How Transformers Learn Causal Structures In-Context: Explainable Mechanism Meets Theoretical Guarantee

Anonymous authors
Paper under double-blind review

Abstract

Transformers have demonstrated remarkable in-context learning abilities, adapting to new tasks from just a few examples without parameter updates. However, theoretical understanding of this phenomenon typically assumes fixed dependency structures, while real-world sequences exhibit flexible, context-dependent relationships. We address this gap by investigating whether transformers can learn causal structures – the underlying dependencies between sequence elements – directly from in-context examples. We propose a novel framework using Markov chains with randomly sampled causal dependencies, where transformers must infer which tokens depend on which predecessors to make accurate predictions. Our key contributions are threefold: (1) We prove that a two-layer transformer with relative position embeddings can implement Bayesian Model Averaging (BMA), the optimal statistical algorithm for causal structure inference; (2) Through extensive experiments and parameter-level analysis, we demonstrate that transformers trained on this task learn to approximate BMA, with attention patterns directly reflecting the inferred causal structures; (3) We provide information-theoretic guarantees showing how transformers recover causal dependencies and extend our analysis to continuous dynamical systems, revealing fundamental differences in representational requirements. Our findings bridge the gap between empirical observations of in-context learning and theoretical understanding, showing that transformers can perform sophisticated statistical inference over structural uncertainty.

1 Introduction

Modern transformers exhibit a remarkable capability: they can adapt to entirely new tasks using only a handful of examples, without any parameter updates. This phenomenon, known as in-context learning (ICL) Brown et al. (2020), has revolutionized our understanding of what neural networks can achieve. A model trained on diverse text can suddenly perform arithmetic, translate languages, or write code – all by simply observing a few demonstrations. Yet despite extensive empirical success Wei et al. (2022); Garg et al. (2023) and theoretical investigations von Oswald et al. (2023); Akyürek et al. (2023); Goel & Bartlett (2024), a fundamental question remains: how do transformers adapt to the varying dependency structures present in real-world sequences? (Allen-Zhu & Li, 2023; Bietti et al., 2023; Zhao et al., 2023; Wibisono & Wang, 2024)

The Theory-Practice Gap. Current theoretical understanding of ICL rests on a critical simplification: most analyses assume that dependencies between sequence elements follow a fixed, predetermined structure. For instance, theoretical works typically study settings where tokens are independent $[[x_1, f(x_1)], [x_2, f(x_2)], ...]$ or follow rigid patterns like $[x_1, f(x_1), x_2, f(x_2)]$ (Bai et al., 2023; Chen et al., 2024a; Wang et al., 2025). However, natural language and real-world sequences exhibit far richer structure – words depend on previous words in complex, context-dependent ways that vary across sentences and domains. Recent work by Nichani et al. (2024) began addressing this by showing transformers can encode fixed causal structures during training. Specifically, they assume an n-gram causal model (e.g., bigrams where each token depends only on the previous one) (Rajaraman et al., 2024; Edelman et al., 2024), and prove that transformers can embed this structure in their attention weights to perform inference. However, in real-world scenarios, the dependency

graph itself is not fixed but varies across different sequences. For example, in language, the syntactic structure can change dramatically between different documents, and in stock price prediction, the relationships between assets can shift over time. Thus, a key challenge is

Can transformers infer and adapt to causal structure in-context? ($\star$)

Our Approach. We introduce a novel framework where sequences are generated from Markov chains with randomly sampled causal dependencies. In our setting, each token depends on exactly one predecessor, or its "parent", but crucially, these parent relationships are not fixed and must be inferred from context examples, which is a collection of sequences sharing the same underlying causal structure. This setup captures the essence of ($\star$) by requiring the model to adapt to different latent structures across contexts. The transformer must infer these latent dependencies from context examples to accurately predict new sequences – mirroring how language models must adapt to different syntactic structures or reasoning patterns.

Main Contributions. We consider two types of Markov chains: discrete chains over a finite vocabulary and continuous linear dynamical systems. Our work makes the following contributions:

(1) Theoretical Construction: For discrete Markov chains, we prove that a two-layer transformer with relative position embeddings can implement Bayesian Model Averaging (BMA), the statistically optimal algorithm for inferring causal structures from observations. Our construction shows how attention mechanisms can perform sophisticated probabilistic inference over structural uncertainty. **(2) Empirical Verification:** Through extensive experiments on Markov chains, we demonstrate that transformers trained via gradient descent converge to solutions remarkably similar to our theoretical construction. Parameter-level analysis reveals that learned attention patterns directly encode posterior probabilities over causal structures, providing mechanistic insight into how transformers perform statistical inference. **(3) Information-Theoretic Analysis:** We establish conditions under which causal structures can be recovered in-context, using mutual information and data processing inequalities. Additionally, we show that gradient-based learning naturally discovers these structures early in training through χ^2-mutual information maximization. **(4) Extensions to Continuous Systems:** We extend our framework to linear dynamical systems in continuous space, revealing fundamental differences in how transformers handle discrete versus continuous causal inference. While transformers show strong empirical performance, we identify representational limitations that prevent exact BMA implementation in continuous settings.

Paper Organization. Section 2 introduces our problem formulation and model architecture. Section 3 presents our main theoretical and empirical results for Markov chains. Section 4 extends the analysis to continuous dynamical systems. Appendix A discusses related work.

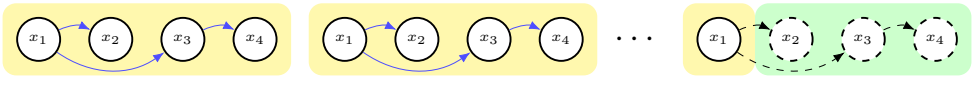
2 Preliminary

2.1 Task Setup

To investigate the question ($\star$), we consider data are generated from distributions with a latent causal structure. Each sample is a sequence of tokens $\boldsymbol{x}_{1:H} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_H]$, where the h-th token $\boldsymbol{x}_h$ depends on one of its predecessors, called the parent token $\boldsymbol{x}_{\mathrm{pa}(h)}$. This dependency relation is represented as a directed tree graph $\mathcal{G} = \{\mathrm{pa}(h)\}_{h\in[H]}$, where $\mathrm{pa}(h) \sim \mathrm{Unif}(1, \ldots, h-1), \forall h \in 2, \ldots, H$. Given the causal structure defined above, the generative process can be written as $\boldsymbol{x}_h = G(\boldsymbol{x}_{\mathrm{pa}(h)})$, where $G(\cdot)$ denotes either stochastic sampling from the transition kernel $\pi(\cdot|\boldsymbol{x}_{\mathrm{pa}(h)})$ of Markov chains, or a deterministic transformation with additive Gaussian noise in dynamical systems. $G(\cdot)$ is fixed during sampling the whole dataset.

For the in-context learning task, suppose we have $L+1$ samples $\{\mathbf{x}_{1:H}^{(l)}\}_{l\in[L+1]}$ from the same causal graph $\mathcal{G}$, the first L samples are provided as in-context demonstrations from which the model may infer the latent graph structure, while the last sample is the target for prediction. Except the first token $\boldsymbol{x}_1^{L+1}$, every token $\boldsymbol{x}_h^{L+1}$ in this trajectory is required to be predicted via next-token prediction conditioned on $\mathbf{x}_{1:H}^{1:L}$ and its past observations $\boldsymbol{x}_{1:h-1}^{L+1}$.

Markov Chain. Following Markovian assumption adopted in Edelman et al. (2024); Nichani et al. (2024); Chen et al. (2024b); D'Angelo et al. (2025), we assume sequences are sampled from



Example 1 **Example 2** **Example $L+1$**

Figure 1: **Task overview of in-context causal structure learning.** Each training sequence consists of L examples with observed variables and hidden parent relations, followed by a new example $L+1$ where the model must infer the underlying parent indices in-context from previous demonstrations.

a Markov chain with random dependencies. In this setting, Tokens $\{\mathbf{x}_h\}$ are drawn from a finite vocabulary $\mathcal{V} = \{e_1, \ldots, e_d\}$, where $|\mathcal{V}| = d$ and $\{\boldsymbol{e}_i\}$ are one-hot vectors. The random dependencies are specified by latent causal graph $\{\mathrm{pa}(h)\}_{h\in[H]}$. Let $\pi : \mathcal{V} \to \Delta(\mathcal{V})$ denote the Markov transition kernel, where $\Delta(\mathcal{V})$ is the probability simplex over $\mathcal{V}$. Then each token is generated as $\mathbf{x}_h \sim \pi(\cdot \mid \mathbf{x}_{\mathrm{pa}(h)}) \in \Delta(\mathcal{V})$, $\forall h \in [H]$, where by slight abuse of notation, we also regard π as the stochastic matrix $\pi \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ with $\pi[i,j] = \pi(j|i)$, $\sum_j \pi[i,j] = 1$.

Dynamical System Beyond the discrete Markov chain case, we also consider a more challenging setting with continuous sampling space. Here tokens $\{\boldsymbol{x}_h\}$ are dense vectors in $\mathbb{R}^d$. The link function $g(\cdot)$ replaces the discrete transition kernel, and we instantiate it as a *linear dynamical system with additive Gaussian noise*: $\boldsymbol{x}_h = g(\boldsymbol{x}_{\mathrm{pa}(h)}) = \frac{1}{c}\big(A^\top \boldsymbol{x}_{\mathrm{pa}(h)} + \varepsilon_h\big)$, where $A \in \mathcal{O}(\mathbb{R}^d)$ is orthogonal, $\boldsymbol{x}_1 \sim \mathcal{N}(0, I_d)$, $\varepsilon_h \sim \mathcal{N}(0, \sigma^2 I_d)$, and $c^2 = 1 + \sigma^2$ ensures variance stability.

These settings evaluate the extent to which transformers can perform in-context causality learning.

Goal: Inferring the Causal Structure. The task formulation naturally raises the following question: *Given L in-context examples, how can the model infer the underlying graph structure $\mathcal{G}$?* A classical approach to this problem is *Bayesian Model Averaging* (BMA), which leverages Bayes' rule to compute the posterior distribution over possible parameter space. Treating the parent structure $\mathrm{pa}(h)$ as the parameter to be estimated, the distribution of having parent h' will be predicted as its posterior probability given L observations:

$$\mathbb{P}\big(\mathrm{pa}(h) = h' \,\big|\, \boldsymbol{x}_{1:H}^{1:L}\big) = \frac{\mathbb{P}\big(\boldsymbol{x}_{1:H}^{1:L} \,\big|\, \mathrm{pa}(h) = h'\big)\ \mathbb{P}(\mathrm{pa}(h) = h')}{\sum_{h''\in[H]} \mathbb{P}\big(\boldsymbol{x}_{1:H}^{1:L} \,\big|\, \mathrm{pa}(h) = h''\big)\ \mathbb{P}(\mathrm{pa}(h) = h'')}. \tag{1}$$

By Eq. (1) and our task assumption, we have the following lemma of the formulation of BMA.

Lemma 1. *Suppose L samples are observed from the above Markov chain (or dynamical system) $\mathrm{x}_{1:H}$ with latent causal structure $\mathcal{G}$. Bayesian Model Averaging give prediction of $pa(h) \in [h-1]$:*

$$\mathbb{P}(pa(h) = h' \mid \boldsymbol{x}_{1:H}^{1:L}) = \frac{\exp\big(\sum_{l\in[L]} \log \pi(\boldsymbol{x}_h^l | \boldsymbol{x}_{h'}^l)\big)}{\sum_{h''\in[h-1]} \exp\big(\sum_{l\in[L]} \log \pi(\boldsymbol{x}_h^l | \boldsymbol{x}_{h''}^l)\big)} = \sigma\big(\hat{\boldsymbol{p}}^{h,L}(\log\pi)\big)_{h'}, \tag{2}$$

where $\hat{\boldsymbol{p}}^{h,L}(\log\pi) \in \mathbb{R}^d$ and $\hat{\boldsymbol{p}}_{h'}^{h,L} = \sum_{l\in[L]} \log \pi(\boldsymbol{x}_h^l | \boldsymbol{x}_{h'}^l)$. See Appendix C.2 for detailed proof.

This Bayesian formulation provides a principled baseline for inferring causal structure, and serves as a point of comparison for the in-context learning behavior of transformers.

