1+e^{-p_1})\end{aligned}$$

then

$$\begin{aligned}p_1 - p(0) &\leq \frac{1.01}{4}\left(\frac{\alpha^\star}{1+\alpha^\star}\right)^2 (1+e^{p_1}), \\ p_1 &\leq \frac{1}{2\left(\frac{\alpha^\star}{1+\alpha^\star}\right)^2 - 1},\end{aligned}$$

and we take $\alpha^\star > 1$.

Since for $T_{1/2}^g \leq t \leq T_1^g$,

$$\begin{aligned}\frac{\mathrm{d}p}{\mathrm{d}t} &\leq 2e^{-p}g^\star\left(g^\star - \frac{1}{8}g^\star\right), \\ \frac{\mathrm{d}p}{\mathrm{d}t} &\geq \frac{1}{2}e^{-p}g^\star\left(g^\star - \frac{1}{1+e^{-p_1}}g^\star\right),\end{aligned}$$

we have

$$T_1^g - t_1 \leq \mathcal{O}\left((e^{2p_1}-1)\left(\frac{1+\alpha^\star}{\alpha^\star}\right)^2\right).$$

Hence, putting the two part of time together we have

$$\begin{aligned}T_1^g &\leq \mathcal{O}\left(p(0) + (e^{2p_1}-1)\left(\frac{1+\alpha^\star}{\alpha^\star}\right)^2\right) \\ &= \mathcal{O}\left(\sigma_{\text{init}} + (e^{2p_1}-1)\left(\frac{1+\alpha^\star}{\alpha^\star}\right)^2\right) = \mathcal{O}(1).\end{aligned} \tag{20}$$

Part II. Analysis for the convergence phase $t > T_1^g$.

We will prove that, in this phase, (p, g) keep in a stable region, and the convergence occurs.

Recall the dynamics:

$$\begin{aligned}\frac{\mathrm{d}p}{\mathrm{d}t} &= \frac{m(p)g}{M(p)^2}\left(g^\star - g\frac{1+e^{-p(L-2)}}{1+e^{-p}} + g\frac{m(2p)}{m(p)}\right), \\ \frac{\mathrm{d}g}{\mathrm{d}t} &= \frac{1}{M(p)}\left(g^\star - g\frac{1+e^{-p(L-2)}}{1+e^{-p}}\right).\end{aligned}$$

Using contradiction, it is easy to verify that for all $t > T_1^g$,

$$g^\star < g(t) < 2g^\star, \quad \frac{\mathrm{d}p(t)}{\mathrm{d}t} > 0,$$

which means g has entered a stable region (although it is possible that g is non-monotonic), while p keeps increase. In fact, if $\tilde{t} := \inf\{t > 0 : g(t) = 2g^\star\}$, then $\frac{\mathrm{d}g}{\mathrm{d}t}|_{\tilde{t}} < 0$, which leads to a contradiction. If $\hat{t} := \inf\{t > 0 : \frac{\mathrm{d}p(t)}{\mathrm{d}t} = 0\}$, then

$$\left(g^\star - g\frac{1+e^{-p(L-2)}}{1+e^{-p}} + g\frac{m(2p)}{m(p)}\right)\Big|_{\hat{t}} = 0, \quad \frac{\mathrm{d}g}{\mathrm{d}t} < 0,$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(g^\star - g\frac{1+e^{-p(L-2)}}{1+e^{-p}} + g\frac{m(2p)}{m(p)}\right) = -g'\frac{1+e^{-p(L-2)}}{1+e^{-p}} + g'\frac{m(2p)}{m(p)} > 0,$$

where the last inequality leads to a contradiction.

Thus, $p(t) > p(T_1^g) > p(0) = \sigma_{\text{init}}$ holds in this phase. Therefore, the dynamics

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \frac{e^{-p}g(1+\xi_1(p))}{1+\xi_2(p)}\left(g^\star - g\frac{1+e^{-p(L-2)}}{1+e^{-p}} + \frac{ge^{-p}(1+\xi_3(t))}{(1+e^{-p})^2(1+\xi_5(t))}\right),$$
$$\frac{\mathrm{d}g}{\mathrm{d}t} = \frac{1+\xi_3(p)}{1-e^{-p}}\left(g^\star - g\frac{1+e^{-p(L-2)}}{1+e^{-p}}\right),$$

also satisfy

$$|\xi_1(p)|, \cdots, |\xi_5(p)| \le 0.0001, \ \forall t > T_1^g.$$

For simplicity, we consider the transform:

$$u := e^{-p}.$$

Then the dynamics of u and g can be written as:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\frac{(1+\xi_1(p))u^2 g}{1+\xi_2(p)}\left(g^\star - g\frac{1+u^{L-2}}{1+u} + \frac{gu(1+\xi_4(p))}{(1+u)^2(1+\xi_5(p))}\right),$$
$$\frac{\mathrm{d}g}{\mathrm{d}t} = \frac{1+\xi_3(p)}{1-u}\left(g^\star - g\frac{1+u^{L-2}}{1+u}\right).$$

Notice that this dynamics are controlled by high-order terms. Consequently, we construct a variable to reflect the dynamics of high-order term:

$$v := ug^\star + (g^\star - g).$$

Then the dynamics of u and v satisfy:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\frac{(1+\xi_1(p))u^2 g}{1+\xi_2(p)}\left(\frac{v-u^{L-2}g}{1+u} + \frac{gu(1+\xi_4(p))}{(1+u)^2(1+\xi_5(p))}\right),$$
$$\frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{(1+\xi_1(p))u^2 gg^\star}{1+\xi_2(p)}\left(\frac{v-u^{L-2}g}{1+u} + \frac{gu(1+\xi_4(p))}{(1+u)^2(1+\xi_5(p))}\right) - \frac{1+\xi_3(p)}{1-u^2}\left(v-u^{L-2}g\right).$$

Now we consider the Lyapunov function about u, v:

$$G(u,v) := \frac{1}{2}\left(u^2+v^2\right).$$

Then it is straightforward:

$$\begin{aligned}\frac{\mathrm{d}G}{2\mathrm{d}t} &= u\frac{\mathrm{d}u}{\mathrm{d}t} + v\frac{\mathrm{d}v}{\mathrm{d}t} \\ &= -\frac{u^3 g(1+\xi_1(p))}{1+\xi_2(p)}\left(\frac{v-u^{L-2}g}{1+u} + \frac{gu(1+\xi_4(p))}{(1+u)^2(1+\xi_5(p))}\right) \\ &\quad -\frac{(1+\xi_1(p))u^2 vgg^\star}{1+\xi_2(p)}\left(\frac{v-u^{L-2}g}{1+u} + \frac{gu(1+\xi_4(p))}{(1+u)^2(1+\xi_5(p))}\right)\end{aligned}$$

$$- \frac{1+\xi_3(p)}{1-u^2}\left(v - u^{L-2}g\right)v.$$

By $|\xi_1|, \cdots, |\xi_5| \leq 0.0001$, we have the following estimate for the Lyapunov dynamics:

$$\begin{aligned}\frac{\mathrm{d}G}{2\mathrm{d}t} \leq & \frac{1.001g}{1+u}|u^3v| + \frac{1.0001g^2}{1+u}u^{L+1} - \frac{0.999g^2}{(1+u^2)}u^4 \\ & - \frac{0.999gg^\star}{1+u}u^2v^2 + \frac{1.001g^2g^\star}{1+u}|u^Lv| + \frac{1.001g^2g^\star}{(1+u^2)}|u^3v| \\ & - \frac{0.999}{1-u^2}v^2 + \frac{1.001g}{1-u^2}|u^{L-2}v|\end{aligned}$$

