On the Emergence of Position Bias in Transformers

Xinyi Wu [1] **Yifei Wang** [2] **Stefanie Jegelka** [3 2] **Ali Jadbabaie** [1]

Abstract

Recent studies have revealed various manifestations of position bias in transformer architectures, from the "lost-in-the-middle" phenomenon to attention sinks, yet a comprehensive theoretical understanding of how attention masks and positional encodings shape these biases remains elusive. This paper presents a graph-theoretic framework for analyzing position bias in multi-layer attention. Modeling attention masks as directed graphs, we quantify how tokens interact with contextual information based on their sequential positions. We uncover two key insights: First, causal masking inherently biases attention toward earlier positions, as tokens in deeper layers attend to increasingly more contextualized representations of earlier tokens. Second, we characterize the competing effects of the causal mask and relative positional encodings, such as the decay mask and rotary positional encoding (RoPE): while both mechanisms introduce distance-based decay within individual attention maps, their aggregate effect across multiple attention layers—coupled with the causal mask—leads to a trade-off between the long-term decay effects and the cumulative importance of early sequence positions. Through controlled numerical experiments, we not only validate our theoretical findings but also reproduce position biases observed in real-world LLMs. Our framework offers a principled foundation for understanding positional biases in transformers, shedding light on the complex interplay of attention mechanism components and guiding more informed architectural design.

1. Introduction

The attention mechanism is central to transformer architectures (Vaswani et al., 2017), which form the backbone of state-of-the-art foundation models, including large language models (LLMs). Its success lies in its ability to dynamically weigh input elements based on their relevance, enabling efficient handling of complex dependencies (Kim et al., 2017; Bahdanau et al., 2015). However, despite this widespread success, many questions remain unanswered regarding how these mechanisms process information and the artifacts they may introduce.

One particularly intriguing aspect that demands such a theoretical investigation is *position bias*, i.e., the bias of the model to focus on certain regions of the input, which significantly impacts the performance and reliability of transformers and LLMs (Zheng et al., 2023; Wang et al., 2024; Hou et al., 2024). For instance, these models often suffer from the "lost-in-the-middle" problem, where retrieval accuracy significantly degrades for information positioned in the middle of the input sequence compared to information at the beginning or end (Liu et al., 2024; Zhang et al., 2024; Guo & Vosoughi, 2024). Similarly, in-context learning is highly sensitive to the order of illustrative examples: simply shuffling independently and identically distributed (i.i.d.) examples can lead to significant performance degradation (Min et al., 2022; Lu et al., 2022; Zhao et al., 2021). Moreover, recent research has also revealed that attention sinks (Xiao et al., 2024; Gu et al., 2025; Guo et al., 2024) – positions that attract disproportionately high attention weights – arise at certain positions regardless of semantic relevance, suggesting an inherent positional bias.

These empirical findings suggest that while transformers effectively encode and process positional information through the combined use of attention masks and positional encodings (PEs) (Wang et al., 2024; Fang et al., 2025), these design elements also appear to introduce systematic positional biases, often independent of semantic content. This raises a fundamental and intriguing question about the role of positional information in attention mechanisms:

How do attention masks and positional encodings shape position bias in transformers?

[*]Equal contribution [1]MIT IDSS & LIDS [2]MIT CSAIL [3]TU Munich. Correspondence to: Xinyi Wu <xinyiwu@mit.edu>.

Accepted at Methods and Opportunities at Small Scale (MOSS), ICML 2025, Vancouver, Canada.

To address the question, we propose a novel graph-theoretic framework for analyzing attention score distributions in multi-layer attention settings. Building upon Wu et al. (2024), we model attention masks as directed graphs, enabling rigorous mathematical analysis of attention patterns. This approach proves particularly powerful for studying multi-layer attention mechanisms, as it allows us to precisely quantify how each token's contextual representation is composed from information at different positions in the sequence. By tracking the information flow through the attention layers, we can systematically examine how positional biases emerge and propagate across layers, providing insights into the complex interplay between attention masks, PEs, and the network's depth.