2.2 Model Architecture

2.2.1 Standard Transformer

Decoder-only Transformer is a neural network structure dealing with sequential data. Given a sequence of tokens $(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T)$, transformers first embed tokens and add a positional encoding to the tokens: $\boldsymbol{h}_t^{(0)} = E(\boldsymbol{w}_t) + P(t) \in \mathbb{R}^d, \forall t \in [T]$. In a matrix form, the mapped tokens of input is $\boldsymbol{H}^{(0)} = \boldsymbol{h}_{1:T}^{(0)\top} \in \mathbb{R}^{T\times d}$. Subsequent layers consist of multi-head attention layers (MHA) followed by multilayer perceptron layers (MLP). At layer l, the hidden features $\boldsymbol{H}^{(l-1)}$ will be updaed as follows. First, causal-mask self-attention layer will compute the ouput by:

$$\mathrm{Attn}(\boldsymbol{H}; \boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V) = \sigma\Big(\mathcal{M}\Big(\frac{(\boldsymbol{H}\boldsymbol{W}_Q)(\boldsymbol{H}\boldsymbol{W}_K)^\top}{\sqrt{d_k}}\Big)\Big)\boldsymbol{H}\boldsymbol{W}_V,$$

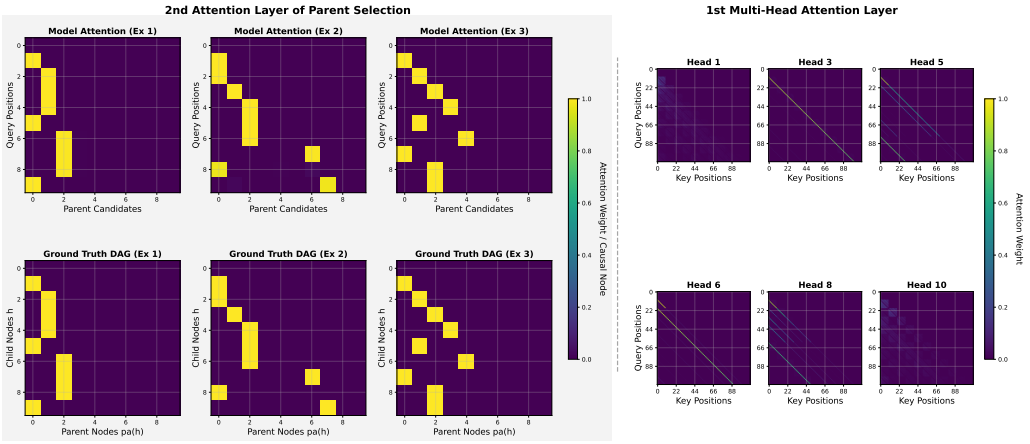


Figure 2: Visualization of Attention Weights $\mathcal{A}^{(1)}, \mathcal{A}^{(2)}$. Left: $\mathcal{A}^{(2)}$ on 3 examples. Transformer shows promising capabilities to select causal tokens in context. Right: Six representative heads (out of 10) from $\mathcal{A}^{(1)}$ (Head 1 and 10 degenerate). (trained with $L = 10, H = 10, d = 5$, 1024 steps).

where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d\times d_k}$, $\sigma(\boldsymbol{v})_i = \frac{\exp(v_i)}{\sum_j \exp(v_j)}$ applied to matrix row-wisely, $\mathcal{M}$ is the causal mask where $\mathcal{M}(\boldsymbol{X})_{ij}$ is $-\infty$ if $i > j$ else $\boldsymbol{X}_{ij}$. Then multi-headed attention gives the output:

$$\mathrm{MHA}(\boldsymbol{H}) = \left(\bigoplus_{m=1}^{M} \mathrm{Attn}(\boldsymbol{H}; \boldsymbol{W}_Q^m, \boldsymbol{W}_K^m, \boldsymbol{W}_V^m)\right)\boldsymbol{W}_O,$$

where $\bigoplus$ denotes the concatenation of vectors and $\boldsymbol{W}_O \in \mathbb{R}^{Md_k\times d}$. Getting intermediate features $\mathrm{MHA}_l(\boldsymbol{H}^{(l-1)})$ from attention layer, this feature will be added to the **residual stream** which aggregate the previous output: $\hat{\boldsymbol{H}}^l = \boldsymbol{H}^{(l-1)} + \mathrm{MHA}_l(\boldsymbol{H}^{(l-1)})$. FFN layer adopts this as input and updates this stream as:

$$\mathrm{FFN}(\hat{\boldsymbol{H}}) = \sigma\big(\hat{\boldsymbol{H}}\boldsymbol{W}_1\big)\,\boldsymbol{W}_2, \quad \boldsymbol{H}^{(l)} = \hat{\boldsymbol{H}}^{(l)} + \mathrm{FFN}_l(\hat{\boldsymbol{H}}^{(l)}),$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{d\times d_m}, \boldsymbol{W}_2 \in \mathbb{R}^{d_m\times d}$ and $\sigma(\cdot)$ is the activation function. Finally the output of L-layer Transformer is $\sigma(\boldsymbol{H}^{(L)}\boldsymbol{W}_U)$ projected to vocabulary logits by $\boldsymbol{W}_U \in \mathbb{R}^{d\times V}$.

2.2.2 Disentangled Transformers

To better analyze the role each part of transformers play in learning a task, prior works Friedman et al. (2023) propose the disentangled transformer which decouples the twisted features in the residual stream. Instead of adding each layer's output, disentangled transformer concentenates it with residual stream. Considering the decoder-based attention-only transformers we will mainly focus on, it will update the hidden states $\boldsymbol{H}^{(l-1)} \in \mathbb{R}^{T\times d_{l-1}}$ by

$$\boldsymbol{H}^{(l)} = [\boldsymbol{H}^{(l-1)}, \mathrm{Attn}_1(\boldsymbol{H}^{(l-1)}), \ldots, \mathrm{Attn}_M(\boldsymbol{H}^{(l-1)})] \in \mathbb{R}^{T\times(1+M)d_{l-1}}, \tag{3}$$

where in each attention head, $\boldsymbol{W}_K\boldsymbol{W}_Q^\top$ is reparametrized by $\boldsymbol{W}_{KQ}$, $\boldsymbol{W}_O\boldsymbol{W}_V$ by $\boldsymbol{W}_{OV}$ and the initial input $\boldsymbol{H}^{(0)} \in \mathbb{R}^{T\times d_0}$ is given by $\boldsymbol{h}_t^{(0)} = [E(\boldsymbol{w}_t), P(\boldsymbol{w}_t)] = [\boldsymbol{e}_{\boldsymbol{w}_t}, \boldsymbol{e}_t] \in \mathbb{R}^{d+T}, \forall t$. Consider in our task, the input sequence consists of $L+1$ examples of length-H chains, leading to the embedding size T of $P(\boldsymbol{w}_t) \in \mathbb{R}^T$ equal to $(L+1)H$. If we set vocabulary dimension $d = H = L = 10$, then $d = 10 \ll T = 110$ in the input embedding and $\boldsymbol{W}_{KQ}$ in the 1st layer will have $\Theta(H^2L^2) = \Theta(10^4)$ parameters. Instead, considering $\boldsymbol{w}_t$ is the h-th token in example l, we use two types of embeddings representing this: $\mathrm{Pos}_L(\boldsymbol{w}_t) = \boldsymbol{e}_l \in \mathbb{R}^L$, $\mathrm{Pos}_H(\boldsymbol{w}_t) = \boldsymbol{e}_h \in \mathbb{R}^H$. This reduces the required parameter for training. And the formulation of this transformer is given by Eq. (34). Empirical results presented in Appendices G, H and I demonstrate that the standard disentangled transformer with full absolute position encoding, its positional variants and standard transformers with FFNs all behave consistently with the model described below. To enable a tractable parameter-level analysis, we adopt a simplified structure that reduces the parameter count and simplifies training.

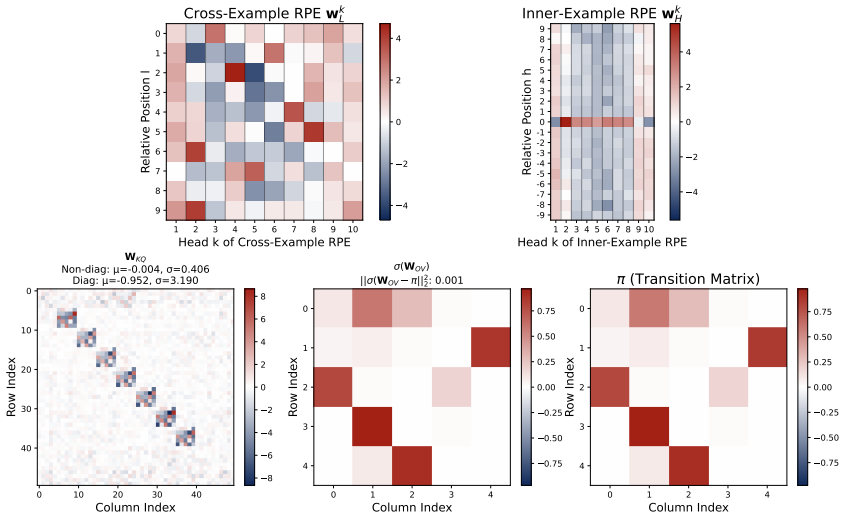


Figure 3: Parameter visualization of 2-layer transformer with RPE. Heads of inner-example RPE $\mathbf{w}_H^k$ uniformly show the largest value at position $h = 0$ except Head 1, 9, 10. Correspondingly, $\boldsymbol{W}_{KQ}$ shows similar blocks on diagonal except block 1, 9, 10. Besides, $\sigma(\boldsymbol{W}_{OV})$ approximates π.

Relative Position Embedding. While the original Transformer employs the above absolute positional encodings, subsequent research has demonstrated the advantages of relative positional embeddings (RPE) Shaw et al. (2018); Su et al. (2023). RPE is parametrized by a vector $\mathbf{w} \in \mathbb{R}^T$ which assigns attention score $\mathbf{w}(i,j)$ only via relative distance $i-j$ between positions of query i and key j. Similar to the above absolute embedding, we adopt two types of RPE: $\mathbf{w}_H \in \mathbb{R}^{2H-1}$ representing the order l from $L+1$ examples and $\mathbf{w}_L \in \mathbb{R}^L$ representing the order h from H tokens.

$$\mathbf{w}_H(h,h') = \mathbf{w}_H[h-h'],\ \forall (h,h') \in [H]^2, \quad \mathbf{w}_L(l,l') = \begin{cases} \mathbf{w}_L[l-l'], & l > l', \\ -\infty, & \text{else.} \quad \text{(for causal mask)} \end{cases}$$

Attention with RPE. With RPE in the 1st layer, the output $\boldsymbol{u}_t$ for $\boldsymbol{x}_t = \boldsymbol{x}_h^l$ is given by:

$$\begin{aligned} \boldsymbol{u}_t = \mathrm{Attn}_{\boldsymbol{x}_t \to \boldsymbol{x}_{1:T}} &= \sum_{t'} \sigma_{t'}(\mathbf{w}_H(h,\cdot) + \mathbf{w}_L(l,\cdot))\boldsymbol{x}_{t'} \\ &= \sum_{t' \leftrightarrow (h',l')} \frac{\exp(\mathbf{w}_H(h,h') + \mathbf{w}_L(l,l'))}{\sum_{t''} \exp(\mathbf{w}_H(h,h'') + \mathbf{w}_L(l,l''))} \boldsymbol{x}_{t'}. \end{aligned} \tag{4}$$

Self-Attention Layer. Suppose the 1st layer use K heads, for the 2nd layer, the input is $\boldsymbol{v}_t = \boldsymbol{H}_t^{(1)} \in \mathbb{R}^{d+Kd}$ given by disentangled residual in Eq. (3). The features of last example $L+1$: $\boldsymbol{v}_h^{L+1}$ are taken as query, key and value tokens into the attention layer. The output gives Transformer's prediction. Suppose the input is $\boldsymbol{x}_{1:T} = \boldsymbol{x}_{1:H}^{1:L+1}$, this transformer architecture is formulated as follows:

$$\begin{aligned} &\textbf{1st RPE Attention (K-head):} && \boldsymbol{u}_h^k = \mathrm{Attn}^k_{\boldsymbol{x}_h^{L+1} \to \boldsymbol{x}_{1:T}} = \sigma(\mathbf{w}_H^k(h,\cdot) + \mathbf{w}_L^k(L+1,\cdot))\boldsymbol{x}_{1:T}^\top && \in \mathbb{R}^d, \\ &\textbf{Disentangled Residual:} && \boldsymbol{v}_h = [\boldsymbol{u}_h^1, \ldots, \boldsymbol{u}_h^K],\ \ \boldsymbol{z}_h = [\boldsymbol{x}_h^{L+1}, \boldsymbol{v}_h] && \in \mathbb{R}^{d+Kd}, \\ &\textbf{2nd Attention (1-head):} && \boldsymbol{f}_{\mathtt{tf}}(\cdot \mid \mathcal{H}_h^L) = \sigma\left(\boldsymbol{z}_{1:h-1}^\top \boldsymbol{W}_{KQ} \boldsymbol{z}_h\right)^\top \boldsymbol{z}_{1:h-1}^\top \boldsymbol{W}_{OV} \\ & && \qquad = \sigma(\boldsymbol{v}_{1:h-1}^\top \boldsymbol{W}'_{KQ} \boldsymbol{v}_h)^\top \boldsymbol{x}_{1:h-1}^{L+1\ \top} \boldsymbol{W}'_{OV} && \in \mathbb{R}^d, \end{aligned} \tag{5}$$

where $f_{\mathtt{tf}}(\cdot \mid \mathcal{H}_h^L) \in \mathbb{R}^d$ denotes the ouput of the transformer based on context $\mathcal{H}_h^L = [\boldsymbol{x}_{1:H}^{1:L}, \boldsymbol{x}_{1:h-1}^{L+1}]$ (or the context denoted by $\mathcal{H}$ for brevity) and we assume some blocks in $\boldsymbol{W}_{KQ}, \boldsymbol{W}_{OV}$ are 0:

$$\boldsymbol{W}_{KQ} = \begin{bmatrix} 0_{d\times d} & 0_{d\times Kd} \\ 0_{Kd\times d} & \boldsymbol{W}'_{KQ} \end{bmatrix}, \boldsymbol{W}_{OV} = \begin{bmatrix} \boldsymbol{W}'_{OV} & 0_{d\times Kd} \\ 0_{Kd\times d} & 0_{Kd\times Kd} \end{bmatrix}, \tag{6}$$

where $\boldsymbol{W}'_{KQ} \in \mathbb{R}^{Kd\times Kd}$, $\boldsymbol{W}'_{OV} \in \mathbb{R}^{d\times d}$ are trainable and this simplification is supported by the results on disentangled transformers in Appendix G. To train transformers, cross-entropy loss is used for $\boldsymbol{x} \in \mathcal{V}$ of Markov chain (MC) and MSE loss for dyanmical system (DS) shown in Eq. (15).

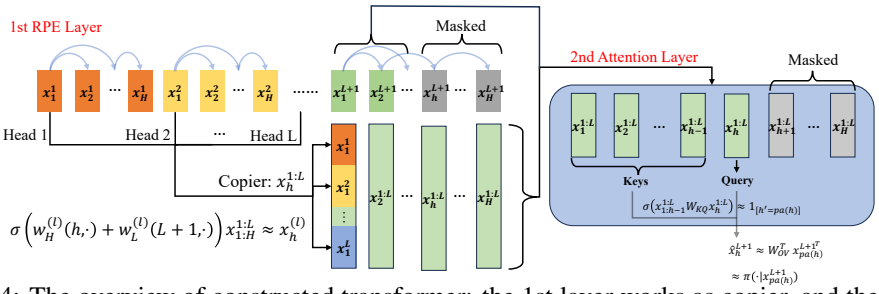


Figure 4: The overview of constructed transformer: the 1st layer works as copier, and the attention patterns in 2nd layer follows BMA, which approximately select correct parent token.