By $u^{L-5} = e^{-p(L-5)} < 0.0001$ and $0 < u < e^{-p(T_1^g)}$, we further have:

$$\begin{aligned}&\frac{\mathrm{d}G}{2\mathrm{d}t} \leq \frac{1.002g}{1+u}|u^3v| - \frac{0.99g^2}{(1+u)^2}u^4 - \frac{0.999gg^\star}{1+u}u^2v^2 + \frac{1.005g^2g^\star}{(1+u)^2}|u^3v| - \frac{0.999}{1-u^2}v^2 \\ \leq & -\frac{0.99g^2}{(1+u)^2}u^4 - \frac{0.99gg^\star}{1+u}u^2v^2 - \frac{0.99}{1-u^2}v^2 + 1.01\left(\frac{g}{1+u} + \frac{g^2g^\star}{(1+u)^2}\right)|u^3v|.\end{aligned}$$

By using the following inequalities:

$$\begin{aligned}\frac{g^2g^\star}{(1+u)^2}|u^3v| &\leq \frac{1}{2}\left(\frac{1.98}{1.01}\frac{gg^\star}{1+u}u^2v^2 + \frac{1.01}{1.98}\frac{g^3g^\star}{(1+u)^3}u^4\right) \\ \frac{g}{1+u}|u^3v| &\leq \frac{1}{2}\left(\frac{0.99}{1.01}(1+u)v^2 + \frac{1.01}{0.99}\frac{g^2}{(1+u)^3}u^6\right) \\ -\frac{1}{1-u^2} + \frac{1}{2}(1+u) &< -\frac{2}{5}\end{aligned}$$

we have

$$\frac{\mathrm{d}G}{\mathrm{d}t} \leq -0.99\frac{g^2}{(1+u)^2}u^4 + \frac{1.01}{3.96}\frac{g^3g^\star}{(1+u)^3}u^4 + \frac{1.01}{1.98}\frac{g^2}{(1+u)^3}u^6 - \frac{1.98}{5}v^2.$$

Since $g^\star < g < 2g^\star$, $u > 0$ for $t > T_1^g$, and $\frac{u^2}{1+u} \leq \frac{1}{2}$ for $0 \leq u \leq 1$, we have:

$$\begin{aligned}&\frac{1}{4}\frac{g^3g^\star}{(1+u)^3} + \frac{1}{2}\frac{g^2u^2}{(1+u)^3} \leq \frac{g^2}{(1+u)^2}\left(\frac{g^{\star 2}}{2(1+u)} + \frac{u^2}{2(1+u)}\right) \\ \leq & \frac{g^2}{(1+u)^2}\left(\frac{1}{2} + \frac{1}{4}\right) = \frac{3}{4}\frac{g^2}{(1+u)^2},\end{aligned}$$

then

$$\begin{aligned}\frac{\mathrm{d}G(u,v)}{\mathrm{d}t} &\leq -0.22\frac{g^2}{(1+u)^2}u^4 - \frac{2}{5}v^2 \\ &\leq -\frac{0.99}{16}g^{\star 2}u^4 - \frac{1.98}{5}v^2 \leq -\frac{g^{\star 2}}{65}G(u,v)^2,\end{aligned}$$

which implies:

$$G(u(t),v(t)) \leq \frac{1}{G(u(t_1),v(t_1)) + \frac{g^{\star 2}}{64}(t-t_1)}, \quad \forall t > T_1^g.$$

Hence,

$$u^2(t), \quad v^2(t) = \mathcal{O}\left(\frac{1}{g^{\star 2}t}\right) = \mathcal{O}\left(\frac{1}{t}\right), \quad \forall t > T_1^g = \mathcal{O}(1)$$

which implies:

$$\begin{aligned} e^{-p(t)} = u(t) = \mathcal{O}\left(\frac{1}{\sqrt{t}}\right), \quad \forall t > T_1^g = \mathcal{O}(1); \\ g(t) - g^\star = g^\star u(t) - v(t) \le \mathcal{O}\left(\frac{g^\star}{\sqrt{t}}\right) + \mathcal{O}\left(\frac{1}{\sqrt{t}}\right) = \mathcal{O}\left(\frac{1}{\sqrt{t}}\right), \quad \forall t > T_1^g = \mathcal{O}(1). \end{aligned} \tag{21}$$

Notably, these proofs capture the **entire** training dynamics of p, g, from $t = 0$ to $t = T_1^g$, and finally to $t \to +\infty$, providing a fine-gained analysis for each phase.

C.2 Dynamics of the parameters for induction head

Recall the partial loss about the induction head:

$$\mathcal{L}_{\mathsf{IH}_2}(\theta) = \frac{1}{2}\mathbb{E}_X\left[\left(\frac{1}{1+\alpha^\star}(x_s)_{s=2}^{L-1} \cdot \mathrm{softmax}\left(\left(w^{\star 2} x_L x_{s-1}\right)_{s=2}^{L-1}\right) - h \cdot (x_s)_{s=2}^{L-2} \cdot \mathrm{softmax}\left(\left(w^2 x_L x_{s-1}\right)_{s=2}^{L-2}\right)\right)^2\right].$$

Technical simplification. Unlike $\mathcal{L}_{\mathsf{G}_4}(\theta)$, the denominators of the softmax terms $\mathrm{softmax}\left(\left(w^{\star 2} x_L x_{s-1}\right)_{s=2}^{L-1}\right)$ and $\mathrm{softmax}\left(\left(w^2 x_L x_{s-1}\right)_{s=2}^{L-2}\right)$ in $\mathcal{L}_{\mathsf{IH}_2}(\theta)$ depend on the input tokens X, making it hard to derive a closed-form expression for $\mathcal{L}_{\mathsf{IH}_2}(\theta)$. In Bai et al. (2023), the authors consider a simplified transformer model, which replaces the softmax $\mathrm{softmax}(z_1, \cdots, z_L)$ with $\frac{1}{L}\exp(z_1, \cdots, z_L)$. This approximation is nearly tight when $z_1, \cdots, z_L \approx 0$. Notice that 1) $w^2 x_L x_{s-1} \approx 0$ holds near the small initialization, i.e., for $w \approx \sigma_{\text{init}} \ll 1$. In fact, our analysis shows that $w \approx \sigma_{\text{init}}$ is maintained over a long period. 2) $w^\star = \mathcal{O}(1)$, which implies that $w^2 x_L x_{s-1} \approx 0$ for most input sequence. Thus, we adopt the simplification used in Bai et al. (2023), resulting in the following approximation of the loss function:

$$\mathcal{L}_{\mathsf{IH}_2}(\theta) := \frac{1}{2}\mathbb{E}_X\left[\left(\frac{1}{1+\alpha^\star}\frac{1}{L-2}\sum_{s=2}^{L-1}\exp(w^{\star 2} x_L x_{s-1}) x_s - h\frac{1}{L-2}\sum_{s=2}^{L-2}\exp(w^2 x_L x_{s-1}) x_s\right)^2\right].$$

Then by a straightforward calculation with Lemma D.3, we can derive its explicit formulation:

$$\mathcal{L}_{\mathsf{IH}_2}(\theta) = \frac{(1-4w^{\star 4})^{-\frac{1}{2}}}{2(1+\alpha^\star)^2(L-2)} + \frac{1}{2}\frac{h^2}{L-2}(1-4w^4)^{-\frac{1}{2}} - \frac{h(1-(w^2+w^{\star 2})^2)^{-\frac{1}{2}}}{(1+\alpha^\star)(L-2)}. \tag{22}$$

Furthermore, we can calculate GF dynamics as follows:

$$\begin{aligned} \frac{\mathrm{d}w}{\mathrm{d}t} &= \frac{h}{(1+\alpha^\star)(L-2)}(1-(w^2+w^{\star 2})^2)^{-\frac{3}{2}} \cdot (w^2+w^{\star 2}) \cdot 2w - \frac{h^2}{L-2}(1-4w^4)^{-\frac{3}{2}} \cdot 4w^3, \\ \frac{\mathrm{d}h}{\mathrm{d}t} &= \frac{1}{(1+\alpha^\star)(L-2)}(1-(w^2+w^{\star 2})^2)^{-\frac{1}{2}} - \frac{h}{L-2}(1-4w^4)^{-\frac{1}{2}}. \end{aligned}$$

For simplicity, we denote:

$$w^\star := w^\star, \quad h^\star := \frac{1}{1+\alpha^\star}.$$

Part I. The trend and monotonicity of w, h.