2. Problem Setup

Notation We use the shorthand $[n] := \{1, \dots, n\}$. For a matrix M, we denote its i-th row by $M_{i,:}$ and its j-th column by $M_{:,j}$. Throughout the analysis in the paper, we formalize the attention mask to be a directed graph $\mathcal{G}$. Formally, we represent a directed graph with N nodes by $\mathcal{G}$ and let $E(\mathcal{G})$ be the set of directed edges of $\mathcal{G}$. A directed edge $(j, i) \in E(\mathcal{G})$ from node j to i in $\mathcal{G}$ means that in the attention mechanism, token j serves as a direct context for token i or token i attends to token j. The set $\mathcal{N}_i$ of all neighbors of node i is then $\{k : (k, i) \in E(\mathcal{G})\}$.

(Masked) attention mechanism Given the representation $X \in \mathbb{R}^{N\times d}$ of N tokens, the raw attention score matrix is computed as $Z = XW_Q(XW_K)^\top/\sqrt{d_{QK}}$, where $W_Q, W_K \in \mathbb{R}^{d\times d'}$ are the query and the key matrix, respectively, and $\sqrt{d_{QK}}$ is a temperature term to control the scale of raw attention scores. Without loss of generality, we assume $d_{QK} = 1$ in our analysis. To enforce a masked attention, we create a sparse attention matrix $A \in \mathbb{R}^{N\times N}$ based on Z whose sparsity pattern is specified by a directed graph $\mathcal{G}$: we normalize Z_{ij} among all allowed token interactions $(k, i) \in E(\mathcal{G})$ such that if $(j, i) \in E(\mathcal{G})$, $A_{ij} = \mathrm{softmax}_{\mathcal{G}}(Z_{ij}) = \frac{\exp(Z_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(Z_{ik})}$, and $A_{ij} = 0$ otherwise.

For our analysis, we consider single-head (masked) self-attention networks (SANs). The layerwise update rule can be written as $X^{(t+1)} = A^{(t)}X^{(t)}W_V^{(t)}$, where $A^{(t)} = \mathrm{softmax}_{\mathcal{G}^{(t)}}(X^{(t)}W_Q^{(t)}(X^{(t)}W_K^{(t)})^\top/\sqrt{d_{QK}})$ and $W_V^{(t)} \in \mathbb{R}^{d\times d'}$ is the value matrix. For simplicity, throughout the paper, we assume that $d = d'$ and $\mathcal{G}^{(t)} = \mathcal{G}$.

Relative Positional Encoding The *decay mask* represents the relative distance between two tokens by introducing an explicit bias favoring more recent tokens. Formally, it can be written as $D_{ij} = -(i - j)m$ if $j \le i$ and 0 otherwise. Then applying the decay mask is essentially $A^{(t)}_{\text{decay}} = \mathrm{softmax}_{\mathcal{G}}(X^{(t)}W_Q^{(t)}(X^{(t)}W_K^{(t)})^\top + D)$. Note that while the decay mask formulation follows ALiBi (Press et al., 2022), it can be generalized to more complex variants such as KERPLE (Chi et al., 2022).

3. Main Results

In the transformer model, the attention mechanism is the sole module that allows tokens to interact with one another and incorporate contextual information from the sequence. It iteratively refines the contextual representation of each token across layers, allowing information to flow and accumulate based on relevance. This concept of contextualization through attention has its origins in the development of attention mechanisms, which predate transformers (Kim et al., 2017; Bahdanau et al., 2015). From the perspective of contextualization, the attention mechanism can be expressed in the following form (Kim et al., 2017):

$$X^{(t+1)}_{i,:} = \sum_{j=1}^{N} \underbrace{(A^{(t)}\cdots A^{(0)})_{ij}}_{\mathbb{P}^{(t)}(z_i=j|X^{(0)})} \cdot \underbrace{X^{(0)}_{j,:}W_V^{(0)}\cdots W_V^{(t)}}_{f^{(t)}(X^{(0)}_{z_i,:})}, \tag{1}$$

where z_i is a categorical latent variable with a sample space $\{1, \dots, N\}$ that selects the input $X_{j,:}$ to provide context for token i. In this formulation, $A^{(t)}$ represents the attention matrix at layer t, $\mathbb{P}^{(t)}(z_i = j \mid X^{(0)})$ denotes the cumulative probability of selecting input token j as the context for token i at depth t , and $f^{(t)}(\cdot)$ is a learned transformation function.