3 Can Transformers In-Context Learn Causal Structures?

3.1 2-Layer Transformer Learnt to Select Causal Structure In-Context

To investigate the question ($\star$), we first train 2-layer transformers with RPE introduced above on the Markov chain setting. Each input has $L+1$ samples $\{\boldsymbol{x}_{1:H}^l\}$ of Length-H Markov chain with causal structure $\mathcal{G}$. The input sequence length is $T = H(L+1)$. We set the transformer has K heads in the first RPE layer: $\{(\mathbf{w}_H^k, \mathbf{w}_L^k)\}_{k\in[K]}$ and 1 head for the 2nd attention layer $(\boldsymbol{W}_{KQ}, \boldsymbol{W}_{OV})$. All RPE parameters are intialized randomly from Gaussian distribution and $(\boldsymbol{W}_{KQ}, \boldsymbol{W}_{OV})$ from zero.

For an attention layer, the attention weights $\mathcal{A}$ normalized by the σ reveal which tokens a query primarily attends to, enabling mechanistic interpretability analyses such as circuit discovery Olsson et al. (2022). We first look at the attention patterns $\mathcal{A}^{(1)}, \mathcal{A}^{(2)}$ from the 1st and 2nd transformer layer. Mathematically, they are matrix where the i-th row denotes the attention weights to the whole sequence and $\mathcal{A}_{ij}^{(*)} = \mathcal{A}_{i\to j}^{(*)}$ is formulated by:

$$\mathcal{A}^{(*)} = \sigma(\widetilde{\mathcal{A}}^{(*)}),\ \widetilde{\mathcal{A}}_{i\to j}^{(1),k} = \mathbf{w}_H^k(h_i, h_j) + \mathbf{w}_L^k(l_i, l_j),\ \widetilde{\mathcal{A}}_{h\to h'}^{(2)} = \boldsymbol{v}_{h'}^\top \boldsymbol{W}_{KQ}\boldsymbol{v}_h,$$

where we consider the index i is related to token $\boldsymbol{x}_i$ from Eq. (5) which is the h_i-th token of l_i-th example (for index j, the notation is similar), $i, j \in [T]$, $\boldsymbol{v}_{h^{(\prime)}}$ are the hidden feature $\boldsymbol{v}_{1:H}^{L+1}$ of $L+1$-th example from Layer 1 and $\mathcal{A}^{(1),k} \in \mathbb{R}^{T\times T}, \mathcal{A}^{(2)} \in \mathbb{R}^{H\times H}$. Trained attention patterns of $\mathcal{A}^{(2)}$ match the groundtruth causal structure in Fig. 2. And for the 1st layer, some heads of attention weights $\mathcal{A}^{(1),k}$ didn't learn meaningful features shown by Fig. 2 (e.g., Head 1, 9, 10). Then we dive into the parameter level, and visualize the trainable parameters of 2-layer transformer $\mathbf{w}_H^k, \mathbf{w}_L^k, \boldsymbol{W}_{KQ}, \boldsymbol{W}_{OV}$. Positional or diagonal patterns in $\mathbf{w}_H^k, \boldsymbol{W}_{KQ}$ and the similarity between $\boldsymbol{W}_{OV}$ and $\log\pi$ can be observed in Fig. 3. To fully understand why the transformer can select causal structures and what it learnt, we will need to analyze it theoretically.

Takeaway 1. Transformer formulated by Eq. (5) effectively identifies latent causal parents in-context (Fig. 2) and learns highly structural parameters aligned with the task (Fig. 3).

3.2 Constructed Transformers Implement Statistical Algorithm

Based on the patterns observed in Fig. 3, we make the following assumptions for the transformer defined by Eq. (5):

$$\tilde{\mathbf{w}}_H^k[h] = \beta\begin{cases}+1, & h=0,\\ -1, & h\in[\pm H]\setminus 0,\end{cases} \quad \exists\, k'\in[L]\ s.t.\ \tilde{\mathbf{w}}_L^k[l] = \beta\begin{cases}+1, & l=k',\\ -1, & l\in[L]\setminus k',\end{cases}$$

$$\widetilde{\boldsymbol{W}}_{KQ} = \begin{bmatrix}\boldsymbol{W} & 0_{d\times d} & \cdots & 0_{d\times d}\\ 0_{d\times d} & \boldsymbol{W} & \cdots & 0_{d\times d}\\ \vdots & \vdots & \ddots & \vdots\\ 0_{d\times d} & 0_{d\times d} & \cdots & \boldsymbol{W}\end{bmatrix}, \qquad \sigma(\widetilde{\boldsymbol{W}}_{OV}) = \pi, \tag{7}$$

where for RPE, we can simply assume one element k' of $\tilde{\mathbf{w}}_L^k$ dominates it while 0-th entry dominates $\tilde{\mathbf{w}}_H^k$: $\tilde{\mathbf{w}}_H^k[0] \gg \tilde{\mathbf{w}}_H^k[-0]$, $\tilde{\mathbf{w}}_L^k[k'] \gg \tilde{\mathbf{w}}_L^k[-k']$ and $\boldsymbol{W}$ is unknown. Since the K heads are identical up to their indices, we assume without loss of generality that the dominant entry of $\tilde{\mathbf{w}}_L^k$ occurs at position k, i.e., $\tilde{\mathbf{w}}_L^k[k] = \beta$ and we set $K = L$. The theorem below shows that aligned with the above restriction, a constructed transformer can implement statistical algorithm for inferring causal structure $\{pa(h)\}$ hidden behind $\boldsymbol{x}_{1:H}^{1:L}$ and predicting $\boldsymbol{x}_h^{L+1} \sim \pi(\cdot|\boldsymbol{x}_{pa(h)}^{L+1})$:

Theorem 1. *Under the restriction by Eq. (7), the transformer $\boldsymbol{f}_\theta$ is parameterized by $\theta \in \{(\beta, \boldsymbol{W})\}$. Then $\boldsymbol{f}_\theta$ with $\boldsymbol{W} = \log \pi$ in Eq. (7) whose second attention layer $\mathcal{A}^{(2)}(\mathcal{H}; \theta)$ approximates Bayesian Model Averaging (see Lemma 1) satisfying the following convergence property:*

$$\lim_{\beta \to \infty} \mathcal{A}^{(2)}_{h\to\cdot}(\mathcal{H}; \theta) = \lim_{\beta \to \infty} \sigma(\tilde{\mathcal{A}}^{(2)}_{h\to\cdot}(\mathcal{H}; \theta)) = \sigma(\boldsymbol{p}^{h,L}_{\texttt{BMA}}). \tag{8}$$

Further, the transformer's prediction of the distribution of last example $\boldsymbol{x}_{1:H}$ with L context examples converges to the true conditional distribution given the causal parent guaranteed by Theorem 2:

$$\lim_{\beta, L \to \infty} \boldsymbol{f}_\theta(\cdot \mid \mathcal{H}_h^L) = \pi(\cdot \mid \boldsymbol{x}_{pa(h)}), \forall h \in [H]. \tag{9}$$

Proof Sketch. Figure. 4 gives an overview of the consctruction: in the first RPE attention layer, each head from Eq. (7) is assigned to retrieve one historical copy of the same token $\boldsymbol{x}_h$, so that concatenating L heads recovers all past observations $\boldsymbol{x}_h^{1:L}$. In the second layer, with the condition in Eq. (7), the attention score between tokens (h, h') reduces to a bilinear form $\hat{\boldsymbol{p}}^h_{h'}(\boldsymbol{W}) = \sum_l \boldsymbol{x}_{h'}^{l\top} \boldsymbol{W} \boldsymbol{x}_h^l$, which by $\boldsymbol{W} = \log \pi$ coincides with the BMA score $\boldsymbol{p}^{h,L}_{\texttt{BMA}} = \sum_{l \in [L]} \log \pi(\boldsymbol{x}_h^l | \boldsymbol{x}_{h'}^l)$. With the causal mask, the softmax attention exactly matches the parent-selection distribution in BMA. By the theoretical guarantee of causal token selection (Theorem 2), OV matrix $\boldsymbol{W}_{OV}$ receives the correct parent $\boldsymbol{x}_{pa(h)}^{L+1}$ and make prediction for $\pi(\cdot \mid \boldsymbol{x}_{pa(h)}^{L+1})$. The full technical proof is deferred to Appendix C.1. □

D'Angelo et al. (2025) also consider an in-context causal learning task. With minor modification of RPE structure and above construction, we can still show transformers can exactly implement BMA.[1]

Takeaway 2. Two-layer transformers can explicitly implement BMA for causal token selection.

3.3 What Algorithm does the Transformer Learn?

Although we have constructed a transformer implementing this algorithm, what do transformers actually learn after training? Since the core lies in the attention weight $\mathcal{A}^{(2)}$ and $\boldsymbol{W}_{KQ}$ (where we use $\boldsymbol{W}_{\texttt{tf}}$ to denote the trainable submatrix in Eq. (7)) recovering graph structures, we next analyze their characteristics in detail. We first define the following parent selection metric which quantitively shows the loss of algorithms to predict parent indices:

$$\mathcal{L}_{pa}(\mathcal{A}^{(2)}(\boldsymbol{x}_{1:H}^{1:L+1}; \mathcal{G}), \mathcal{G}) = -\frac{1}{H} \sum_{h \in [H]} \boldsymbol{e}_{pa(h)}^\top \log \mathcal{A}^{(2)}_{h\cdot} = -\frac{1}{H} \sum_{h \in [H]} \log \mathcal{A}^{(2)}_{h \to pa(h)}, \tag{10}$$

where $\mathcal{A}^{(2)}$ is seen as an algorithm of predicting $e_{pa(h)}$ given input $\boldsymbol{x}_{1:H}^{1:L+1}$ and we have $\mathcal{L}_{pa}(\mathcal{A}_{\texttt{BMA}}, \mathcal{G}) = -\frac{1}{H} \sum_{h \in [H]} \boldsymbol{e}_{pa(h)}^\top \sigma\big(\hat{\boldsymbol{p}}_h(\log \pi)\big)$ by Eq. (2) where $\hat{\boldsymbol{p}}_h(\boldsymbol{W}) = \sum_l \boldsymbol{x}_{1:h-1}^{l\top} \boldsymbol{W} \boldsymbol{x}_h^l$. We visualize this metric $\mathcal{L}_{pa}$ during transformers' training process in Fig. 7 and compare it with BMA's. We observe that the transformer's parent selection loss decreases in training while remaining above the loss of BMA, gradually approaching it.

Generalized parent selection with size L' varying. We further test how well the transformer and BMA generalize in parent selection under different sample sizes L': Since $\mathcal{A}^{(2)}$ and $\mathcal{A}_{\texttt{BMA}}$ are formulated via $\hat{\boldsymbol{p}}^L = \sum_{l \in [L]} \boldsymbol{x}_{1:h-1}^{l\top} \boldsymbol{W} \boldsymbol{x}_h^l$, we vary the number of demonstrations as a set of L', and finally compute $\hat{\boldsymbol{p}}^{L'}$, $\mathcal{A}^{L'}_{h\cdot}$, and the parent selection loss $\mathcal{L}^{L'}_{\text{pa}}(\boldsymbol{W})$ with $\boldsymbol{W} \in \{\boldsymbol{W}^{(L)}_{\texttt{tf}}, \log \pi\}$.

[1] A detailed comparison with D'Angelo et al. (2025), including similarities and differences, is provided in Appendix A (Related Work).

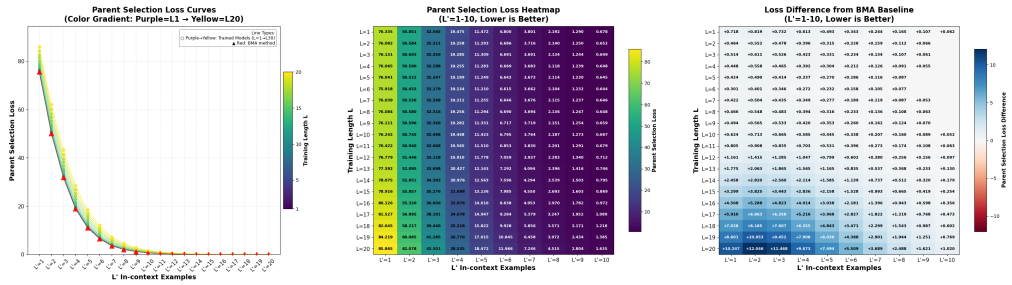


Figure 5: Generalization of Parent Selection loss $\{\mathcal{L}_{\text{pa}}^{L'}\}$ for transformers trained with $L \in \{1, \ldots, 20\}, d = 10$, and $H = 15$ with first layer fixed as constructed.

From Fig. 5, we observe that: 1) across different test sizes L', the trained transformers achieve performance close to BMA (loss differences mostly within a small margin); 2) smaller training length L often generalize better, with parent loss curves approaching BMA more closely; 3) for a model with fixed training size L, the parent loss decreases rapidly as L' increases, converging toward zero. The above results suggest the trained transformers have comparable performance to BMA.