For simplicity, we denote the tuning time point of h:

$$T_2^h := \inf\left\{t > 0 : \frac{\mathrm{d}h(t)}{\mathrm{d}t} = 0\right\}.$$

In this step, we will prove the following three claims regarding the trend and monotonicity of w, h, which are essential for our subsequent analysis:

- **(P1.1)** h initially increases beyond $h^\star$, and then remains above this value.

- **(P1.2)** w keeps increasing but always stays below $w^\star$.
- **(P1.3)** h increases before T_2^h, but decreases after T_2^h.

(P1.1) *h initially increases beyond $h^\star$, and then remains above this value.*

We will prove that initially, h increases beyond $h^\star$, and keeps growing beyond $h^\star$. Define

$$T_1^h := \inf\{t > 0 : h(t) > h^\star\},$$

we will prove that h remains above $h^\star$ thereafter.

For simplicity, we denote

$$\psi(x) = (1-x^2)^{-\frac{1}{2}}, \quad \phi(x) = (1-x^2)^{-\frac{3}{2}} \cdot x,$$

then the dynamics holds:

$$\begin{aligned}
\frac{\mathrm{d}h}{\mathrm{d}t} &= \frac{h}{L-2}\psi(w^2+w^{\star 2})\left[\frac{h^\star}{h} - \frac{\psi(2w^2)}{\psi(w^2+w^{\star 2})}\right], \\
\frac{\mathrm{d}w}{\mathrm{d}t} &= \frac{2h^2 w}{L-2} \cdot \phi(w^2+w^{\star 2}) \cdot \left[\frac{h^\star}{h} - \frac{\phi(2w^2)}{\phi(w^2+w^{\star 2})}\right].
\end{aligned}$$

Notice that $\frac{\phi(2w^2)}{\phi(w^2+w^{\star 2})} < \frac{\psi(2w^2)}{\psi(w^2+w^{\star 2})},\ w < w^\star$, while $\frac{\phi(2w^2)}{\phi(w^2+w^{\star 2})} > \frac{\psi(2w^2)}{\psi(w^2+w^{\star 2})},\ w > w^\star$.

We denote the first hitting time of h decreasing to $h^\star$ as $\dot{t}$:

$$\dot{t} := \inf\left\{t > T_2^h : h(t) < h^\star\right\}.$$

If $w(\dot{t}) \geq w^\star$, then at the first hitting time of w increasing to $w^\star$, $\frac{\mathrm{d}w}{\mathrm{d}t} < 0$, which leads to a contradiction. If $w(\dot{t}) < w^\star$, then $\frac{\mathrm{d}h}{\mathrm{d}t}|_{\dot{t}} > 0$, which also leads to a contradiction. Hence, $\dot{t} = +\infty$, which means that h always remains above $h^\star$ for $t > T_2^h$.

(P1.2) *w keeps increasing but always below $w^\star$.*

We first prove that w always remains below $w^\star$. We denote the first hitting time of w increasing to $w^\star$ as t', then it is not difficult to see $\frac{\mathrm{d}w}{\mathrm{d}t}|_{t'} < 0$, which leads to a contradiction.

Next we prove that w keeps increasing throughout. We define the following functions

$$\begin{aligned}
H &:= \frac{1}{1+\alpha^\star}\left(1-(w^2+w^{\star 2})^2\right)^{-\frac{3}{2}}(w^2+w^{\star 2}) - h(1-4w^4)^{-\frac{3}{2}} \cdot 2w^2 \\
Q &:= \frac{1}{1+\alpha^\star}\left(1-(w^2+w^{\star 2})^2\right)^{-\frac{1}{2}} - h\left(1-4w^4\right)^{-\frac{1}{2}}
\end{aligned}$$

If at some $\bar{t}$, $\frac{\mathrm{d}w}{\mathrm{d}t}$ reaches its zero point at the first time, then

$$\left.\frac{\mathrm{d}H}{\mathrm{d}t}\right|_{\bar{t}} = -h'(\bar{t})(1-4w^{\star 4})^{-\frac{3}{2}} \cdot 2w(\bar{t}) > 0,$$

which leads to a contradiction. Hence $\bar{t}$ does not exist and w keeps increasing.

(P1.3) *After the tuning point $t > T_2^h$, h will be monotonically decreasing.*

The first sign-changing zero point of $\frac{\mathrm{d}h}{\mathrm{d}t}$ is T_2^h, then $Q(T_2^h) = 0$. $H(T_2^h) > 0$,

$$\begin{aligned}
\left.\frac{\mathrm{d}Q}{\mathrm{d}t}\right|_{T_2^h} &= \frac{1}{1+\alpha^\star}(1-(w(T_2^h)^2+w^{\star 2})^2)^{-\frac{1}{2}} \cdot 2w(T_2^h) \cdot w'(T_2^h) \\
&\quad \cdot \left[(1-(w(T_2^h)^2+w^{\star 2})^2)^{-1} \cdot (w(T_2^h)^2+w^{\star 2}) - (1-4w(T_2^h)^4)^{-1} \cdot 4w(T_2^h)^2\right].
\end{aligned}$$

We can see that T_2^h is a sign-changing zero point only if

$$\frac{(1-4w(T_2^h)^4) \cdot (w(T_2^h)^2+w^{\star 2})}{(1-(w(T_2^h)^2+w^{\star 2})^2) \cdot 4w(T_2^h)^2} < 1,$$

i.e. we have:

$$w(T_2^h) > w^\circ := \sqrt{\frac{3 - 4w^{\star 4} - \sqrt{(4w^{\star 4} - 3)^2 - 16w^{\star 4}}}{8w^{\star 2}}} \geq \frac{w^\star}{2}, \tag{23}$$

when $w^\star = \mathcal{O}(1)$.

Next we show that h keeps decreasing after T_2^h. We denote the first zero point of $\frac{\mathrm{d}h}{\mathrm{d}t}$ as t°, then $Q(t^\circ) = 0$. Since $\frac{\mathrm{d}w}{\mathrm{d}t}|_{t^\circ} > 0$, we have $\frac{\mathrm{d}Q}{\mathrm{d}t}|_{t^\circ} > 0$ which leads to a contradiction. Hence t° does not exist and h keeps decreasing after T_2^h.

Part II. Estimation of T_1^h, T_2^h, and the tight estimate of $w(t)$ before T_2^h.

At the first stage, we prove that h grows first and w barely increases. If $w \leq 0.01w^\star$ and $h \leq \frac{1}{1+\alpha^\star}\frac{(1-w^{\star 4})^{-\frac{1}{2}}}{(1-0.01^4 w^{\star 4})^{-\frac{1}{2}}}$,

$$\frac{\mathrm{d}h}{\mathrm{d}t} \geq \frac{-1}{L-2}\left[h(1-0.01^4 w^{\star 4})^{-\frac{1}{2}} - \frac{1}{1+\alpha^\star}(1-w^{\star 4})^{-\frac{1}{2}}\right],$$

$$h \geq \frac{1}{1+\alpha^\star}\frac{(1-w^{\star 4})^{-\frac{1}{2}}}{(1-0.01^4 w^{\star 4})^{-\frac{1}{2}}} - \left[\frac{1}{1+\alpha^\star}\frac{(1-w^{\star 4})^{-\frac{1}{2}}}{(1-0.01^4 w^{\star 4})^{-\frac{1}{2}}} - h(0)\right]\exp\left(\frac{-t}{(L-2)(1-0.01w^{\star 4})^{\frac{1}{2}}}\right). \tag{24}$$

For h increasing from $h(0)$ to $\frac{1}{1+\alpha^\star}$, it takes

$$\begin{aligned} T_1^h &\leq (1-0.01w^{\star 4})^{\frac{1}{2}}(L-2)\ln\left(\frac{1}{1-\frac{(1-w^{\star 4})^{\frac{1}{2}}}{(1-0.01^4 w^{\star 4})^{\frac{1}{2}}}}\right) \\ &\leq 2(L-2)(1-\frac{1}{2}w^{\star 4}) = \mathcal{O}(L). \end{aligned} \tag{25}$$

For $0 \leq t \leq T_1^h$,

$$\frac{\mathrm{d}w}{\mathrm{d}t} \leq \frac{1}{L-2}(1-4w^{\star 4})^{-\frac{3}{2}} \cdot w^{\star 2} \cdot 4w.$$

Hence, it take $\mathcal{O}(L\log(1/\sigma_{\text{init}}))$ for w to reach $0.01w^\star$, which allows sufficient time for h to reach $\frac{1}{1+\alpha^\star}$ beforehand.