This probabilistic formulation reveals two key aspects of the attention mechanism: it acts as both a context selector and a feature aggregator. As a selector, it assigns probabilities $\mathbb{P}^{(t)}$ that quantify the relevance of each token j to target token i at depth t. As an aggregator, it combines these selected contexts weighted by their respective probabilities $\mathbb{P}^{(t)}$ to form the contextualized representation $X^{(t)}$. Since position bias fundamentally manifests as systematic preferences in how

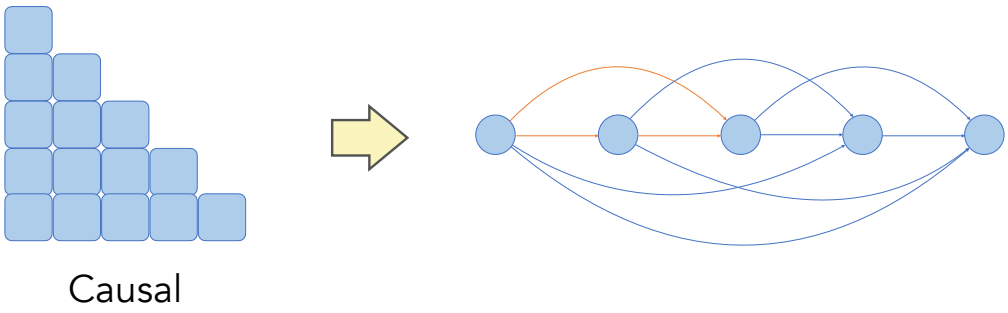

Figure 1. Causal graph and its corresponding directed graph used in the analysis (self-loops are omitted for clarity). A directed edge from token j to i indicates that i attends to j.

tokens select and incorporate context from different positions, analyzing the attention mechanism's behavior is crucial for understanding these biases. By examining how attention masks and PEs affect the probability distribution $\mathbb{P}^{(t)}$, we can investigate how position-dependent patterns emerge and propagate through multi-layer attention in transformers.

Finally, we adopt the following assumptions in our analysis: **A1** There exists $C \in \mathbb{R}$ such that $\max_{t\in\mathbb{N}}\{\|W_Q^{(t)}\|_2, \|W_K^{(t)}\|_2\} \leq C$. **A2** The sequence $\{\|\prod_{t=0}^{k} W_V^{(t)}\|_2\}_{k=0}^{\infty}$ is bounded. In particular, **A1** assumes that the key and query weight matrices are bounded, which is crucial for efficient attention computation in practice (Alman & Song, 2023), whereas **A2** is to ensure boundedness of the node representations' trajectories $X^{(t)}$ for all $t \geq 0$ (Wu et al., 2024).

3.1. Attention Masks: A Graph-Theoretic View

We first analyze the case without PEs, focusing on the effect of attention masks. A graph-theoretic perspective offers a powerful framework for analyze multi-layer attention: the flow of attention across tokens can be represented as paths in a directed graph defined by the mask, where each path captures how information is transmitted between tokens (see Figure 1 for an illustration). The number of steps in a path corresponds to the number of layers. By accounting for all such paths, we can quantify the cumulative influence of each token in the context computation of other tokens. Our first result states that for a causal mask $\mathcal{G}$, as tokens in deeper layers attend to increasingly more contextualized representations of earlier tokens, the context of each token converges exponentially toward the first token in the sequence.

Theorem 3.1. *Let $\mathcal{G}$ be the causal mask. Under* **A1**-**A2**, *given $X^{(0)} \in \mathbb{R}^{N\times d}$, for every token $i \in [N]$, $\lim_{t\to\infty} \mathbb{P}^{(t)}(z_i = 1|X^{(0)}) = 1$. Moreover, there exist $0 < C, \epsilon < 1$ where $N\epsilon < 1$ such that $\mathbb{P}^{(t)}(z_i = j|X^{(0)}) \leq C(1-(j-1)\epsilon)^t$, for all $1 < j \leq i$ and $t \geq 0$.*

3.2. Relative PEs: A Competing Decay Effect

Having analyzed how attention masks bias the model toward the beginning of the sequence, we now shift our focus to studying PEs, the other key mechanism for representing positional information in transformers. As the name suggests, relative PEs incorporate positional information by modifying the original attention scores in a way that reflects the relative positions of tokens. Among these, the decay mask (Press et al., 2022) explicitly introduces a distance-based decay effect into the attention mechanism. We begin by examining the effect of the decay mask on individual attention layers.