Parameter Verification. Beyond behavioral agreement, a crucial question is whether the transformer encodes the BMA inference rule within its learned parameters. Therefore, we evaluate the similarity between the trained weight $\boldsymbol{W}_{\texttt{tf}}^{(L)}$ and the theoretical BMA parameter $\boldsymbol{W} = \log \pi$. As a first attempt, we check whether $\sigma(\boldsymbol{W}_{\texttt{tf}}) = \pi$, since for stochastic matrix π it holds that, $\sigma(\boldsymbol{W}_{\texttt{tf}}) = \pi \iff \boldsymbol{W}_{\texttt{tf}} = \log \pi + \boldsymbol{b}\mathbf{1}^\top, \forall \boldsymbol{b}$ where $\boldsymbol{b}\mathbf{1}^\top$ denotes a row-wise shift of $\log \pi$, which is canceled out by the row-softmax σ. This provides a reasonable way to normalize KQ matrix and makes the scale comparable to π with scale $[0, 1]$. However, the empirical results do not support this hypothesis (c.f. Fig. 6, first three subfigures). With some efforts, we can see the attention mechanism $\sigma(\boldsymbol{v}_{1:h-1}^\top \boldsymbol{W} \boldsymbol{v}_h)$ introduces an additional degree of freedom:

Proposition 1 (Invariances of attention scores)**.** *Since attention operates on a single query $\boldsymbol{v}_h$, if the columns of $\boldsymbol{W}_{\texttt{tf}}$ differ from those of $\log \pi$ by an additive factor, i.e. $\boldsymbol{W}_{\texttt{tf}} = \log \pi + \mathbf{1}\boldsymbol{a}^\top, \boldsymbol{a} \in \mathbb{R}^d$, then transformer with $\boldsymbol{W}_{\texttt{tf}}$ learnt BMA by: $\sigma(\sum_l \boldsymbol{x}_{1:h-1}^{l\top} \boldsymbol{W}_{\texttt{tf}} \boldsymbol{x}_h^l) = \sigma(\sum_l \boldsymbol{x}_{1:h-1}^{l\top} \log \pi \boldsymbol{x}_h)$. Further, if Markov chain $\boldsymbol{x}_{1:H}$ is stationary and $\boldsymbol{W}_{\texttt{tf}} = \log \pi + \mathbf{1}\boldsymbol{a}^\top + \boldsymbol{b}\mathbf{1}^\top, \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$, the above conclusion also holds asymptotically. See the detailed proof in Appendix C.3.*

Following this proposition, we evaluate the discrepancy between $\sigma(\boldsymbol{W}_{\texttt{tf}}\mathbf{1}\boldsymbol{a}^\top)$ and π. As illustrated in Fig. 6, the deviation remains small, with

$$\frac{1}{d}||\sigma(\boldsymbol{W}_{\texttt{tf}} - \mathbf{1}\boldsymbol{a}^\top) - \pi||_F < 0.05. \tag{11}$$

This also holds across various vocabulary size $d \in \{10, 30, 50\}$, further confirming the structural alignment between the learned model and the BMA algorithm. Taken together with our theoretical construction and empirical results, these findings strongly suggest that transformers implement the BMA method for in-context causal parent selection.

Takeaway 3. Transformers with trainable $\boldsymbol{W}_{\texttt{tf}}$ closely approximate BMA in causal token selection (Fig. 5) and learn parameters which explicitly show strong alignment with BMA (Fig. 6).

3.4 Theoretical Understanding and Guarantee of Learned Algorithm

Beyond identifying what algorithm a trainable transformer adopts, we further establish a theoretical understanding of transformers' in-context causal structure selection mechanism via information-theoretic principles. Our approach follows Nichani et al. (2024), which leverages *mutual information* together with the data's inherent property of the *Data Processing Inequality* (DPI). In contrast to their gradient-based proof, we show that transformers can exploit this property **directly in context**. Moreover, our analysis generalizes the χ^2–mutual information framework of Nichani et al. (2024); D'Angelo et al. (2025) to the setting reducible to classical mutual information, by exploiting the

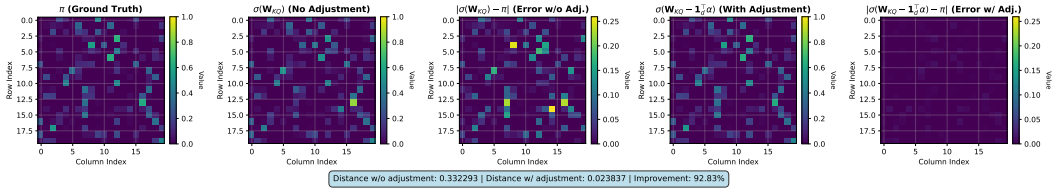


Figure 6: Parameter-Level Comparison between transformer and BMA ($\boldsymbol{W}_{KQ} = \log \pi$). Here, $\boldsymbol{W}_{KQ}$ denotes the diagonal block being trained. Trained with $d = 20$, $H = 50$, $L = 3$, and 2048 training steps while $\boldsymbol{W}_{OV}$ is fixed. (See results with both $\boldsymbol{W}_{OV}$ and $\boldsymbol{W}_{KQ}$ trainable in Fig. 8.)

information-theoretic structure characterized in Lemma 3. Finally, our proof of Theorem 2 doesn't rely on the stationarity assumption on the data distribution not requiring the chain to be mixed.[2]

Definition 1 (Mutual Information and Conditional Entropy)**.** *Consider* x, y *are two random variables in disrcrete or continuous space* Ω*,* $P_{\mathrm{x},\mathrm{y}}$ *and* $P_{\mathrm{x}}, P_{\mathrm{y}}$ *denote joint and marginal distribution. The mutual infomration* $I(\mathrm{x};\mathrm{y})$*, entropy* $H(\mathrm{x})$ *and the conditional entropy* $H(\mathrm{x}|\mathrm{y})$ *is given by:*

$$I(\mathrm{x};\mathrm{y}) = \int_x \int_y P_{\mathrm{x},\mathrm{y}}(x,y) \log \frac{P_{\mathrm{x},\mathrm{y}}(x,y)}{P_{\mathrm{x}}(x)P_{\mathrm{y}}(y)}, \quad H(\mathrm{x}) = -\int_x P_{\mathrm{x}}(x) \log P_{\mathrm{x}}(x),$$
$$H(\mathrm{x}|\mathrm{y}) = \int_x \int_y P_{\mathrm{x},\mathrm{y}}(x,y) \log \frac{P_{\mathrm{x},\mathrm{y}}(x,y)}{P_{\mathrm{y}}(y)} = H(\mathrm{x}) - I(\mathrm{x};\mathrm{y}), \tag{12}$$

Further, χ^2*-mutual information is given by:* $I_{\chi^2}(\mathrm{x};\mathrm{y}) \coloneqq \int_x \int_y \frac{P_{\mathrm{x},\mathrm{y}}(x,y)^2}{P_{\mathrm{x}}(x)P_{\mathrm{y}}(y)} - 1.$

I, I_{χ^2} can be uniformly derived from f-divergence which helps to prove DPI for generalized f-mutual information I_f. These information metrics reveal an essential property in data:

Lemma 2 (DPI. Theorem 3.9 and 7.16 in Polyanskiy & Wu (2023))**.** *If random variables* $\mathrm{x} \to \mathrm{y} \to \mathrm{z}$*, i.e. statisfies the Markov property* $p(x,y,z) = p(x)p(y|x)p(z|y)$*, then we have* $I_f(\mathrm{y};\mathrm{z}) \geq I_f(\mathrm{x};\mathrm{z})$*. Further, for classic mutual information,* $I(\mathrm{x};\mathrm{z}) = I(\mathrm{y};\mathrm{z})$ *iff* $I(\mathrm{x};\mathrm{y}|\mathrm{z}) = 0$ *iff* $\mathrm{x} \to \mathrm{z} \to \mathrm{y}$.

Suppose we have a Markov chain $\mathrm{x}_{1:H}$ with latent causal structure $\{pa(h)\}$. Since $\forall h' \neq pa(h)$, we have $P(\mathrm{x}_h = z|\mathrm{x}_{pa(h)} = y, \mathrm{x}_{h'} = x) = P(z|y)$, it is easy to verify $\mathrm{x}_{h'} \to \mathrm{x}_{pa(h)} \to \mathrm{x}_h$ while $\mathrm{x}_{h'} \to \mathrm{x}_h \to \mathrm{x}_{pa(h)}$ doesn't hold. Thus, we have the following corollary:

Corollary 1. *For Markov chain with causal structure* $\mathcal{G}$*,* $I(\mathrm{x}_h;\mathrm{x}_{pa(h)}) > I(\mathrm{x}_h;\mathrm{x}_{h'}), \forall h' \neq pa(h)$.

Applying this corollary, we get the following Lemma:

Lemma 3. *For Markov chain with causal structure* $\mathcal{G}$ *and transition kernel* $p(\cdot|\cdot)$*, we have*

$$\mathbb{E}[\log p(\mathrm{x}_h|\mathrm{x}_{pa(h)})] > \mathbb{E}[\log p(\mathrm{x}_h|\mathrm{x}_{h'})], \quad \forall h' \neq pa(h). \tag{13}$$

The LHS above equals $H(\mathrm{x}_h|\mathrm{x}_{pa(h)})$, while the RHS differs from conditional entropy but follows $H(\mathrm{x}_h|\mathrm{x}_{h'})$ from the non-negativity of the KL divergence. Then by the relation of conditional entropy and mutual information, we can apply DPI to prove the Lemma. See Appendix C.4. With Lemma 3, we can build the relation between $\mathbb{E}[\log p(\mathrm{x}_h|\mathrm{x}_{h'})]$ and attention weights, showing the theoretical guarantee concerning parent selection for the transformer:

Theorem 2. *Suppose the transformer is constructed as in Theorem 1, which implements the BMA method. Then the attention weights* $\tilde{\mathcal{A}}^L_{h\cdot} = \tilde{\mathcal{A}}^{(2)}_{h\cdot}(\boldsymbol{x}^{1:L+1}_{1:H})$ *predicting parent index* $pa(h)$ *will statisty:*

$$\lim_{L\to\infty} \tilde{\mathcal{A}}^L_{h\cdot} = \lim_{L\to\infty} \sigma(\hat{\boldsymbol{p}}^{h,L}) = \boldsymbol{e}_{pa(h)} \in \mathbb{R}^H, \quad \textit{where } \hat{\boldsymbol{p}}^{h,L}_{h'} = \sum_{l=1}^{L} \log \pi(\boldsymbol{x}^l_h|\boldsymbol{x}^l_{h'}).$$

The proof of the theorem is deferred to Appendix C.5.

Takeaway 4. Information-theoretic analysis reveals that the selection mechanism exploits the conditional entropy, where the DPI guarantees the identifiability of the true causal parent.

[2] This ensures that when the chain has not mixed, i.e., when the time index h is small, the parent-selection guarantee still holds.

3.5 Causal Structure in Training Dynamics

We further look at the training dynamics of the transformer model. We prove the random causal structure embedded in inputs will be recovered in the gradients of loss w.r.t. the core $\boldsymbol{W}_{KQ}$ matrix:

Theorem 3 (Informal). *Consider the transformer $\boldsymbol{f}_\theta$ constructed as in Theorem 1 with trainable diagonal block $\boldsymbol{W}$ of $\boldsymbol{W}_{KQ}$ specified in Eq. (7) and trained with cross-entropy loss*

$$\mathcal{L}(\theta) = -\sum_{h=1}^{H} \mathbb{E}_{(\boldsymbol{x}_{1:H}^{1:L+1},\mathcal{G})\sim P_\pi}[\log(\boldsymbol{f}_{\theta_0}(\boldsymbol{x}_h^{L+1}|\mathcal{H}) + \epsilon)] = -\sum_{h=1}^{H} \mathbb{E}_{\mathcal{G}\sim P_\mathcal{G}}[\ell(\theta; h, \mathcal{G})], \tag{14}$$

with joint distribution P_π of latent graph and input, $\ell(\theta; h, \mathcal{G}) = \mathbb{E}_{\boldsymbol{x}_{1:H}^{1:L+1}\sim P_{\pi|\mathcal{G}}}[\log \boldsymbol{f}_{\theta_0}(\boldsymbol{x}_h^{L+1}|\mathcal{H})]$ and $\hat{\boldsymbol{p}} = \sum_l \boldsymbol{x}_{1:h-1}^{l\top} \boldsymbol{W} \boldsymbol{x}_h^l \in \mathbb{R}^{h-1}$. If the Markov chain is stationary, i.e., $\boldsymbol{x}_h \sim \mu^\pi, \forall h \in [H]$, at initialization of f_{θ_0} with $\boldsymbol{W} = \boldsymbol{0}$ assuming f_θ outputs μ^π for any input, then the gradient statisfies

$$\frac{\partial \ell(\theta_0; h, \mathcal{G})}{\partial \hat{\boldsymbol{p}}}_{pa(h)} \geq \frac{\partial \ell(\theta_0; h, \mathcal{G})}{\partial \hat{\boldsymbol{p}}}_{h'}, \quad \forall h' \neq pa(h).$$

The theorem above is proved by leveraging the χ^2-mutual information, as detailed in Appendix C.6. This result provides an explanation of how transformers can extract meaningful information from data. To further support the theory, we verify it empirically by visualizing $\frac{\partial \mathcal{L}(\boldsymbol{W})}{\partial \hat{\boldsymbol{p}}}$ in Fig. 9.

Takeaway 5. Gradient at initialization is able to recover the latent causal structure, driven by χ^2-mutual information, which facilitates structural discovery in early training (Thm. 3 and Fig. 9).