Therefore, there exists a small constant $\varepsilon(w(0), w^\star)$ only depends on $w(0)$ and $w^\star$ such that h is dominated by $1 + \varepsilon(w(0), w^\star)$ times right hand side of (24), from which we deduce that (25) is a tight estimation of T_1^h instead of an upper bound, i.e. $T_1^h = \Theta(L)$.

We then give a bound for $h(T_2^h)$. By $\frac{\mathrm{d}h}{\mathrm{d}t} = 0$,

$$h(T_2^h)/h^\star \leq \frac{(1-4w^4)^{\frac{1}{2}}}{(1-(w^2+w^{\star 2})^2)^{\frac{1}{2}}} := r(w).$$

Moreover, $r(w)$ is an decreasing function of w for $w > w^\circ$, and w° is a function of $w^\star$, we have

$$h(T_2^h)/h^\star \leq r(w^\circ) := R(w^\star),$$

where w° is a function about $w^\star$, defined in Eq. (23). It is clear that

$$R(w^\star = 0) = 1, \quad R'(w^\star = 0) = 0.$$

Then using the continuity of $R'(\cdot)$ (in $[0, 0.4]$), there exists $c > 0$ such that $|R'(w^\star)| < 0.04$ holds for all $0 < w^\star < c$, which implies:

$$R(w^\star) = R(0) + \int_0^{w^\star} R'(v)\mathrm{d}v < 1 + 0.04w^\star, \quad 0 < w^\star < c.$$

i.e., if $w^\star = O(1)$, then $R(w^\star) < 1 + 0.04w^\star$. This implies:

$$h^\star \le h(t) \le (1 + 0.04375w^\star)h^\star, \quad \forall t \ge T_1^h. \tag{26}$$

By some computation, we can prove that $w^\circ(w^\star)$ is an increasing function of $w^\star$, and is always above $\frac{1}{2}w^\star$. Thus we obtain a lower bound of w° for the estimation of lower bound of T_2^h:

For the second stage, h barely changes and w starts to grow exponentially fast, and we use the tight estimation of $T_{1/2}^w := \inf\left\{t > 0 : w(t) > \frac{1}{2}w^\star\right\}$ to give a lower bound of T_2^h. During this stage,

$$\begin{aligned}\frac{\mathrm{d}w}{\mathrm{d}t} &\le \frac{2w}{(1+\alpha^\star)^2(L-2)}\left[(1-(w^2+w^{\star 2})^2)^{-\frac{3}{2}}\cdot(w^2+w^{\star 2}) - (1-4w^4)^{\frac{3}{2}}\right] \\ &\le \frac{2w}{(1+\alpha^\star)^2(L-2)}(1-4w^{\star 4})^{\frac{3}{2}}\cdot 2w^{\star 2},\end{aligned}$$

and w has upper bound

$$w \le w(0)\exp\left(\frac{4w^{\star 2}(1-4w^{\star 4})^{\frac{3}{2}}}{(1+\alpha^\star)(L-2)}t\right). \tag{27}$$

Hence, the lower bound of time for w to reach $\frac{1}{2}w^\star$ is

$$T_{1/2}^w - T_1^h = \frac{(1+\alpha^\star)^2(L-2)}{4w^{\star 2}(1-4w^{\star 4})^{\frac{3}{2}}}\ln(\frac{w^\star}{2w(0)}),$$

and lower bound for $T_{1/2}^w$ is

$$\begin{aligned}T_{1/2}^w &\ge (L-2)\left[\frac{(1+\alpha^\star)^2\ln(\frac{w^\star}{2w(0)})}{4w^{\star 2}(1-4w^{\star 4})^{\frac{3}{2}}} - \ln\left(1-(1-w^{\star 4})^{\frac{1}{2}}\right)\right] \\ &\ge \frac{(L-2)(1+\alpha^\star)^2}{16w^{\star 2}}\ln\left(\frac{1}{w(0)}\right) = \Omega\left(\frac{(1+\alpha^\star)^2 L}{w^{\star 2}}\log\left(\frac{1}{\sigma_{\text{init}}}\right)\right).\end{aligned} \tag{28}$$

On the other hand, we estimate the lower bound of w. Let

$$C(x) = (1-x^2)^{-\frac{3}{2}}\cdot x,$$

then

$$C'(x) = 3(1-x^2)^{-\frac{5}{2}}x^2 + (1-x^2)^{-\frac{3}{2}} > 1, \quad 0 < x < 1,$$

$$C''(x) = 15x^3(1-x^2)^{-\frac{7}{2}} + 6x(1-x^2)^{-\frac{5}{2}} + 3x(1-x^2)^{-\frac{5}{2}} > 0, \quad 0 < x < 1.$$

$C(x)$ is a monotonically increasing convex function on $(0,1)$ and $C(x) \ge x$.

Using conclusions above, before w^2 increases to $\frac{1}{2\gamma(w^\star)+\beta-1}w^{\star 2}$ for some $\beta > 0$,

$$\begin{aligned}&C(w^2 + w^{\star 2}) \\ &\ge C((2\gamma(w^\star)+\beta)w^2) \\ &\ge C(2\gamma(w^\star)\cdot w^2) + C(\beta w^2) \quad \text{(Lemma D.6)} \\ &\ge \gamma(w^\star)\cdot C(2w^2) + \beta w^2 \quad (C(ax) \ge aC(x), \text{for } a > 1)\end{aligned}$$

then we have

$$\begin{aligned}\frac{\mathrm{d}w}{\mathrm{d}t} &\ge \frac{2w}{(1+\alpha^\star)^2(L-2)}(C(w^2+w^{\star 2}) - \gamma(w^\star)\cdot C(2w^2)) \\ &\ge \frac{2w}{(1+\alpha^\star)^2(L-2)}\frac{\beta}{\gamma(w^\star)+\beta}w^{\star 2}\end{aligned}$$

and

$$w \ge w(0)\exp\left(\frac{2\beta}{\gamma(w^\star)+\beta}\frac{1}{(1+\alpha^\star)^2(L-2)}w^{\star 2}t\right).$$

Take $\beta = 2$, then

$$w \geq w(0) \exp\left(\frac{w^{\star 2} t}{(1+\alpha^\star)^2 (L-2)}\right), \ \forall t \in [0, T^w_{1/2}]. \tag{29}$$

From the above inequality, (28) is not only an upper bound, but a tight estimation of $T^w_{1/2}$, i.e.

$$T^w_{1/2} = \Theta\left(\frac{(1+\alpha^\star)^2 L}{w^{\star 2}} \log\left(\frac{1}{\sigma_{\text{init}}}\right)\right).$$

Part II. Dynamics after the critical point $T^w_{1/2}$.

For simplicity, we consider:

$$v := w^2,$$

and denote $v^\star := w^{\star 2}, h^\star := \frac{1}{1+\alpha^\star}$. Then we focus on the dynamics of v and h.