Lemma 3.2. *Consider the decay mask in Section 2 where $\mathcal{G}$ is causal. Under* **A1**-**A2**, *given $X^{(0)} \in \mathbb{R}^{N\times d}$, there exists $C_{\max}, C_{\min} > 0$ such that $C_{\min}e^{-(i-j)m} \leq (A^{(t)}_{\text{decay}})_{ij} \leq C_{\max}e^{-(i-j)m}$, for all $j \leq i \in [N]$ and $t \geq 0$.*

Lemma 3.2 demonstrates that the decay mask introduces an exponential decay effect into each attention map, with the strength of the effect determined by the token distances. However, while this result characterizes the behavior of individual attention layers, the interaction between layers in a multi-layer setting leads to more intricate behaviors. Building on Lemma 3.2 , Theorem 3.3 examines the cumulative effect of the decay mask across multiple layers when combined with the causal mask.

Theorem 3.3. *Consider the decay mask in Section 2 where $\mathcal{G}$ is causal. Fix $T \geq 0$. Under* **A1**-**A2**, *given $X^{(0)} \in \mathbb{R}^{N\times d}$, it holds for all $j \leq i \in [N]$ and $t \leq T$, $\mathbb{P}^{(t)}_{\text{decay}}(z_i = j|X^{(0)}) = \Theta\left(\binom{t+i-j}{i-j}e^{-(i-j)m}\right)$.*

Notably, if we denote $L(x) = \log(\binom{t+x}{x}e^{-xm})$, then $L(x)$ is not a monotone function of the distance x between two

tokens. More precisely, under Stirling's approximation, the critical point, where the highest attention score occurs, is at $x^* = t/(e^m - 1)$. This means that increasing the decay strength m decreases x^*, making the model more biased towards recent tokens, whereas increasing the number of attention layers increases x^*, making the model more biased towards initial tokens. This trade-off between layer-wise decay and cross-layer accumulation transforms the initially monotonic decay pattern within each attention map into a more intricate, non-monotonic behavior when aggregated throughout the network.

4. Experiments

In this section, we validate our theoretical findings via carefully designed numerical experiments. To ensure a controlled setup that enables precise manipulation of positional biases in the data, we adopt the synthetic data-generating process and simplified self-attention network framework proposed in Reddy (2024).

Task structure Following Reddy (2024), we adopt the following information retrieval task: The model is trained to predict the label y_{query} of a target x_{query} using the cross-entropy loss, given an alternating sequence of n items and n labels: $x_1, y_1, \ldots, x_n, y_n, x_{\text{query}}$. The sequence is embedded in d dimensions. Each x_i is sampled from a Gaussian mixture model with K classes, and y_i is the corresponding class label assigned prior to training from the total L labels ($L \leq K$). The burstiness B is the number of occurrences of x_i from a particular class in an input sequence. Importantly, at least one item in the context belongs to the same class as the query. To control position bias in the training data, x_{query} can either be explicitly assigned to the class of a specific x_i, introducing position-dependent bias in the data, or randomly assigned to the class of any x_i, simulating a scenario without position bias in the data. Following Reddy (2024), we set $n = 8$ and $d = 64$.

Tracking position bias To quantify position bias, we evaluate model performance using sequences containing novel classes not seen during training. Specifically, by generating new class centers for the Gaussian mixture and randomly assigning one of the L existing labels to these novel classes, we ensure that the model relies on contextual information rather than memorized class features. Crucially, we can systematically vary the position of the correct answer within test sequences to measure retrieval accuracy changes, thereby isolating and quantifying position-dependent biases in the model's behavior.

Network architecture The input sequences are passed through an attention-only network followed by a classifier. Each attention layer has one attention head. The classifier is then a three-layer MLP with ReLU activations and a softmax layer which predicts the probabilities of the L labels.