4 Dynamical System Extension: From Discrete to Continuous

Further, we consider to investigate the Markov chain in continuous space, where we look at the linear dynamical system with latent causal structures: $\mathrm{x}_h = \frac{1}{c}(\boldsymbol{A}^\top \mathrm{x}_{pa(h)} + \varepsilon_h) \in \mathbb{R}^d, \varepsilon_h \in \mathcal{N}(0, \sigma^2 I_d)$. We first train a transformer with RPE introduced in Eq. (5) on data generated from the dynamical system. Similar experimental results on attention weights $\mathcal{A}^{(1)}, \mathcal{A}^{(2)}$ and parameter visualizations can be found in Appendix Fig. 10, 11, 12, and 13. These RPE parameters are consistent with the construction in Eq. (7). Moreover, the attention weights $\mathcal{A}^{(2)}$ of the transformer yields accurate predictions of parent indices across many examples. Similar to the discrete case, we can define the transition $p(\cdot|\cdot)$ by $\mathrm{x}_h|\mathrm{x}_{pa(h)} \sim \mathcal{N}(\frac{1}{c}\boldsymbol{A}^\top \mathrm{x}_{pa(h)}, \frac{\sigma^2}{c^2} I_d)$. Consequently, Eq. (2) specifies the BMA formulation under the dynamical system setting. In this context, Lemma 3 remains valid and guarantees the asymptotic correctness of BMA's parent selection. To investigate transformers' mechanism of parent selection, we test the parent selection loss $\mathcal{L}_{pa}^{L'}$ of the transformer and BMA in dynamical setting, where we set various L' in-context samples as introduced in Sec. 3.3. Fig. 15 demonstrates that the transformer with trainable $(\boldsymbol{W}, \boldsymbol{W}_{OV})$ achieves performance comparable to BMA method when L' approaches 20. However, the loss $\mathcal{L}_{pa}^{L'}$ between transformers and BMA remains a noticeable gap. We conjecture that the proposition below may explain this discrepancy:

Proposition 2 (Representation Limitation of Transformers). *Under the observation restriction in Eq. (7), both the transformer and BMA take the unified form $\mathcal{A}_{h\to h'} = \sigma(\boldsymbol{p}^h)_{h'}$. In the DS setting, transformer logits are bilinear, $\boldsymbol{p}_{\mathtt{tf},h'}^h = \sum_l \boldsymbol{x}_{h'}^{l\top} \boldsymbol{W}_{\mathtt{tf}} \boldsymbol{x}_h^l$, whereas BMA logits are $\boldsymbol{p}_{\mathtt{BMA},h'}^h = c_1 \sum_l \boldsymbol{x}_{h'}^{l\top} \boldsymbol{A} \boldsymbol{x}_h^l + d \sum_l \|\boldsymbol{x}_{h'}^l\|^2$ with $d \neq 0$. There exists no $\boldsymbol{W}_{\mathtt{tf}}$ such that $\sigma(\boldsymbol{p}_{\mathtt{tf}}^h) = \sigma(\boldsymbol{p}_{\mathtt{BMA}}^h)$ holds for all DS samples $(\boldsymbol{x}_{1:H}^{1:L+1}, \mathcal{G}) \sim P_\pi$. Hence transformers under Eq. (7) cannot represent BMA in the DS setting. However, for MC setting, $\boldsymbol{W}_{\mathtt{tf}} = \log \pi$ gives the BMA form.*

For BMA, $(\boldsymbol{p}_{\mathtt{BMA}}^h)_{h'} = \sum_l \log p(\boldsymbol{x}_h^l | \boldsymbol{x}_{h'}^l)$. In dynamical systems, transition $p(\cdot|\cdot)$ involves not only cross but also quadratic terms. If the representation equation holds, substituting $\boldsymbol{x}_h = \frac{1}{c}\boldsymbol{A}^\top \boldsymbol{x}_{pa(h)} + \varepsilon_h$ and using the independence of ε_h forces the coefficient of ε_h to vanish, which implies $\boldsymbol{W} = c_1 \boldsymbol{A}$, contradicting the original representation. So no matrix $\boldsymbol{W}$ can yield $\mathcal{A}_{h\cdot}^{(2)} = \sigma(\hat{\boldsymbol{p}}^h(\boldsymbol{W}))$ as in Eq. (18) to represent BMA. See Appendix C.7 for detailed proof.

REFERENCES

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning, 2023. URL https://arxiv.org/abs/2306.00297. 13

Kabir Ahuja, Madhur Panwar, and Navin Goyal. In-context learning through the bayesian prism. *arXiv preprint arXiv:2306.04891*, 2023. 13

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models, 2023. URL https://arxiv.org/abs/2211.15661. 1

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of Language Models: Part 1, Learning Hierarchical Language Structures. *SSRN Electronic Journal*, May 2023. Full version available at https://ssrn.com/abstract=5250639. 1, 14

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection, 2023. URL https://arxiv.org/abs/2306.04637. 1

Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint, 2023. URL https://arxiv.org/abs/2306.00802. 1

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165. 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. 1

Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Unveiling induction heads: Provable training dynamics and feature learning in transformers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=4fN2REs0Ma. 2, 13

Francesco D'Angelo, Francesco Croce, and Nicolas Flammarion. Selective induction heads: How transformers select causal structures in context. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bnJgzAQjWf. 2, 7, 8, 13

Ezra Edelman, Nikolaos Tsilivis, Benjamin L. Edelman, Eran Malach, and Surbhi Goel. The evolution of statistical induction heads: In-context learning markov chains. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 64273–64311. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/75b0edb869e2cd509d64d0e8ff446bc1-Paper-Conference.pdf. 1, 2, 13

Dan Friedman, Alexander Wettig, and Danqi Chen. Learning transformer programs, 2023. URL https://arxiv.org/abs/2306.01128. 4

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes, 2023. URL https://arxiv.org/abs/2208.01066. 1

Gautam Goel and Peter Bartlett. Can a transformer represent a kalman filter?, 2024. URL https://arxiv.org/abs/2312.06937. 1

Tianyu Guo, Wei Hu, Song Mei, Huan Wang, Caiming Xiong, Silvio Savarese, and Yu Bai. How do transformers learn in-context beyond simple functions? a case study on learning with representations. *arXiv preprint arXiv:2310.10616*, 2023. 13

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980. 20

Ashok Vardhan Makkuva, Marco Bondaschi, Adway Girish, Alliot Nagle, Martin Jaggi, Hyeji Kim, and Michael Gastpar. Attention with markov: A framework for principled analysis of transformers via markov chains, 2025. URL https://arxiv.org/abs/2402.04161. 13

Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent, 2024. URL https://arxiv.org/abs/2402.14735. 1, 2, 8, 13, 14, 17, 18, 26, 33

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022. URL https://arxiv.org/abs/2209.11895. 6, 14

Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2023. ISBN 9781108832908. doi: 10.1017/9781108966351. 9

Nived Rajaraman, Marco Bondaschi, Kannan Ramchandran, Michael Gastpar, and Ashok Vardhan Makkuva. Transformers on markov data: Constant depth suffices, 2024. URL https://arxiv.org/abs/2407.17686. 1

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations, 2018. URL https://arxiv.org/abs/1803.02155. 5

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864. 5

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent, 2023. URL https://arxiv.org/abs/2212.07677. 1, 13

Jiuqi Wang, Ethan Blaser, Hadi Daneshmand, and Shangtong Zhang. Transformers can learn temporal difference methods for in-context reinforcement learning, 2025. URL https://arxiv.org/abs/2405.13861. 1

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR. 1

Kevin Christian Wibisono and Yixin Wang. From unstructured data to in-context learning: Exploring what tasks can be learned and when, 2024. URL https://arxiv.org/abs/2406.00131. 1, 14

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021. 13, 14

Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. *arXiv preprint arXiv:2305.19420*, 2023. 13

Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. Do transformers parse while predicting the masked word?, 2023. URL https://arxiv.org/abs/2303.08117. 1

A Notation and Related Work

Notation. We use $[h]$ to denote the set $\{1, 2, \ldots, h\}$. For causal structure, we use $pa(h)$ to represent the parent index of node h. The stationary distribution of Markov chain $\boldsymbol{x}_h \sim \pi(\cdot|\boldsymbol{x}_{pa(h)})$ is denoted by $\mu^\pi \in \Delta^d$. For transformer model, the input of a sequence of vectors is given by $\boldsymbol{x}_{1:T} := [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T] \in \mathbb{R}^{d\times T}$. Given the input, we denote the attention scores of standard self-attention layer as $\boldsymbol{p}^t := \boldsymbol{x}_{1:T}^\top \boldsymbol{W}_K^\top \boldsymbol{W}_Q \boldsymbol{x}_t \in \mathbb{R}^T$. However the causal mask $\mathcal{M}$ in attention layer will lead to $\hat{\boldsymbol{p}}^t := \boldsymbol{x}_{1:t-1}^\top \boldsymbol{W}_K^\top \boldsymbol{W}_Q \boldsymbol{x}_t \in \mathbb{R}^{t-1}, \sigma(\hat{\boldsymbol{p}}^t)_{t'} = \sigma(\mathcal{M}(\boldsymbol{p}^t))_{t'}, \forall t' \in [t-1]$. We do not distinguish between them in the proofs. For the matrix form of the attention of an input sequence, we use $\tilde{\mathcal{A}}$ and $\mathcal{A}$ to denote attention weights and scores correspondingly, where we have $\boldsymbol{p}_{t'}^t = \mathcal{A}_{t\to t'}$ and $\sigma(\tilde{\mathcal{A}}) = \mathcal{A}$. In training, we use cross-entropy loss and MSE loss for Markov chain and dynamical system settings respectively:

$$\begin{aligned}\mathcal{L}^{MC}(\theta) &= -\frac{1}{H}\sum_{h=2}^{H} {\boldsymbol{x}_h^{L+1}}^\top \log\left(\sigma(\boldsymbol{f}_{\texttt{tf}}(\cdot \mid \mathcal{H}_h)) + \epsilon\right), \\ \mathcal{L}^{DS}(\theta) &= -\frac{1}{H}\sum_{h=2}^{H} ||\boldsymbol{x}_h^{L+1} - \boldsymbol{f}_{\texttt{tf}}(\cdot \mid \mathcal{H}_h)||_2^2,\end{aligned} \tag{15}$$

where θ represents all trainable parameters and ϵ is a small value to avoid numerical issues by $\log$.

Related Work. A growing body of work studies the in-context learning (ICL) ability from different perspectives. One line of work understands ICL as a form of Bayesian inference, showing how the latent concept can be approximately inferred under restrictive theoretical assumptions (Xie et al., 2021; Zhang et al., 2023; Ahuja et al., 2023). Another direction of research investigates how transformers can simulate standard algorithms, such as gradient descent on linear regression (von Oswald et al., 2023; Ahn et al., 2023; Guo et al., 2023). While these works demonstrate the ICL power of transformers, they commonly assume i.i.d or uncorrelated input tokens. To move beyond i.i.d. assumptions, recent works investigate ICL with *correlated data*, particularly Markovian sequences (Edelman et al., 2024; Chen et al., 2024b; Makkuva et al., 2025). These settings provide insight into how transformers handle in-context learning with sequential dependencies, but typically focus on fixed dependency structures. In contrast, our work addresses *variable causal structures* that differ across prompts. Pioneering this direction, Nichani et al. (2024) demonstrated that transformers can encode fixed parent–child dependencies (e.g., bigrams) in Markov chains. Building on this, D'Angelo et al. (2025) introduced *selective induction heads*, enabling transformers to identify the underlying Markovian order (or "lag") from a candidate set in-context learning this structure. Our work generalizes this setting. While D'Angelo et al. (2025) focus on inferring a single global structural parameter (the lag k) shared across the sequence, we tackle local structure inference where dependencies can vary arbitrarily for each position, effectively modeling latent trees rather than fixed-lag chains. Theoretically, D'Angelo et al. (2025) constructs a three-layer transformer that asymptotically implements maximum likelihood estimation, where its construction is verified via attention pattern visualization as well as quantitive validation through KL divergence of next-token prediction targets. In our work, we theoretically derive a two-layer architecture that explicitly implements Bayesian Model Averaging (BMA) in-context. Empirically, we go beyond behavioral metrics by providing *parameter-level* verification, demonstrating that the trained weights directly encode the transition kernel. Furthermore, we provide theoretical understanding of in-context causal structure learning based on the Data Processing Inequality (DPI) and extend our analysis to continuous dynamical systems, revealing representational gaps not occuring in the discrete setting.

B Conclusion

In this work, we investigated the capability of transformers to infer and adapt to latent causal structures in-context, moving beyond the fixed dependency assumptions common in prior theoretical analysis. We proposed a novel framework based on Markov chains with randomly sampled causal dependencies, requiring the model to identify position-specific predecessor-successor relationships from context examples. First, we provided a constructive proof that a two-layer transformer with relative position embeddings (RPE) can explicitly implement Bayesian Model Averaging (BMA). This demonstrates that the attention mechanism is theoretically capable of performing statistical

inference over structural uncertainty. Second, through extensive experiments and parameter-level analysis, we showed that trained transformers implements BMA method which converge to this theoretical construction: the learned attention patterns directly encode the posterior probabilities of causal parents, and the weights explicitly recover the log-transition kernel of the underlying generative process. Third, we established information-theoretic guarantees using the Data Processing Inequality (DPI) which help understand how the selection mechanism identifies causal structures in context and showed that gradients at initialization recover these dependencies via χ^2-mutual information. Finally, we extended our framework to continuous linear dynamical systems. While transformers continue to exhibit strong empirical performance in this setting, we identified the representational difference that prevents the exact implementation of BMA, unlike in the discrete case. Collectively, our findings offer a mechanistic explanation of how transformers perform in-context causal learning, highlighting their ability to act as statistical inference engines for both discrete and continuous data.

Broader Implications. Our findings support theoretical frameworks that model in-context learning as a statistical inference task (Xie et al., 2021). Distinct from "Induction Heads" which typically focus on copying fixed positional dependencies (Olsson et al., 2022; Nichani et al., 2024), we demonstrate a probabilistic setting where the model must infer a latent dependency structure that varies per example. This provides a mechanistic grounding for how LLMs adapt to flexible, context-dependent rules rather than relying solely on fixed n-gram statistics (Allen-Zhu & Li, 2023). Furthermore, this helps understand why LLMs demonstrate ICL capabilities on empirical task with "unstructured" language data (Wibisono & Wang, 2024), mirroring our setting where the transition mappings between words are fixed while the structural positions of a couple of words vary from input to input.