Additionally, we introduce a few notations used in this part:

$$\phi(x) := \frac{x}{(1-x^2)^{3/2}}, \quad \psi(x) := \frac{1}{(1-x^2)^{1/2}}.$$

Then the dynamics of v and g are:

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \frac{4vh}{L-2}\big(h^\star \phi(v+v^\star) - h\phi(2v)\big),$$
$$\frac{\mathrm{d}h}{\mathrm{d}t} = \frac{1}{L-2}\big(h^\star \psi(v+v^\star) - h\psi(2v)\big).$$

Step II.1. *A coarse estimate of the relationship between v and h.*

It is easy to verify the monotonicity that $\frac{\mathrm{d}v}{\mathrm{d}t} > 0$ and $\frac{\mathrm{d}h}{\mathrm{d}t} < 0$ for $t > t_2$. Additionally, we have

$$\frac{\psi(v+v^\star)}{\psi(2v)} < \frac{h}{h^\star} < \frac{\phi(v+v^\star)}{\phi(2v)}.$$

Then by Monotone convergence theorem, we obtain:

$$\lim_{t\to+\infty} v = v^\star, \quad \lim_{t\to+\infty} h = h^\star.$$

Step II.2. *Convergence analysis by Lyapunov function.*

This step aims to establish the convergence rate of v and h.

In fact, the dynamics of v, h can be approximately characterized by their linearized dynamics. In contrast, the dynamics of p, g are controlled by high-order terms. Therefore, the proof for v and h is significantly simpler than the corresponding proof for p and g. We only need to consider the simplest Lyapunov function:

$$G(v,h) := \frac{1}{2}\Big((v-v^\star)^2 + (h-h^\star)^2\Big).$$

It is easy to verify that

$$\begin{aligned}
&(L-2)\frac{\mathrm{d}G(v,h)}{\mathrm{d}t} = (v-v^\star)\frac{\mathrm{d}v}{\mathrm{d}t} + (h-h^\star)\frac{\mathrm{d}h}{\mathrm{d}t} \\
=&4vh(v-v^\star)\big(h^\star\phi(v+v^\star) - h\phi(2v)\big) + (h-h^\star)\big(h^\star\psi(v+v^\star) - h\psi(2v)\big) \\
=&4vh(v-v^\star)\Big(\phi(v+v^\star)(h^\star-h) - h(\phi(v+v^\star)-\phi(2v))\Big) \\
&+ (h-h^\star)\Big((h^\star-h)\psi(v+v^\star) + h(\psi(v+v^\star)-\psi(2v))\Big) \\
=&-4vh^2(v^\star-v)(\phi(v+v^\star)-\phi(2v)) - \psi(v+v^\star)(h-h^\star)^2
\end{aligned}$$

$$+ 4vh\phi(v+v^\star)(v-v^\star)(h^\star-h) + h(h-h^\star)(\psi(v+v^\star)-\psi(2v)).$$

Let $v^\star \le 0.3 = \mathcal{O}(1)$. Recalling (23) and (26), as well as the monotonicity about p and w, we have:

$$\frac{v^\star}{4} < v(t) < v^\star; \quad h^\star < h(t) < 1.02h^\star, \quad \forall t > T_2^h.$$

Combining these estimates with the properties of ϕ and ψ, we have the following straight-forward estimates:

$$\begin{aligned}
\phi(v+v^\star)-\phi(2v) = \phi'(\xi)(v^\star - v) = \frac{1+2\xi^2}{(1-\xi^2)^{5/2}}(v^\star - v) \ge v^\star - v;\\
\phi(v+v^\star) \le \phi(2v^\star) \le 1;\\
\psi(v+v^\star) = \frac{1}{(1-(v+v^\star)^2)^{1/2}} \ge 1;\\
\psi(v+v^\star)-\psi(2v) = \psi'(\xi)(v^\star - v) = \frac{\xi}{(1-\xi^2)^{3/2}}(v^\star - v) \le 1.3v^\star(v^\star - v).
\end{aligned}$$

Thus, we have the following estimate for the Lyapunov function:

$$\begin{aligned}
&(L-2)\frac{\mathrm{d}G(v,h)}{\mathrm{d}t}\\
\le& -\frac{4}{1.02}v^\star h^{\star 2}(v-v^\star)^2 - (h-h^\star)^2\\
&+ 4.08v^\star h^\star(v-v^\star)(h^\star-h) + 1.3\cdot 1.02 v^\star h^\star(v^\star - v)(h-h^\star)\\
=& -\frac{4}{1.02}v^\star h^{\star 2}(v-v^\star)^2 - (h-h^\star)^2 + 5.41v^\star h^\star(v^\star - v)(h-h^\star)\\
\le& -3.92v^\star h^{\star 2}(v-v^\star)^2 - (h-h^\star)^2 + \left(9.6v^{\star 2}h^{\star 2}(v-v^\star)^2 + \frac{3}{4}(h-h^\star)^2\right)\\
\le& -(3.92 - 9.6\cdot 0.3)v^\star h^{\star 2}(v-v^\star)^2 - 0.25(h-h^\star)^2 \le -\frac{1}{4}v^\star h^{\star 2}G(v,h).
\end{aligned}$$

Consequently, we have the exponential bound for all $t > T_2^h$:

$$G(v(t),h(t)) \le G\left(v(T_2^h),h(T_2^h)\right)\exp\left(-\frac{v^\star h^{\star 2}}{4(L-2)}(t-T_2^h)\right), \quad \forall t > T_2^h,$$

This can imply:

$$\begin{aligned}
(h(t)-h^\star)^2 &= (h(T_2^h)-h^\star)^2\exp\left(-\Omega\left(\frac{w^{\star 2}(t-T_2^h)}{L(1+\alpha^\star)^2}\right)\right)\\
&= \mathcal{O}\left(h^{\star 2}\exp\left(-\Omega\left(\frac{w^{\star 2}(t-T_2^h)}{L(1+\alpha^\star)^2}\right)\right)\right), \quad \forall t > T_2^h;\\
(w(t)-w^\star)^2 &= (w(T_2^h)-w^\star)^2\exp\left(-\Omega\left(\frac{w^{\star 2}(t-T_2^h)}{L(1+\alpha^\star)^2}\right)\right)\\
&= \mathcal{O}\left(w^{\star 2}\exp\left(-\Omega\left(\frac{w^{\star 2}(t-T_2^h)}{L(1+\alpha^\star)^2}\right)\right)\right), \quad \forall t > T_2^h.
\end{aligned} \tag{30}$$

Notably, these proofs capture the **entire** training dynamics of w, h, from $t = 0$ to $t = T_1^h$, to $t = T_{1/2}^w \le T_2^h$, and finally to $t \to +\infty$, providing a fine-gained analysis for each phase.

C.3 Proof of Theorem 5.5

This theorem is a direct corollary of our analysis of the entire training dynamics in Appendix C.1 and C.2, leveraging the relationship between the parameters and the loss.

Proof of Phase I (partial learning).