4.1. The Effects of Depth and Relative PEs

To investigate the position bias arising solely from the architectural design of the attention mechanism, we use training sequences without positional bias, where the position of x_i sharing the same class as x_{query} is uniformly random in $\{1, 2, \ldots, n\}$. To evaluate the position bias in the trained model, we construct test sequences of the form $[\boldsymbol{a}, b]$. Here, the bolded term $\boldsymbol{a}$ explicitly marks the correct position, ensuring y_a matches y_{query}, while position b serves as a baseline. In these sequences, x_a and x_b are identical vectors, allowing us to control for the influence of semantic information on the model's retrieval accuracy. We then measure the retrieval accuracy gap between pairs of sequences where the content at positions a and b is identical, but the correct position varies. This gap, defined as $[\boldsymbol{a}, b] - [\boldsymbol{b}, a]$, quantifies the model's positional preference independent of semantic information. To perform this evaluation, we construct three pairs of test sets, each containing $10,000$ sequences: [**first**, middle] vs. [**middle**, first], [**first**, last] vs. [**last**, first], and [**middle**, last] vs. [**last**, middle]. Here "first" (position 1), "middle" (position $n/2$), and "last" (position n) denote fixed positions within a sequence.

Figure 2 shows the average results over five runs, where a vs. b denotes the gap $[\boldsymbol{a}, b] - [\boldsymbol{b}, a]$. The magnitude of each bar represents the size of the performance gap, and the sign of each bar reflects the direction of the bias: a positive sign indicates a bias toward earlier positions, while a negative sign indicates a bias toward later positions. We highlight several key observations. First, increasing model depth consistently amplifies the bias toward earlier parts of the sequence, regardless of the PE used. Furthermore, the decay mask indeed reduces the bias toward the beginning induced by the causal mask and increases the focus on recent tokens.

5. Conclusion

In this paper, we study position bias in transformers through a probabilistic and graph-theoretic lens, developing a theoretical framework that quantifies how positional information influences context construction across multi-layer attention. By

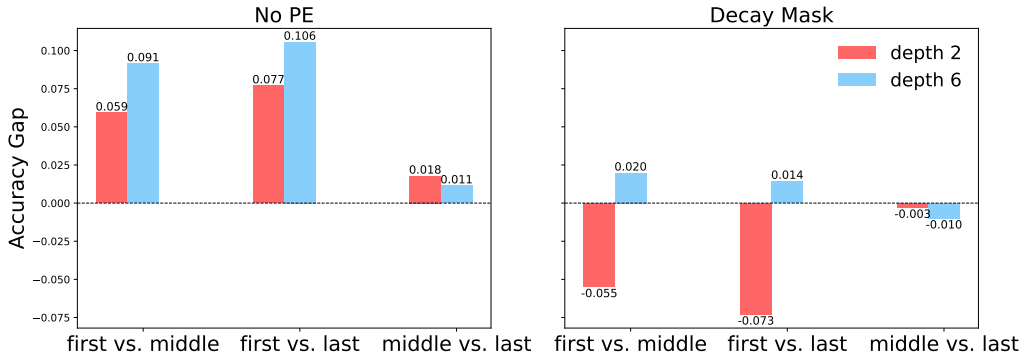

Figure 2. Position bias arising solely from the architectural design of the attention mechanism, with **no positional bias in the training data**. a vs. b denotes the gap for the case $[\boldsymbol{a}, b] - [\boldsymbol{b}, a]$, where bar magnitude indicates gap size, positive indicates bias toward earlier position, and negative indicates bias toward later position.

deepening our understanding of how architectural choices in transformers shape positional dependencies, our work provides a foundation for designing attention mechanisms with predictable and task-aligned positional properties.

References

Alman, J. and Song, Z. Fast attention requires bounded entries. In *NeurIPS*, 2023.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

Chi, T.-C., Fan, T.-H., Ramadge, P. J., and Rudnicky, A. I. Kerple: Kernelized relative positional embedding for length extrapolation. In *NeurIPS*, 2022.

Fang, L., Yifei Wang, K. G., Fang, L., and Wang, Y. Rethinking invariance in in-context learning. In *ICLR*, 2025.

Gu, X., Pang, T., Du, C., Liu, Q., Zhang, F., Du, C., Wang, Y., and Lin, M. When attention sink emerges in language models: An empirical view. In *ICLR*, 2025.

Guo, T., Pai, D., Bai, Y., Jiao, J., Jordan, M. I., and Mei, S. Active-dormant attention heads: Mechanistically demystifying extreme-token phenomena in llms. 2024.