Limitations and Future Work. We acknowledge that real-world sequences often involve complex non-linear dynamics or hierarchical dependencies (e.g., context-free grammar) beyond the Markovian and dynamical systems studied here. However, our primary objective in this work was to prioritize mechanistic interpretability for Markov chain or dynamical system: explicitly characterizing how transformers infer latent structures in-context on these tasks. By focusing on these tractable settings, we were able to derive exact theoretical guarantees and provide parameter-level verification that the model implements Bayesian Model Averaging. We believe this explainable framework serves as a necessary foundation, and we leave the extension to more complex non-linear and hierarchical data generating processes for future exploration.

C Proofs of Technical Lemmas

C.1 Proof of Theorem 1

Proof. By the condition of $(\tilde{\mathbf{w}}_H, \tilde{\mathbf{w}}_L)$ in Eq. (7), the attention score $\hat{\mathcal{A}}^{(1)}_{t\to\cdot}$ of query $\boldsymbol{x}_t = \boldsymbol{x}_h^{L+1}$ is:

$$\hat{\mathcal{A}}^{(1),k}_{t\to t'} = \tilde{\mathbf{w}}_H^k[h_t - h_{t'}] + \tilde{\mathbf{w}}_L^k[l_t - l_{t'}] = 2\beta \begin{cases} +1, & \text{if } h_t = h_{t'},\ l_t - l_{t'} = k, \\ -1, & \text{if } h_t \neq h_{t'},\ l_t - l_{t'} \neq k, \\ 0, & \text{otherwise.} \end{cases}$$

Then the output $\tilde{\boldsymbol{u}}_h^k = \mathrm{Attn}^k_{\boldsymbol{x}_t \to \boldsymbol{x}_{1:T}}$ of the 1st attention layer will be calculated as:

$$\begin{aligned} \tilde{\boldsymbol{u}}_h^k &= \sigma(\tilde{\mathbf{w}}_H^k(h,\cdot) + \tilde{\mathbf{w}}_L^k(L+1,\cdot))\boldsymbol{x}_{1:T}^\top \\ &\xrightarrow{\beta\to\infty} \boldsymbol{x}_h^{l_k\top} = (1_{[h_{t'}=h, l_{t'}=L+1-k]})_{t'\in[T]}\boldsymbol{x}_{1:T}^\top. \quad (l_k = L+1-k) \end{aligned} \tag{16}$$

By disentangled residual, the output of K heads ($K = L$) will be concatenated as:

$$\boldsymbol{v}_h = [\boldsymbol{u}_h^1, \ldots, \boldsymbol{u}_h^L], \text{ with } \boldsymbol{u}_h^k = \boldsymbol{x}_h^{L+1-k} \text{ by Eq. (16).}$$

For the 2nd layer, with diagonal condition of $\boldsymbol{W}_{KQ}$, attention weight $\mathcal{A}^{(2)} \in \mathbb{R}^{H\times H}$ is given by:

$$\hat{\mathcal{A}}^{(2)}_{h\to h'} = \boldsymbol{v}_{h'}^\top \widetilde{\boldsymbol{W}}_{KQ}\boldsymbol{v}_h = \sum_{l=1}^{L} \boldsymbol{x}_{h'}^{l\top}\boldsymbol{W}\boldsymbol{x}_h^l, \quad \mathcal{A}^{(2)} = \sigma(\mathcal{M}(\hat{\mathcal{A}}^{(2)})) \in \mathbb{R}^{H\times H}, \tag{17}$$

where $\mathcal{M}$ is the causal mask enforcing $\mathcal{A}^{(2)}$ to be strictly lower-triangular after softmax. If we define the vector $\hat{\boldsymbol{p}}^h \in \mathbb{R}^{h-1}$ with $\hat{\boldsymbol{p}}^h_{h'} := \hat{\mathcal{A}}^{(2)}_{h\to h'}$, we have $\forall h' \in [h-1]$:

$$\mathcal{A}^{(2)}_{h\to h'} = \sigma(\mathcal{M}_h(\hat{\mathcal{A}}_{h\to\cdot}))_{h'} = \sigma(\hat{\boldsymbol{p}}^h)_{h'}, \quad \hat{\boldsymbol{p}}^h(\boldsymbol{W}) = \sum_l \boldsymbol{x}_{1:h-1}^{l\top}\boldsymbol{W}\boldsymbol{x}_h^l, \tag{18}$$

where $\mathcal{M}_h(\cdot)$, is the causal mask applied to row h setting $\mathcal{M}_h(\boldsymbol{v})_{h'} = -\infty$ if $h' \geq h, \forall \boldsymbol{v} \in \mathbb{R}^H$. Then we set $\boldsymbol{W}$ as $\log \pi$ (elementwise) which leads to

$$\hat{\boldsymbol{p}}^h_{h'}(\log \pi) = \hat{\mathcal{A}}^{(2)}_{h \to h'} = \sum_l \log \pi(\boldsymbol{x}^l_h | \boldsymbol{x}^l_{h'}), \quad \mathcal{A}^{(2)}_{h\cdot} = \sigma(\mathcal{M}_h(\hat{\mathcal{A}}_{h\to\cdot})) \in \mathbb{R}^H. \tag{19}$$

With the form of $\mathcal{A}^{(2)}$ in Eq. (19) and Lemma 1 to be proved, we can show the BMA method of Eq. (1) has the same formulation:

$$\mathbb{P}(pa(h) = h' | \boldsymbol{x}^{1:L}_{1:H}) = \sigma(\hat{\boldsymbol{p}}^h(\boldsymbol{W} = \log \pi))_{h'}. \tag{20}$$

Combining Eq. (20) with the limit behavior of the first layer in Eq. (16), we obtain the first convergence result for parent selection as $\beta \to \infty$:

$$\lim_{\beta \to \infty} \mathcal{A}^{(2)}_{h\to\cdot}(\mathcal{H}; \theta) = \sigma(\hat{\boldsymbol{p}}^h(\boldsymbol{W} = \log \pi)) = \sigma(\boldsymbol{p}^{h,L}_{\texttt{BMA}}).$$

Furthermore, guaranteed by the consistency of BMA (Theorem 2), as sample size $L \to \infty$, the posterior concentrates on the true parent $pa(h)$. Thus, the prediction of the token distribution converges in the limit $\beta, L \to \infty$ as:

$$\lim_{\beta, L \to \infty} \boldsymbol{f}_\theta(\cdot \mid \mathcal{H}) = \sigma\left(\boldsymbol{W}^\top_{OV} \sum_{h'} 1_{[h'=pa(h)]} \boldsymbol{x}^{L+1}_{h'}\right) = \pi(\cdot \mid \boldsymbol{x}^{L+1}_{pa(h)}). \qquad \square$$

In proving the theorem, we rely on the Lemma 1 proved below, which illustrates the relation between BMA and attention weights.

C.2 Proof of Lemma 1

Proof. Here we use $p(s|s')$ to denote $\mathbb{P}(\boldsymbol{x}_h = s | \boldsymbol{x}_{pa(h)} = s')$ for generality beyond discrete Markov chain. Based on Bayesian Theorem, it can be calculated by Eq. (1). Due to the Markovian property $p(\boldsymbol{x}_h | \boldsymbol{x}_{1:h-1}) = p(\boldsymbol{x}_h | \boldsymbol{x}_{pa(h)})$, the joint distribution of this chain $\mathrm{x}_{1:H}$:

$$\begin{aligned} p(\boldsymbol{x}_{1:H}) &= p(\boldsymbol{x}_1) \prod_{h=2}^H p(\boldsymbol{x}_h | \boldsymbol{x}_{1:h-1}) = p(\boldsymbol{x}_1) \prod_{h=2}^H p(\boldsymbol{x}_h | \boldsymbol{x}_{pa(h)}) \\ &= p(\boldsymbol{x}_1) \prod_{i \neq h} p(\boldsymbol{x}_i | \boldsymbol{x}_{pa(i)}) \cdot p(\boldsymbol{x}_h | \boldsymbol{x}_{pa(h)}) \end{aligned} \tag{21}$$

Here $pa(h), pa(i)$ in Eq. (21) are random index with prior: $pa(h) \sim \text{Uniform}([h-1])$. With condition $pa(h) = h$ in Eq. (1), we can substitue $pa(h)$ in Eq. (21) with h'. Since $\{pa(h'')\}$ are random index out of interests, these terms are eliminated:

$$\frac{\mathbb{P}\left(\boldsymbol{x}^{1:L}_{1:H} \,\middle|\, \mathrm{pa}(h) = h'\right)}{\sum_{h'' \in [h-1]} \mathbb{P}\left(\boldsymbol{x}^{1:L}_{1:H} \,\middle|\, \mathrm{pa}(h) = h''\right)} = \frac{\prod_l \left(p(\boldsymbol{x}^l_1) \prod_{i \neq h} p(\boldsymbol{x}^l_i | \boldsymbol{x}^l_{pa(i)}) \cdot p(\boldsymbol{x}^l_h | \boldsymbol{x}^l_{h'})\right)}{\sum_{h''} \prod_l \left(p(\boldsymbol{x}^l_1) \prod_{i \neq h} p(\boldsymbol{x}^l_i | \boldsymbol{x}^l_{pa(i)}) \cdot p(\boldsymbol{x}^l_h | \boldsymbol{x}^l_{h''})\right)}, \tag{22}$$

which leads to:

$$\mathbb{P}(pa(h) = h' \mid \boldsymbol{x}^{1:L}_{1:H}) = \frac{\exp\left(\sum_{l \in [L]} \log p(\boldsymbol{x}^l_h | \boldsymbol{x}^l_{h'})\right)}{\sum_{h'' \in [h-1]} \exp\left(\sum_{l \in [L]} \log p(\boldsymbol{x}^l_h | \boldsymbol{x}^l_{h''})\right)} = \sigma\left(\hat{\boldsymbol{p}}^h(\log W^P)\right)_{h'}, \tag{23}$$

where the matrix $W^P = \pi$ is induced by transition kernel $P = \pi$ in Markov chain. $\square$

C.3 Proof of Proposition 1

First, suppose we have $\boldsymbol{W}_{\texttt{tf}} = \log \pi + \mathbf{1}\boldsymbol{a}^\top$. The attention weights of the transformer are given by:

$$\boldsymbol{p}^h_{\texttt{tf}} = \sum_l \boldsymbol{x}^{l\top}_{1:h-1} \boldsymbol{W}_{\texttt{tf}} \boldsymbol{x}^l_h = \sum_l \boldsymbol{x}^{l\top}_{1:h-1} \log \pi \boldsymbol{x}^l_h + \sum_l \boldsymbol{x}^{l\top}_{1:h-1} \mathbf{1}\boldsymbol{a}^\top \boldsymbol{x}^l_h.$$

For the second term, since $\{\boldsymbol{x}_{h'}\}$ are one-hot, we have

$$\sum_l \boldsymbol{x}^{l\top}_{1:h-1} \mathbf{1}\boldsymbol{a}^\top \boldsymbol{x}^l_h = \mathbf{1}_{h-1} \boldsymbol{a}^\top \left(\sum_l \boldsymbol{x}^l_h\right) = c(\boldsymbol{a}, h) \mathbf{1}_{h-1},$$

where $c(\boldsymbol{a}, h) = \boldsymbol{a}^\top \left(\sum_l \boldsymbol{x}_h^l \right)$ is a constant with fixed index h. Then by softmax operation, we have:

$$\begin{aligned}\sigma(\boldsymbol{p}_{\texttt{tf}}^h) &= \sigma\Big(\sum_l \boldsymbol{x}_{1:h-1}^{l\top} \log \pi \boldsymbol{x}_h^l + c(\boldsymbol{a}, h) \mathbf{1}_{h-1} \Big) \\ &= \sigma\Big(\sum_l \boldsymbol{x}_{1:h-1}^{l\top} \log \pi \boldsymbol{x}_h^l \Big) \\ &= \sigma(\boldsymbol{p}_{\texttt{BMA}}^h (\log \pi)),\end{aligned}$$

where the last equality comes from Lemma 1. And this shows the transformer with $\boldsymbol{W} + \mathbf{1}\boldsymbol{a}^\top$ gives the same prediction of BMA's. Further, suppose $\boldsymbol{W} = \log \pi + \mathbf{1}\boldsymbol{a}^\top + \boldsymbol{b}\mathbf{1}^\top$. If we have $\boldsymbol{x}_h \sim \mu^\pi, \forall h \in [H]$, then we can prove the term from $\boldsymbol{b}\mathbf{1}^\top$ is also a constant vector asymptotically:

$$\begin{aligned}\boldsymbol{p}_{\texttt{tf}}^h &= \sum_l \boldsymbol{x}_{1:h-1}^{l\top} \boldsymbol{W}_{\texttt{tf}} \boldsymbol{x}_h^l \\ &= \sum_l \boldsymbol{x}_{1:h-1}^{l\top} \log \pi \boldsymbol{x}_h^l + \sum_l \boldsymbol{x}_{1:h-1}^{l\top} \mathbf{1}\boldsymbol{a}^\top \boldsymbol{x}_h^l + \sum_l \boldsymbol{x}_{1:h-1}^{l\top} \boldsymbol{b}\mathbf{1}^\top \boldsymbol{x}_h^l \\ &= \sum_l \boldsymbol{x}_{1:h-1}^{l\top} \log \pi \boldsymbol{x}_h^l + c(\boldsymbol{a}, h)\mathbf{1}_{h-1} + \sum_l \boldsymbol{x}_{1:h-1}^{l\top} \boldsymbol{b}.\end{aligned}$$