By combining (19) and (21), it follows that: $\mathcal{L}_{\mathsf{G}_4}(\theta(0)) = \Theta(1)$. Moreover,

$$\mathcal{L}_{\mathsf{G}_4}(\theta(t)) = \mathcal{O}\left(\frac{1}{t}\right), \quad t > T_1^g = O(1).$$

Thus, there exists a sufficiently large $T_{\mathrm{I}} = \Theta(1)$, such that:

$$\mathcal{L}_{\mathsf{G}_4}(\theta(T_{\mathrm{I}})) \leq 0.01\mathcal{L}_{\mathsf{G}_4}(\theta(0)).$$

Recalling our proof in Appendix C.2, for $t < T_{1/2}^h = \mathcal{O}(L)$, it holds that $h(t) < \sigma_{\text{init}} + \mathcal{O}(t/((1+\alpha^\star)L)), w(t) < \sigma_{\text{init}} + o(t/((1+\alpha^\star)L))$. Additionally, since $T_{\mathrm{I}} = \Theta(1) \ll \Theta(L)$, it follows that

$$w(T_{\mathrm{I}}) = \mathcal{O}(\sigma_{\text{init}} + 1/L) < 2\sigma_{\text{init}} \ll w^\star, \quad h(T_{\mathrm{I}}) = \mathcal{O}(\sigma_{\text{init}} + 1/L) < 2\sigma_{\text{init}} \ll h^\star.$$

Substituting these estimates into (22), we obtain by Lipschitz continuity of $\mathcal{L}_{\mathsf{IH}_2}$:

$$\begin{aligned}
|\mathcal{L}_{\mathsf{IH}_2}(\theta(T_{\mathrm{I}})) - \mathcal{L}_{\mathsf{IH}_2}(\theta(0))| &\leq 2\sigma_{\text{init}} \left(\left|\frac{\partial \mathcal{L}_{\mathsf{IH}_2}}{\partial w}\right| + \left|\frac{\partial \mathcal{L}_{\mathsf{IH}_2}}{\partial h}\right|\right) \\
&\leq 2\sigma_{\text{init}} \left(\mathcal{O}\left(\frac{1}{(1+\alpha^\star)L}\right) + o\left(\frac{1}{(1+\alpha^\star)L}\right)\right) \\
&\leq 0.01\mathcal{L}_{\mathsf{IH}_2}(\theta(0)).
\end{aligned}$$

Thus,

$$\mathcal{L}_{\mathsf{IH}_2}(\theta(T_{\mathrm{I}})) \geq 0.99\mathcal{L}_{\mathsf{IH}_2}(\theta(0)).$$

Proof of Phase II (plateau) + Phase III (emergence).

First, (27) and (29) ensures that w grows exponentially before $t < T_{1/2}^w$:

$$\sigma_{\text{init}} \exp\left(\frac{w^{\star 2}}{(1+\alpha^\star)^2(L-2)}t\right) \leq w \leq \sigma_{\text{init}} \exp\left(\frac{4w^{\star 2}(1-4w^{\star 4})^{\frac{3}{2}}}{(1+\alpha^\star)(L-2)}t\right).$$

Thus, we have:

$$w(t) = \sigma_{\text{init}} \exp\left(\Theta\left(\frac{w^{\star 2}t}{(1+\alpha^\star)^2 L}\right)\right), \quad t < \Theta\left(\frac{(1+\alpha^\star)^2 L}{w^{\star 2}} \log\left(\frac{1}{\sigma_{\text{init}}}\right)\right).$$

Now we define the observation time $T_o := T_1^h = \Theta(L)$. Notably,

$$h(T_o) = h^\star, \quad w(T_o) < 0.01w^\star.$$

The exponential growth of w further implies:

$$T_{0.01}^w := \{t > 0 : w(t) > 0.01w^\star\} = \Theta\left(\frac{(1+\alpha^\star)^2 L}{w^{\star 2}} \log\left(\frac{1}{\sigma_{\text{init}}}\right)\right).$$

Regarding the dynamics of h, by (26), we have $|h(t) - h(T_o)| < 0.02|h(T_o)|, \forall t \geq T_o$.

Now we incorporate these facts ($0 < w(T_o) < 0.01w^\star$, $0 < w(T_{0.01}^w) \leq 0.01w^\star$, $|h(T_{0.01}^w) - h(T_o)| < 0.02|h(T_o)|$, $h(T_o) = h^\star$) into the loss (22). By the Lipschitz continuity of $\mathcal{L}_{\mathsf{IH}_2}$, it is straightforward that

$$\mathcal{L}_{\mathsf{IH}_2}(\theta(T_{0.01}^w)) \geq 0.99\mathcal{L}(\theta(T_o)).$$

Thus, we have established the lower bound for T_{II}:

$$T_{\mathrm{II}} := \inf\left\{t > T_o : \mathcal{L}_{\mathsf{IH}_2}(\theta(t)) \leq 0.99 \cdot \mathcal{L}_{\mathsf{IH}_2}(\theta(T_o))\right\}$$

$$\geq T_{0.01}^{w} = \Omega\left(\frac{(1+\alpha^\star)^2 L}{w^{\star 2}} \log\left(\frac{1}{\sigma_{\text{init}}}\right)\right).$$

Combining the loss (22) and our parameter estimates (30), we obtain:

$$\mathcal{L}_{\mathsf{IH}_2}(\theta(t)) = \mathcal{O}\left(\exp\left(-\Omega\left(\frac{w^{\star 2} t}{L(1+\alpha^\star)^2}\right)\right)\right),\ t > T_2^h = \Theta\left(\frac{(1+\alpha^\star)^2 L}{w^{\star 2}} \log\left(\frac{1}{\sigma_{\text{init}}}\right)\right).$$

This implies the upper bound for T_{III}:

$$\begin{aligned} T_{\text{III}} &:= \inf\left\{t > T_o : \mathcal{L}_{\mathsf{IH}_2}(\theta(t)) \leq 0.01 \cdot \mathcal{L}_{\mathsf{IH}_2}(\theta(T_o))\right\} \\ &= T_{1/2}^{w} + \mathcal{O}\left((\alpha^\star+1)^2 L \log(1/\sigma_{\text{init}})/w^{\star 2}\right) = \mathcal{O}\left((\alpha^\star+1)^2 L \log(1/\sigma_{\text{init}})/w^{\star 2}\right). \end{aligned}$$

Combining the fact $T_{\text{II}} < T_{\text{III}}$, the lower bound for T_{II}, and the uppper bound for T_{III}, we obtain the two-sided bounds for both T_{II} and T_{III}:

$$T_{\text{II}},\ T_{\text{III}} = \Theta\left((\alpha^\star+1)^2 L \log(1/\sigma_{\text{init}})/w^{\star 2}\right).$$

Proof of Phase IV (convergence).

By combining the loss (19), (22), and our parameter estimates (21), (30), it follows that:

$$\mathcal{L}_{\mathsf{G}_4}(\theta(t)) = \mathcal{O}\left(\frac{1}{t}\right), \quad \mathcal{L}_{\mathsf{IH}_2}(\theta(t)) = \mathcal{O}\left(\exp\left(-\Omega\left(\frac{w^{\star 2} t}{L(1+\alpha^\star)^2}\right)\right)\right), \quad t > T_{\text{III}}.$$

D Useful Inequalities

Lemma D.1 (Corollary A.7 in Edelman et al. (2022)). *For any $\theta, \theta' \in \mathbb{R}^d$, we have*

$$\|\text{softmax}(\theta) - \text{softmax}(\theta')\|_1 \leq 2\|\theta - \theta'\|_\infty$$

Lemma D.2 (lemma E.1 in Wang and E (2024)). *For any $T \in \mathbb{N}_+$, $q, m \in \mathbb{N}_+$, there exists and absolute constant $C > 0$ and a $\phi_m^{\exp}(t) = \sum_{k=1}^{m} \alpha_k e^{-\beta_k t}$ such that*

$$\|\mathbb{I}(\cdot = T) - \phi_m^{\exp}(\cdot)\|_{\ell_1(\mathbb{N})} \leq \frac{C e^{q+0.01(q+1)T}}{m^q}.$$

where $\beta_k > 0$ holds for any $k \in [m]$.

Lemma D.3. $\mathop{\mathbb{E}}_{X,Y,Z} \exp(aXY) Z^2 = (1-a^2)^{-1/2}$, $a < 1$.