Guo, X. and Vosoughi, S. Serial position effects of large language models. *ArXiv*, 2024.

Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., and Zhao, W. X. Large language models are zero-shot rankers for recommender systems. In *ECIR*, 2024.

Kim, Y., Denton, C., Hoang, L., and Rush, A. M. Structured attention networks. In *ICLR*, 2017.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 2024.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *ICLR*, 2019.

Lu, Y., Bartolo, M., Moore, A., Riedel, S., and Stenetorp, P. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *ACL*, 2022.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*, 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. In *ICLR*, 2022.

Reddy, G. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *ICLR*, 2024.

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Wang, Z., Zhang, H., Li, X., Huang, K.-H., Han, C., Ji, S., Kakade, S. M., Peng, H., and Ji, H. Eliminating position bias of language models: A mechanistic approach. *ArXiv*, abs/2407.01100, 2024.

Wu, X., Ajorlou, A., Wang, Y., Jegelka, S., and Jadbabaie, A. On the role of attention masks and layernorm in transformers. In *NeurIPS*, 2024.

Xiao, G., Tian, Y., Chen, B., Han, S., and Lewis, M. Efficient streaming language models with attention sinks. In *ICLR*, 2024.

Zhang, Z. A., Chen, R., Liu, S., Yao, Z., Ruwase, O., Chen, B., Wu, X., and Wang, Z. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. *ArXiv*, abs/2403.04797, 2024.

Zhao, T. Z., Wallace, E., Feng, S., Klein, D., and Singh, S. Calibrate before use: Improving few-shot performance of language models. In *ICML*, 2021.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS*, 2023.

6. Proof of Theorem 3.1

6.1. Auxiliary results

Lemma 6.1. *Under* **A1**-**A2**, *there exists* $\epsilon > 0$ *such that* $A_{ij}^{(t)} \geq \epsilon$ *for all* $t \geq 0$, $(j, i) \in E$.

Proof. Writing$X^{(t+1)} = A^{(t)} X^{(t)} W_V^{(t)}$ recursively, we get that the token trajectories

$$X^{(t+1)} = A^{(t)} ... A^{(0)} X^{(0)} W_V^{(0)} ... W_V^{(t)} \,, \tag{2}$$

stay uniformly bounded for all $t \geq 0$ by **A2**. Then it follows from **A1** that there exists $C \in \mathbb{R}$ such that for all $t \geq 0$,

$$\begin{aligned} \left\| \left(X^{(t)} W_Q^{(t)} \right)_{i,:} \right\|_2 &= \left\| X_{i,:}^{(t)} W_Q^{(t)} \right\|_2 \leq C \,, \\ \left\| \left(X^{(t)} W_K^{(t)} \right)_{i,:} \right\|_2 &= \left\| X_{i,:}^{(t)} W_K^{(t)} \right\|_2 \leq C \,. \end{aligned} \tag{3}$$

Hence for all $i, j \in [N]$,

$$-C^2 \leq (X^{(t)} W_Q^{(t)} (X^{(t)} W_K^{(t)})^\top)_{ij} \leq C^2 \,.$$

This implies that there exists $\epsilon > 0$ such that $A_{ij}^{(t)} \geq \epsilon$ for all $(j, i) \in E$. □

6.2. Proof of Theorem 3.1

We denote $P^{(t)} := A^{(t)} \cdots A^{(0)}$. It suffices to show that there exists $0 < C < 1$ and $0 < \epsilon < 1$ such that

$$P_{ij}^{(t)} \leq C(1 - (j-1)\epsilon)^t \tag{4}$$

for all $1 < j \leq i$ and $t \geq 0$.

The proof will go by induction:

Base case By Lemma 6.1, it follows that

$$P_{ij}^{(0)} \leq (1-\epsilon)$$

for all $1 < j \leq i$. Then let $C := 1 - \epsilon$.

Induction step Assume that (4) holds, it follows that for all $1 < j \leq i$.

$$P_{ij}^{(t+1)} = \sum_{k=j}^{i} A_{ik}^{(t)} P_{kj}^{(t)} \leq (1-(j-1)\epsilon)C(1-(j-1)\epsilon)^t = C(1-(j-1)\epsilon)^{t+1}\,.$$

From above, we conclude the theorem.