For each term of $\sum_l \boldsymbol{x}_{1:h-1}^{l\top} \boldsymbol{b}$, we have

$$\begin{aligned}\frac{1}{L} \sum_l \boldsymbol{x}_{h'}^{l\top} \boldsymbol{b} &= \frac{1}{L} \sum_l \sum_{s\in[d]} 1_{[\boldsymbol{x}_{h'}^l = e_s]} \boldsymbol{b}_s \\ &\overset{L\to\infty}{\to} \mathbb{E}\Big[\sum_{s\in[d]} 1_{[\boldsymbol{x}_{h'}^l = e_s]} \boldsymbol{b}_s \Big] = \sum_{s\in[d]} \mathbb{P}(\boldsymbol{x}_{h'}^l = e_s) \boldsymbol{b}_s \\ &= \mu^{\pi\top} \boldsymbol{b},\end{aligned}$$

where $\mu^{\pi\top}\boldsymbol{b}$ is a constant $d(\boldsymbol{b})$ w.r.t. h'. Using the same technique in Theorem 2 to eliminate this term which goes to infinity, we have the desired result:

$$\lim_{L\to\infty} \sigma(\boldsymbol{p}_{\texttt{tf}}^{h,L}) = \lim_{L\to\infty} \sigma(\boldsymbol{p}_{\texttt{BMA}}^{h,L}(\log \pi)).$$

C.4 Proof of Lemma 3

Proof. Noting that if $p(\cdot)$ and $q(\cdot)$ are two distribution, then by KL divergence's non-negativity we have: $\int_s p(s) \log q(s) \le \int_s p(s) \log p(s)$. Hence we can get:

$$\begin{aligned}\text{LHS} &= \int_{s,s'} \mathbb{P}(\mathrm{x}_h = s, \mathrm{x}_{h'} = s') \log \mathbb{P}(\mathrm{x}_h = s | \mathrm{x}_{pa(h)} = s') \\ &= \int_{s'} \mathbb{P}(\mathrm{x}_{h'} = s') \int_s \mathbb{P}(\mathrm{x}_h = s | \mathrm{x}_{h'} = s') \log \mathbb{P}(\mathrm{x}_h = s | \mathrm{x}_{pa(h)} = s') \\ &\le \int_{s,s'} \mathbb{P}(\mathrm{x}_{h'} = s') H(\mathrm{x}_h | \mathrm{x}_{h'} = s') = -H(\mathrm{x}_h | \mathrm{x}_{h'}),\end{aligned} \tag{24}$$

where $H(\mathrm{x}_h|\mathrm{x}_{h'}) = H(\mathrm{x}_h) - I(\mathrm{x}_h; \mathrm{x}_{h'})$. By Corollary. 1, we have:

$$\text{LHS} \le H(\mathrm{x}_h) - I(\mathrm{x}_h; \mathrm{x}'_h) < H(\mathrm{x}_h) - I(\mathrm{x}_h; \mathrm{x}_{pa(h)}) = \text{RHS}. \tag{25}$$

□

C.5 Proof of Theorem 2

Proof. Recall that the transformer and BMA have the formula in Eq. (2):

$$\tilde{\mathcal{A}}_{h\to h'}^L = \frac{\exp(\sum_l \log p(\boldsymbol{x}_h^l | \boldsymbol{x}_{h'}^l))}{\sum_{h''\in[h-1]} \exp(\sum_l \log p(\boldsymbol{x}_h^l | \boldsymbol{x}_{h''}^l))} = \frac{1}{\sum_{h''} \boldsymbol{v}_{h''\to h'}}. \tag{26}$$

By the central limit theorem, we have

$$\lim_{L\to\infty}\frac{1}{L}\sum_l \log p(\boldsymbol{x}_h^l|\boldsymbol{x}_{h'}^l) = \mathbb{E}[\log p(\boldsymbol{x}_h|\boldsymbol{x}_{h'})] < \mathbb{E}[\log p(\boldsymbol{x}_h|\boldsymbol{x}_{pa(h)})] = \lim_{L\to\infty}\frac{1}{L}\sum_l \log p(\boldsymbol{x}_h^l|\boldsymbol{x}_{pa(h)}^l).$$

Let $\hat{g}_{h,h'} \triangleq \frac{1}{L}\sum_l \log p(\boldsymbol{x}_h^l|\boldsymbol{x}_{h'}^l)$. For all $h'' \neq pa(h)$, we have:

$$\boldsymbol{v}_{h''\to pa(h)} = \exp\Big(\sum_l \log p(\boldsymbol{x}_h^l|\boldsymbol{x}_{h''}^l) - \sum_l \log p(\boldsymbol{x}_h^l|\boldsymbol{x}_{pa(h)}^l)\Big) = \exp\big(L(\hat{g}_{h,h''} - \hat{g}_{h,pa(h)})\big) \to 0$$

as $L\to\infty$ and $\lim_{L\to\infty}(\hat{g}_{h,h''} - \hat{g}_{h,pa(h)}) < 0$. Hence, we have $\lim_{L\to\infty}\tilde{A}_{h\to pa(h)} = 1$. □

C.6 Proof of Theorem 3

Proof. First, the transformer as constructed can be simplified as:

$$\boldsymbol{f}_\theta^{(\texttt{simp})}(\cdot\mid\mathcal{H}) = \pi^\top \boldsymbol{x}_{1:h-1}\sigma\Big(\sum_l \boldsymbol{x}_{1:h-1}^{l\top}\boldsymbol{W}\boldsymbol{x}_h^l\Big) \in \mathbb{R}^d, \tag{27}$$

Considering $\hat{\boldsymbol{p}} = \sum_l \boldsymbol{x}_{1:h-1}^{l\top}\boldsymbol{W}\boldsymbol{x}_h^l = \boldsymbol{0}$ when $\boldsymbol{W} = \boldsymbol{0}$, then $\boldsymbol{p} = \frac{1}{h-1}\boldsymbol{1}_{h-1}$ and:

$$\boldsymbol{f}_{\theta_0}(\cdot\mid\mathcal{H}) = \pi^\top\bar{\mu}(\boldsymbol{x}_{1:h-1}),\ \text{where}\ \bar{\mu}(\boldsymbol{x}_{1:h-1}) = \frac{1}{h-1}\sum_{h'\in[h-1]}\boldsymbol{x}_{h'}. \tag{28}$$

Then based on $\frac{\partial\sigma(\hat{\boldsymbol{p}})}{\partial\hat{\boldsymbol{p}}} = \mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top$, computing the gradient of $\boldsymbol{W}$ w.r.t loss ℓ in Eq. (14) yields:

$$\begin{aligned}
\frac{\partial\ell(\Theta;h,\mathcal{G})}{\partial\hat{\boldsymbol{p}}} &= \mathbb{E}_{\boldsymbol{X}}[(\frac{\boldsymbol{x}_h}{\boldsymbol{f}_{\theta_0}(\boldsymbol{x}_h)+\epsilon})^\top\frac{\partial\boldsymbol{f}_{\theta_0}}{\partial\hat{\boldsymbol{p}}}] \\
&= \mathbb{E}_{\boldsymbol{X}}[(\frac{\boldsymbol{x}_h}{\boldsymbol{f}_{\theta_0}(\boldsymbol{x}_h)+\epsilon})^\top\frac{1}{h-1}(\pi^\top\boldsymbol{x}_{1:h-1} - \pi^\top\bar{\mu}(\boldsymbol{x}_{1:h-1})1_{h-1}^\top)] \\
&\overset{Eq.\ (28)}{=} \frac{1}{h-1}\mathbb{E}_{\boldsymbol{X}}[(\frac{\boldsymbol{x}_h}{\boldsymbol{f}_{\theta_0}(\boldsymbol{x}_h)+\epsilon})^\top(\pi^\top\boldsymbol{x}_{1:h-1} - \boldsymbol{f}_{\theta_0}(\boldsymbol{x}_h)1_{h-1}^\top)] \\
&= \frac{1}{h-1}\mathbb{E}_{\boldsymbol{X}}\Big[\big[\frac{\pi(\boldsymbol{x}_h|\boldsymbol{x}_1)}{\boldsymbol{f}_{\theta_0}(\boldsymbol{x}_h)+\epsilon},\cdots,\frac{\pi(\boldsymbol{x}_h|\boldsymbol{x}_h-1)}{\boldsymbol{f}_{\theta_0}(\boldsymbol{x}_h)+\epsilon}\big] - 1_{h-1}^\top\Big] \\
&= \frac{1}{h-1}\mathbb{E}_{\boldsymbol{X}}\Big[\big[\frac{\pi(\boldsymbol{x}_h|\boldsymbol{x}_1)}{\mu^\pi(\boldsymbol{x}_h)},\cdots,\frac{\pi(\boldsymbol{x}_h|\boldsymbol{x}_h-1)}{\mu^\pi(\boldsymbol{x}_h)}\big] - 1_{h-1}^\top\Big] \in \mathbb{R}^{h-1}.
\end{aligned}$$

Then let $\hat{g}_{h'}^h$ denote h'-th entry in $\frac{\partial\ell(\theta_0;h,\mathcal{G})}{\partial\hat{\boldsymbol{p}}} \in \mathbb{R}^{h-1}$ ($h'\in[h-1]$), we have:

$$\hat{g}_{h'}^h = \frac{1}{h-1}\mathbb{E}_{\boldsymbol{X}}\big[\frac{\pi(\boldsymbol{x}_h|\boldsymbol{x}_h')}{\mu^\pi(\boldsymbol{x}_h)} - 1\big] = \frac{1}{h-1}\Big(\sum_{s,s'}\frac{\pi(s|s')\mathbb{P}_{\boldsymbol{X}}(\boldsymbol{x}_h = s|\boldsymbol{x}_{h'} = s')}{\mu^\pi(s)} - 1\Big). \tag{29}$$

By Cauchy-Schwartz Inequality and Data Processing Inequality, we have:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{X}}\big[\frac{\pi(\boldsymbol{x}_h|\boldsymbol{x}_h')}{\mu^\pi(\boldsymbol{x}_h)} - 1\big] = \sum_{s,s'}\frac{\pi(s|s')\mathbb{P}_{\boldsymbol{X}}(\boldsymbol{x}_h = s|\boldsymbol{x}_{h'} = s')}{\mu^\pi(\boldsymbol{x}_h)} - 1 \\
&\le \frac{1}{2}\big(I_{\chi^2}(\mathrm{x}_h,\mathrm{x}_{h'}) + I_{\chi^2}(\mathrm{x}_h,\mathrm{x}_{pa(h)})\big) \le I_{\chi^2}(\mathrm{x}_h,\mathrm{x}_{pa(h)}) = \mathbb{E}_{\boldsymbol{X}}\big[\frac{\pi(\boldsymbol{x}_h|\boldsymbol{x}_{pa(h)})}{\mu^\pi(\boldsymbol{x}_h)} - 1\big].
\end{aligned} \tag{30}$$

Eq. (30) has shown the desired result $\hat{g}_{h'}^h \le \hat{g}_{pa(h)}^h$.[3] □

[3] Suppose $\hat{\mu}_X := \pi^\top\bar{\mu} = \boldsymbol{f}_{\theta_0}(\boldsymbol{x}_h)$. If we remove the assumption $f_\theta = \pi^\top\bar{\mu} = \mu^\pi$. Lemma 24 in Nichani et al. (2024) shows $\Big|\mathbb{E}_{\boldsymbol{X}}\big[\frac{\pi(\boldsymbol{x}_h|\boldsymbol{x}_h')}{\hat{\mu}_X(\boldsymbol{x}_h)+\epsilon} - 1\big] - \mathbb{E}_{\boldsymbol{X}}\big[\frac{\pi(\boldsymbol{x}_h|\boldsymbol{x}_h')}{\mu^\pi(\boldsymbol{x}_h)} - 1\big]\Big| \lesssim \frac{1}{\sqrt{T_{\mathrm{eff}}}}(\to 0)$. Under Assumption 1 and Strong Data Processing Inequality in Nichani et al. (2024) (Lemma 5), we can prove the non-asymptotic result $\hat{g}_{pa(h)}^h - \hat{g}_{h'}^h \ge \frac{1}{h-1}(\frac{\gamma^3}{2S} - \frac{2C}{\sqrt{T_{\mathrm{eff}}(\lambda)}})$.

Assumption 1 (Assumptions on transition kernel (Nichani et al. (2024), Assumption 1)). *There exist $\gamma > 0, \lambda < 1$ such that the following hold for π:*

- *(Transition lower bounded):* $\min_{s,s'} \pi(s' \mid s) > \gamma$.
- *(Non-degeneracy of chain):* $\|B(\pi)\|_F > \gamma$.
- *(Spectral gap): The spectral gap of π, $1 - \lambda(\pi)$, satisfies $\lambda(\pi) < \lambda$*
- *(Symmetry): For any permutation matrix σ on $\mathcal{S}$, $\sigma^{-1}\pi\sigma =_d \pi$.*
- *(Constant mean):* $\mathbb{E}_\pi[\pi] = \frac{1}{S} 1_S 1_S^T$.

C.7 Proof of Proposition 2

Proof. In dynamical system setting, the transition $P(\cdot|\cdot)$ is given by the pmf of $\mathrm{x}_h|\mathrm{x}_{pa(h)}$:

$$p(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{1}{(2\pi)^{d/2}\left(\frac{\sigma^2}{c^2}\right)^{d/2}} \exp\left(-\frac{c^2}{2\sigma^2}\left\|\boldsymbol{x} - \tfrac{1}{c}\boldsymbol{A}^\top \boldsymbol{y}\right\|_2^2\right), \ \boldsymbol{A} \in \mathcal{O}(\mathbb{R}^d).$$

Then with σ eliminate constant terms in $\log p$, we get the equivalent form in BMA:

$$\begin{aligned} \log p(\boldsymbol{x}_h \mid \boldsymbol{x}_{h'}) &= \frac{c}{\sigma^2}\, \boldsymbol{x}_h^\top \boldsymbol{A}^\top \boldsymbol{x}_{h'} - \frac{1}{2\sigma^2}\, \boldsymbol{x}_{h'}^\top \boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{x}_{h'} + \mathrm{const}(h), \\ \bar{\boldsymbol{p}}^h_{h'} &:= \sum_{l=1}^{L} \left(\frac{c}{\sigma^2}\, \boldsymbol{x}_{h'}^{l\top} \boldsymbol{A} \boldsymbol{x}_h^l - \frac{1}{2\sigma^2}\, \boldsymbol{x}_{h'}^{l\top} \boldsymbol{x}_{h'}^l\right); \quad \mathbb{P}(pa(h)|\boldsymbol{x}_{1:H}^{1:L}) = \sigma(\bar{\boldsymbol{p}}^h). \end{aligned} \tag{31}$$

Eq. (31) gives the BMA logits in the DS setting in a softmax form. We now show that transformers under the observation restriction Eq. (7) cannot represent BMA in this setting.