Proof of Lemma D.3.

$$\begin{aligned} &\int \exp(aXY) Z^2 \left(\frac{1}{2\pi}\right)^{-3/2} \exp(-\frac{1}{2}X^2 - \frac{1}{2}Y^2 - \frac{1}{2}Z^2)\, dX dY dZ \\ &= \int \frac{1}{2\pi} \exp(-\frac{1}{2}(X - aY)^2 - \frac{1}{2}Y^2 + \frac{1}{2}a^2 Y^2)\, d(X - aY) dY \\ &= \int \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}W^2)\, dW \ (W = (1-a^2)^{1/2} Y) \\ &= (1-a^2)^{-1/2} \end{aligned}$$

□

Lemma D.4. *Let* $M(p) := \frac{1-e^{-p(L-2)}}{1-e^{-p}}$*, then it holds that*

$$\left\|\mathrm{softmax}\big((-p(L-1-s))_{s=1}^{L-1}\big)\right\|_2^2 = \frac{M(2p)}{M(p)^2}.$$

Definition D.5 (weakly majorizes)**.** A vector $\mathbf{x} \in \mathbb{R}^n$ is said to *weakly majorize* another vector $\mathbf{y} \in \mathbb{R}^n$, denoted by $\mathbf{x} \prec_w \mathbf{y}$, if the following conditions hold:

1. $\sum_{i=1}^{k} x_{[i]} \leq \sum_{i=1}^{k} y_{[i]}$ for all $k = 1, 2, \ldots, n-1$,
2. $\sum_{i=1}^{n} x_{[i]} = \sum_{i=1}^{n} y_{[i]}$,

where $x_{[i]}$ and $y_{[i]}$ are the components of $\mathbf{x}$ and $\mathbf{y}$, respectively, arranged in decreasing order.

Lemma D.6 (Weighted Karamata Inequality)**.** *Let* $f\colon \mathbb{R} \to \mathbb{R}$ *be a convex function, and let* $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ *and* $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ *be two vectors in* $\mathbb{R}^n$*. If* $\mathbf{x}$ *weakly majorizes* $\mathbf{y}$ *(i.e.,* $\mathbf{x} \prec_w \mathbf{y}$*), and* $w_1, w_2, \ldots, w_n$ *are non-negative weights such that*

$$\sum_{i=1}^{n} w_i = 1,$$

then the following inequality holds:

$$\sum_{i=1}^{n} w_i f(x_i) \leq \sum_{i=1}^{n} w_i f(y_i).$$

E Experiments

E.1 Experimental details for Figure 2

In line with our theoretical setting, we examine a simplified two-layer transformer, as described in (10). Specifically, the first layer only contains RPE (3) and the second layer consists of two heads: one uses only RPE and the other employs only dot-product structure. The target function is specified by (8) with $\alpha^\star = 1, w^\star = 0.49, \sigma_{\text{init}} = 0.01, L = 40$, and the distribution of each token is Gaussian, i.e., $x_i \overset{iid}{\sim} \mathcal{N}(0, 1)$ for $i \in [L]$. Training is conducted by minimizing the squared loss (11) using online SGD with learning rate 0.1 and batch size $B = 1,000$. Following our theoretical analysis, the two layers are trained sequentially:

- Training Stage I: only the first layer is trained for 100,000 iterations;
- Training Stage II: Subsequently, only the second layer undergoes training for another 100,000 iterations.

The dynamical behavior of the Training Stage II is visualized in Figure 2.

E.2 Additional experiments supporting optimization dynamics

1. **Standard transformers on real-world natural language dataset.**

Setup. We train a two-layer two-head **standard transformer** with RPE (3) (without any simplification) on the **wikitext-2** dataset, a natural language dataset (Merity et al., 2016). The transformer has an embedding dimension $D = 128$ and FFN width $W = 512$. For this dataset, the input dimension is $d = 33278$. We use a context length $L = 200$ and batch size $B = 32$. The parameters are initialized with the scale 0.01. The model is trained for 1,500 epochs on 1 H100, using cross-entropy loss and SGD with learning rate 0.1, and the initialization scale is 0.01. It is important to note that **both layers are trained simultaneously**. The results are presented in Figure 3.

2. **Discrete token distribution in toy setting.**

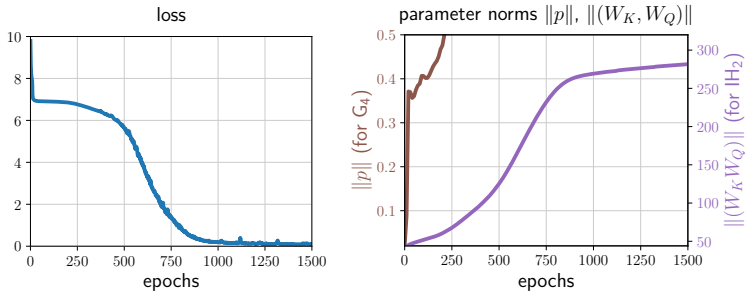

Figure 3: The loss and parameters for the experiment training a two-layer two-head **standard transformer** (without any simplification) on the **wikitext-2** dataset (Merity et al., 2016). Here, $\|p\|$ and $\|(W_K, W_Q)\|$ denote the Frobenius norms of all positional encoding parameters and all W_K, W_Q parameters across layers and heads, respectively, The results show that: the loss exhibits a clear plateau; position encoding p's are learned first; and the dot-product structure W_K, W_Q are learned slowly at the beginning, resembling an exponential increase; additionally, as W_K, W_Q are learned, the loss escapes that plateau. These findings closely resemble the behavior observed in our toy model (Figure 2). This experiment provides further support for our theoretical insights regarding the **time-scale separation** between the learning of positional encoding and the dot-product structure.

Setup. We modified the Gaussian input distribution used in the setup for Figure 2 to a boolean input distribution, where each input token, where each input token $x_i \overset{iid}{\sim} \mathrm{Unif}(\{\pm 1\})$ for $i \in [L]$, All other experimental setups remain the same as in the setup for Figure 2. The training dynamics of Stage (ii) are presented in Figure 4. We can see clearly that the dynamical behavior of the learning process is nearly the same as the one observed for Gaussian inputs in Figure 2.

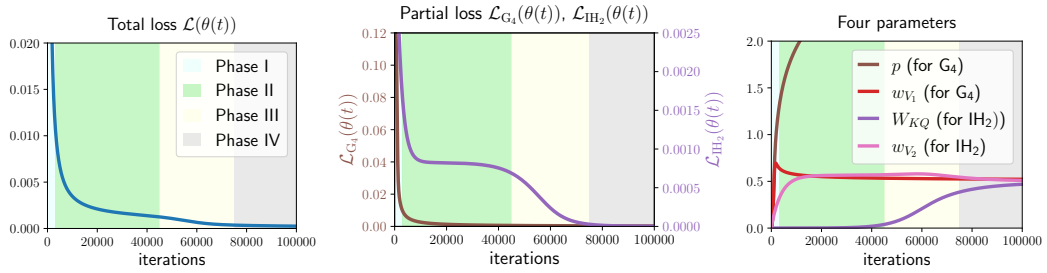

Figure 4: Visualization of the total loss, partial loss, and the parameter dynamics, for the experiment on **discrete token distribution** (Boolean, $X \sim \mathrm{Unif}(\{\pm 1\}^L)$) in our toy setting with $\alpha^\star = 1, w^\star = 0.49, \sigma_{\mathrm{init}} = 0.01, L = 40$. The figure clearly shows that transformer learns the 4-gram component first and then, starts to learn the induction head mechanism. Notably, the entire dynamics exhibit four phases. These results are **extremely similar** to that observed with Gaussian inputs, as shown in Figure 2.

3. **Adam in high-dimensional toy setting.**

Setup. We modified the setup for Figure 2 to employ a high-dimensional model ($D = 100$). Specifically, the target is $w^\star = 0.49 I_D/D$, the dot-produce parameters are $W_K, W_Q \in \mathbb{R}^D$, initialized such that $\|W_K\|_F, \|W_Q\|_F = \sigma_{\mathrm{init}}$. Additionally, for the Adam optimizer, we use learning rate `5e-4`. All other experimental setups remain the same as in the setup for Figure 2.