7. Proof of Lemma 3.2

Fix $t \geq 0$. Let

$$Z_{ij}^{(t)} = (X^{(t)}W_Q^{(t)})_{i,:}(X^{(t)}W_K^{(t)})_{:,j}.$$

Following from Lemma 6.1, there exists $I_{\min}, I_{\max} \in \mathbb{R}$ such that for all $j \leq i \in [N]$,

$$Z_{ij}^{(t)} \in [I_{\min}, I_{\max}].$$

Consider the denominator in the $\mathrm{softmax}(\cdot)$ operation in the calculation of $(A_{\text{decay}}^{(t)})_{ij}$:

$$\begin{aligned}
\sum_{k=1}^{i} e^{Z_{ik}^{(t)} - (i-k)m} &\geq e^{I_{\min}} \sum_{k=0}^{i} e^{-(i-k)m} \\
&= e^{I_{\min}} \frac{1 - e^{-(i+1)m}}{1 - e^{-m}} \\
&\geq e^{I_{\min}} \frac{1 - e^{-2m}}{1 - e^{-m}} \\
&= e^{I_{\min}}(1 + e^{-m})
\end{aligned}$$

and

$$\begin{aligned}
\sum_{k=1}^{i} e^{Z_{ik}^{(t)} - (i-k)m} &\leq e^{I_{\max}} \sum_{k=0}^{\infty} e^{-km} \\
&= \frac{e^{I_{\max}}}{1 - e^{-m}}
\end{aligned}$$

It follows that

$$(A_{\text{decay}}^{(t)})_{ij} \leq \frac{e^{I_{\max} - (i-j)m}}{e^{I_{\min}}(1 + e^{-m})} = C_{\max} e^{-(i-j)m}$$

and

$$(A_{\text{decay}}^{(t)})_{ij} \geq \frac{e^{I_{\min} - (i-j)m}}{e^{I_{\max}}/(1 - e^{-m})} = C_{\min} e^{-(i-j)m}$$

where $C_{\max} := e^{(I_{\max} - I_{\min})}/(1 + e^{-m})$ and $C_{\min} := (1 - e^{-m})e^{(I_{\min} - I_{\max})}$.

8. Proof of Theorem 3.3

Note that in the causal graph $\mathcal{G}$, there are $\binom{t+i-j}{i-j}$ paths of length $t+1$ from token j to token i.

Since going from token j to token i in the causal graph, the connectivity patterns ensure that the token indices along the path are non-decreasing, i.e. if we denote the directed path as $(j, l_1), (l_1, l_2), ..., (l_t, i)$, it holds that $j \leq l_1 \leq l_2 \leq ... \leq l_t \leq i$. Together with Lemma 3.2, we conclude the theorem statement.

9. Experiments

Here we provide more details on the numerical experiments presented in Section 4. All models were implemented with PyTorch (Paszke et al., 2019).

Parameterizing the data distribution As defined in Section 4, the input data distribution is modulated by tuning various parameters. In addition to the parameters described in the main text, for the Gaussian mixture with K classes, each class k is defined by a d-dimensional vector μ_k whose components are sampled $i.i.d.$ from a normal distribution with mean zero and variance $1/d$. Then the value of x_i is given by $\frac{\mu_k+\gamma\eta_i}{\sqrt{1+\gamma^2}}$, where η_i is drawn from the same distribution as the μ_k's and γ sets the within-class variability. Each class is assigned to one of L labels ($L \leq K$). The contents of the labels are drawn prior to training from the same distribution as the μ_k's.

In Reddy (2024), the author found that different configurations of the data generating process give rise to different learning regimes. To enable better information retrieval ability of the model, we choose the configuration suggested by Reddy (2024) that corresponds to the difficult in-weight learning and easy in-context-learning regime to ensure the information retrieval ability of the model. Specifically, we set $\gamma = 0.75$, $K = 2048$, $L = 32$, and $B = 4$.

Relative PE hyperparameters For the decay mask, we set $m = -\log(0.8)$.

Compute We trained all of our models on a Tesla V100 GPU.

Training details In all experiments, we used the AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of 10^{-3}, a weight decay of 10^{-6}, a batch size of 128, and trained for $100,000$ iterations.