The training dynamics are depicted in Figure 5, where, for comparison, results using GD are also presented. In both scenarios, the learning process begins with the 4-gram pattern, followed by a gradual learning phase of the induction head mechanism. Notably, within the given number of iterations, GD remains stuck in the plateau, whereas Adam successfully escapes that plateau.

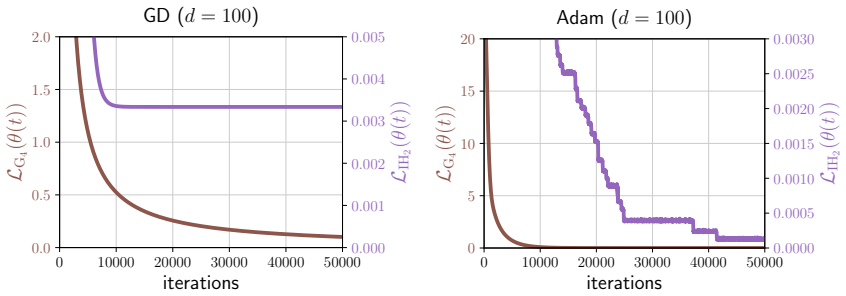

Figure 5: Partial loss for the experiment comparing **GD v.s. Adam optimizer** in high-dimensional settings ($D = 100$). In this setting, a larger D increases the difficulty of the transition from the lazy regime (learning 4-gram) to the rich regime (learning induction head). The results indicate that: (1) GD learns the 4-gram component first but becomes stuck in a plateau when learning induction head; (2) Adam, while eventually transitioning from the lazy regime (learning 4-gram) to the rich regime (learning induction head), experiences a **challenging** transition characterized by **multiple plateaus** during learning induction heads. This finding closely resembles the dynamics for GD.

E.3 Experiments supporting approximation results

1. Supporting the **necessity** of the required H and D in Theorem 4.3.

Setup. We train two-layer transformers (without FFN layers) with varying H and D to learn the generalized induction head (6) with $n = 4$. The input sequence $X = (x_1, \cdots, x_L)$ is boolean, with $x_i \overset{i.i.d.}{\sim} \mathrm{Unif}(\{\pm 1\})$ and $L = 10$. Each model is trained for 200,000 iterations using squared loss and (online) Adam optimizer with learning rate `5e-4` and batch size $B = 100$. Both layers are trained simultaneously. The results for the models with $D = H = 8$ and $D = H = 2$ are presented in Figure 3.

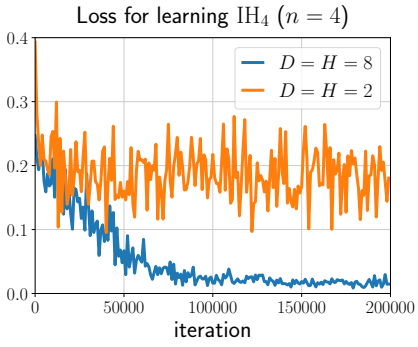

Figure 6: Results supporting the **necessity** of the required number of heads H and embedding dimension D in Theorem 4.3. We train two-layer transformers with varying H and D to learn the target in Eq. (6) with $n = 4$. The results indicate that the transformer with $H = D = 8$ ($> n$) successfully expresses this task, while the transformer with $H = D = 2$ ($< n$) fails. These results confirm that the sufficient conditions provided in Theorem 4.3 ($H \gtrsim n$ and $D \geq nd$, where $d = 1$ in our setting) are also nearly necessary.

2. Supporting our **construction** in Theorem 4.3.

Setup. We linear probing experiments (Alain and Bengio, 2016) on the transformers with $H = D = 8$ trained in the above experiment (Figure 6). For each checkpoint model TF, we denote its output in the first layer on the input sequence X as $\mathrm{TF}^{(1)}(X)$. The probing loss is measured by $\mathrm{dist}\left(X_{\cdot,-n+1:\cdot}; \mathrm{TF}^{(1)}(X)\right) = \min_{P \in \mathbb{R}^{D \times n}} : \sum_{s=n}^{L} \left\| X_{s-n+1:s} - \mathrm{TF}^{(1)}_s(X) P \right\|$, where $n = 4$, $L = 10$,

and $X = (x_1, \cdots, x_L)$ is generated by $x_i \overset{i.i.d.}{\sim} \mathrm{Unif}(\{\pm 1\})$ with testing batch 1000. The results are shown in Figure 7.

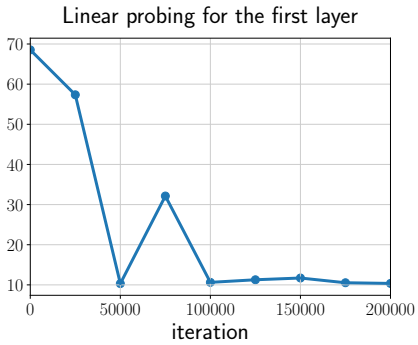

Figure 7: Probing results supporting our **construction** in Theorem 4.3. First, we train a two-layer two-layer transformer with head $H = 8$ and embedding dimension $D = 8$ to learn Eq. (6) with $n = 4$, and the checkpoints are stored during training. For each checkpoint model TF, we denote its output in the *first layer* on the input sequence X as $\mathrm{TF}^{(1)}(X)$. To validate whether it encodes the semantic information $X_{s-n+2:s}$ near each x_s, as predicted by our construction, we conduct a standard linear probing experiment (Alain and Bengio, 2016). Specifically, we measured $\mathrm{dist}\left(X_{\cdot-n+1:\cdot}; \mathrm{TF}^{(1)}(X)\right) = \min_{P \in \mathbb{R}^{D\times n}} : \sum_{s=n}^{L} \left\| X_{s-n+1:s} - \mathrm{TF}_s^{(1)}(X)P \right\|$. As the results shown, the probing loss decreases significantly during training, confirming our key construction in Theorem 4.3: **the first layer is responsible for extracting local semantic information $X_{s-n+2:s}$ near each x_s**, enabling the second layer to generate the final output.

F Detailed Comparison with Related Works

In this section, we discuss the relationship between our work and two closely related studies: Bietti et al. (2024) and Edelman et al. (2024).

Comparison with Bietti et al. (2024).

- **Approximation analysis:**
  - Bietti et al. (2024) focus primarily on the implementation of the vanilla induction head. In contrast, our study extends this analysis by investigating not only how two-layer transformers achieve vanilla induction heads (Eq. (4)) but also how they implement generalized induction heads, i.e., in-context n-grams (Eqs. (6) and (7)).
  - Furthermore, our work provides explicit approximation rate results, offering insights into the distinct roles of multiple heads, positional encoding, dot-product structure, and FFNs in implementing these induction heads.
- **Optimization analysis:**
  - *Study objective:* While Bietti et al. (2024) examines the transition from 2-gram to induction head, our work focuses on the transition from 4-gram to induction head.
  - *study methods:* Bietti et al. (2024) conducts extensive experiments supported by partial theoretical properties but does not fully characterize the training dynamics theoretically. In contrast, our study provides a precise theoretical analysis of the entire training process in a toy model, uncovering the sharp transition from 4-gram to induction head.
  - *Main insights:* Bietti et al. (2024) emphasizes the the role of weight matrices as associative memories and the impact of data distributional properties. Our analysis, on the other hand, identifies two primary drivers of the transition: (1) the time-scale separation due to low-

and high-order parameter dependencies in self-attention; (2) the speed differences caused by the relative proportions of the two components in the mixed target.

Comparison with Edelman et al. (2024). The primary connection between Edelman et al. (2024) and our work lies in the optimization analysis. Specifically, Edelman et al. (2024) focuses on the transition from uni-gram to bi-gram mechanisms in Markov Chain data. In contrast, our study investigates the transition from 4-gram to in-context 2-gram mechanisms (induction head). Additionally, we theoretically identify two primary drivers of the transition: (1) the time-scale separation due to low- and high-order parameter dependencies in self-attention; (2) the speed differences caused by the relative proportions of the two components in the mixed target.