Recall that, under Eq. (7), the transformer logits are

$$\boldsymbol{p}^h_{\mathtt{tf},h'} = \sum_{l=1}^{L} \boldsymbol{x}_{h'}^{l\top} \boldsymbol{W}_{\mathtt{tf}}\, \boldsymbol{x}_h^l,$$

while the BMA logits are

$$\boldsymbol{p}^h_{\mathtt{BMA},h'} = c_1 \sum_{l=1}^{L} \boldsymbol{x}_{h'}^{l\top} \boldsymbol{A} \boldsymbol{x}_h^l + d \sum_{l=1}^{L} \|\boldsymbol{x}_{h'}^l\|^2, \qquad c_1 \neq 0,\ d \neq 0.$$

Suppose, for contradiction, that the transformer exactly represents BMA, i.e.

$$\sigma(\boldsymbol{p}^h_{\mathtt{tf}}) = \sigma(\boldsymbol{p}^h_{\mathtt{BMA}}) \quad \text{for all DS samples and all } h \in [H].$$

Since softmax is invariant under adding a constant independent of h', this means that for each fixed h there exists a scalar $b = b(h)$ such that

$$\boldsymbol{p}^h_{\mathtt{tf},h'} + b = \boldsymbol{p}^h_{\mathtt{BMA},h'} \quad \text{for all } h' \in [H]. \tag{*}$$

Using the DS model $\boldsymbol{x}_h^l = \frac{1}{c}(\boldsymbol{A}^\top \boldsymbol{x}_{pa(h)}^l + \varepsilon_h^l)$, we expand the logits as

$$\boldsymbol{p}^h_{\mathtt{tf},h'} = \underbrace{\frac{1}{c}\sum_{l=1}^{L} \boldsymbol{x}_{h'}^{l\top} \boldsymbol{W}_{\mathtt{tf}} \boldsymbol{A}^\top \boldsymbol{x}_{pa(h)}^l}_{\text{constant term w.r.t. } h,\, \varepsilon_h} + \underbrace{\frac{1}{c}\sum_{l=1}^{L} \boldsymbol{x}_{h'}^{l\top} \boldsymbol{W}_{\mathtt{tf}} \varepsilon_h^l}_{\text{seperate term with } \varepsilon_h},$$

$$\boldsymbol{p}^h_{\mathtt{BMA},h'} = \underbrace{\frac{c_1}{c}\sum_{l=1}^{L} \boldsymbol{x}_{h'}^{l\top} \boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{x}_{pa(h)}^l + d\sum_{l=1}^{L} \|\boldsymbol{x}_{h'}^l\|^2}_{\text{constant term w.r.t. } h,\, \varepsilon_h} + \underbrace{\frac{c_1}{c}\sum_{l=1}^{L} \boldsymbol{x}_{h'}^{l\top} \boldsymbol{A} \varepsilon_h^l}_{\text{seperate term with } \varepsilon_h}.$$

Conditioning on all variables except $\{\varepsilon_h^l\}_{l=1}^L$, both sides of $(*)$ become affine functions of the Gaussian noises ε_h^l. Since the DS distribution has full support and $(*)$ is assumed to hold for all DS samples, the coefficients of the linear terms in $\{\varepsilon_h^l\}$ must match for all realizations. Since $\{\varepsilon_h^l\}_l$ are independently sampled, conditioning on ε_h^l will eliminate other terms. Seeing ε_h^l as the only free variable, its coefficient should be zero to keep the equation held:

$$\boldsymbol{x}_{h'}^{l\top}(\boldsymbol{W}_{\mathtt{tf}} - c_1\boldsymbol{A})\varepsilon_h^l = 0, \forall \varepsilon_h^l \in \mathbb{R}^d \Rightarrow \boldsymbol{x}_{h'}^{l\top}(\boldsymbol{W}_{\mathtt{tf}} - c_1\boldsymbol{A}) = 0.$$

Since in the DS model each $\boldsymbol{x}_{h'}^l \sim \mathcal{N}(0, I_d)$ is non-degenerate with full support $\boldsymbol{x}_h^l \sim \mathcal{N}(0, I_d)$, which forces

$$\boldsymbol{W}_{\mathtt{tf}} = c_1\boldsymbol{A}.$$

Substituting $\boldsymbol{W}_{\mathtt{tf}} = c_1\boldsymbol{A}$ back into $(*)$, the representation equation is formulated by

$$b + \frac{c_1}{c}\sum_{l=1}^{L}\boldsymbol{x}_{h'}^{l\top}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{x}_{pa(h)}^l = \frac{c_1}{c}\sum_{l=1}^{L}\boldsymbol{x}_{h'}^{l\top}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{x}_{pa(h)}^l + d\sum_{l=1}^{L}\|\boldsymbol{x}_{h'}^l\|^2.$$

Hence

$$b = d\sum_{l=1}^{L}\|\boldsymbol{x}_{h'}^l\|^2 \quad \text{for all } h' \in [H].$$

However, for a DS sample the quantities $\sum_l \|\boldsymbol{x}_{h'}^l\|^2$ vary across h' and across samples, while b is a constant (depending only on h). The only way the above equality can hold for all h' and all DS samples is to have $b = d = 0$, which contradicts the assumption $d \neq 0$ in the BMA logits.

We conclude that no $\boldsymbol{W}_{\mathtt{tf}}$ can make the transformer logits represent the BMA logits for all datasets generated from DS. Therefore, under Eq. (7), transformers cannot represent BMA in the DS setting.

C.8 Proof of $\sum_{s,s'} \bar{\mu}_t(s')\mu^\pi(s)\log\pi(s \mid s') \leq \sum_{s,s'} \mu^\pi(s')\pi(s \mid s')\log\pi(s \mid s'), \forall \bar{\mu}_t \in \Delta^d$

Let $\mathcal{S}$ be a finite state space, let $\pi(\cdot \mid \cdot)$ be a Markov transition kernel on $\mathcal{S}$, and let μ^π be a stationary distribution of π, i.e.,

$$\mu^\pi(s) \;=\; \sum_{s' \in \mathcal{S}} \mu^\pi(s')\,\pi(s \mid s') \quad \text{for all } s \in \mathcal{S}.$$

Fix an arbitrary $\bar{\mu}_t \in \Delta^d$, and for brevity write $\mu := \mu^\pi$. We adopt the convention $0\log 0 = 0$ throughout.

We first upper bound the left-hand side. For any $s' \in \mathcal{S}$, consider the Kullback–Leibler divergence

$$\mathrm{KL}\big(\mu \,\|\, \pi(\cdot \mid s')\big) \;=\; \sum_{s \in \mathcal{S}} \mu^\pi(s)\log\frac{\mu^\pi(s)}{\pi(s \mid s')} \;\geq\; 0.$$

Expanding the inequality $\mathrm{KL}(\mu \,\|\, \pi(\cdot \mid s')) \geq 0$ yields

$$\sum_{s \in \mathcal{S}} \mu^\pi(s)\log\pi(s \mid s') \;\leq\; \sum_{s \in \mathcal{S}} \mu^\pi(s)\log\mu^\pi(s) =: C,$$

where the right-hand side C does not depend on s'. Multiplying both sides by $\bar{\mu}_t(s')$ and summing over s', we obtain

$$\sum_{s,s' \in \mathcal{S}} \bar{\mu}_t(s')\mu^\pi(s)\log\pi(s \mid s') \;\leq\; \sum_{s' \in \mathcal{S}} \bar{\mu}_t(s')C = C = \sum_{s \in \mathcal{S}} \mu^\pi(s)\log\mu^\pi(s). \tag{32}$$

This bound holds for any choice of $\bar{\mu}_t \in \Delta^d$.

Next, we lower bound the right-hand side. For each $s' \in \mathcal{S}$, consider the reverse KL divergence

$$\mathrm{KL}\big(\pi(\cdot \mid s') \,\|\, \mu\big) \;=\; \sum_{s \in \mathcal{S}} \pi(s \mid s')\log\frac{\pi(s \mid s')}{\mu^\pi(s)} \;\geq\; 0.$$

Hence

$$\sum_{s\in\mathcal{S}} \pi(s \mid s') \log \pi(s \mid s') \;\geq\; \sum_{s\in\mathcal{S}} \pi(s \mid s') \log \mu^{\pi}(s).$$

Multiplying by $\mu^{\pi}(s')$ and summing over s' yields

$$\begin{aligned}\sum_{s,s'\in\mathcal{S}} \mu^{\pi}(s')\pi(s \mid s') \log \pi(s \mid s') \;&\geq\; \sum_{s,s'\in\mathcal{S}} \mu^{\pi}(s')\pi(s \mid s') \log \mu^{\pi}(s) \\ &= \sum_{s\in\mathcal{S}} \Big(\sum_{s'\in\mathcal{S}} \mu^{\pi}(s')\pi(s \mid s') \Big) \log \mu^{\pi}(s) \\ &= \sum_{s\in\mathcal{S}} \mu^{\pi}(s) \log \mu^{\pi}(s) = C, \end{aligned} \tag{33}$$

where we used the stationarity of μ, namely $\mu^{\pi}(s) = \sum_{s'} \mu^{\pi}(s')\pi(s \mid s')$.

Combining (32) and (33), we conclude that, for any $\bar{\mu}_t \in \Delta^d$,

$$\sum_{s,s'\in\mathcal{S}} \bar{\mu}_t(s')\mu^{\pi}(s) \log \pi(s \mid s') \;\leq\; C \;\leq\; \sum_{s,s'\in\mathcal{S}} \mu^{\pi}(s')\pi(s \mid s') \log \pi(s \mid s'),$$

which establishes

$$\sum_{s,s'} \bar{\mu}_t(s')\mu^{\pi}(s) \log \pi(s \mid s') \;\leq\; \sum_{s,s'} \mu^{\pi}(s')\pi(s \mid s') \log \pi(s \mid s'), \quad \forall \bar{\mu}_t \in \Delta^d. \qquad \square$$

D Experiment Details

All experiments follow the same training setup unless otherwise specified: sequences are generated from a Markov chain with transition kernel $\pi(\cdot \mid \boldsymbol{s}) \sim \text{Dirichlet}(\alpha \cdot \mathbf{1}_d)$ with $\alpha = 0.1$. We use a batch size of 1024 for training and evaluate on 4096 test samples. Parameters are optimized with Adam (Kingma & Ba, 2017), using learning rate 0.05 for discrete Markov chains and 0.001 for dynamical systems. For gradient-based analysis, we adopt SGD with learning rate 1. Fresh data are sampled at each iteration to avoid memorization, and all implementations are based on JAX.

E Additional Experiment Results on Markov Chain

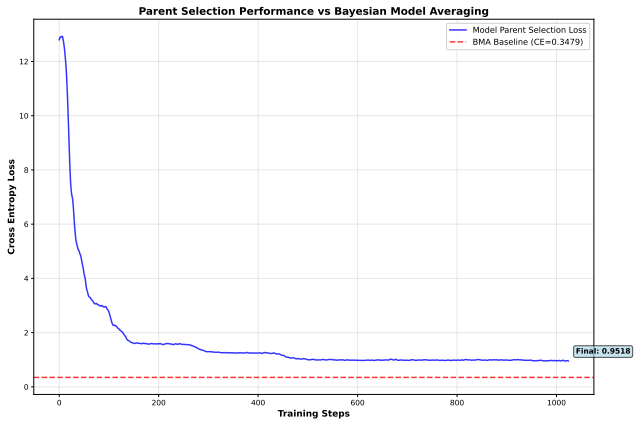


Figure 7: Parent selection $\mathcal{L}_{pa}$ comparison between transformers and BMA during training. The metric is introduced as Eq. (10). Training configuration as the experiment in Fig.2.

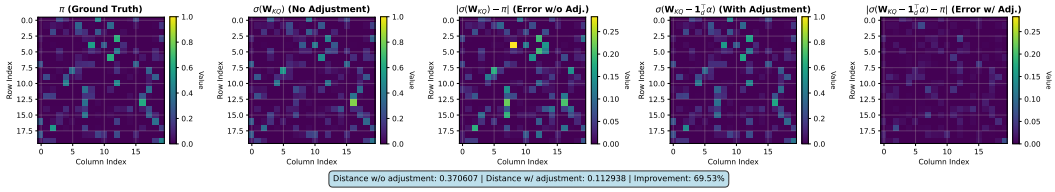


Figure 8: Parameter-Level Comparison between transformer and BMA ($\boldsymbol{W} = \log \pi$). Trainable $\boldsymbol{W}_{KQ}$ and $\boldsymbol{W}_{OV}$. Trained with $d = 20$, $H = 50$, $L = 3$, and 1024 training steps.

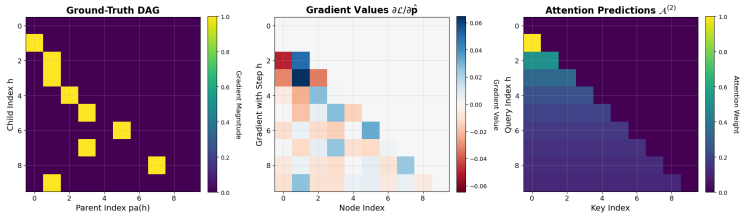


Figure 9: **Gradient Validation of $\frac{\partial \ell}{\partial \tilde{\boldsymbol{p}}}$.** From left to right: ground-truth graph $\mathcal{G}$, the gradients of $\frac{\partial \ell}{\partial \tilde{\boldsymbol{p}}} \in \mathbb{R}^H$ stacked as row vectors, and attention weights $\mathcal{A}_{h\cdot}^{(2)}$ uniformly distributed since $\mathbf{W} = 0$.

F Experiment Results on Dynamical System

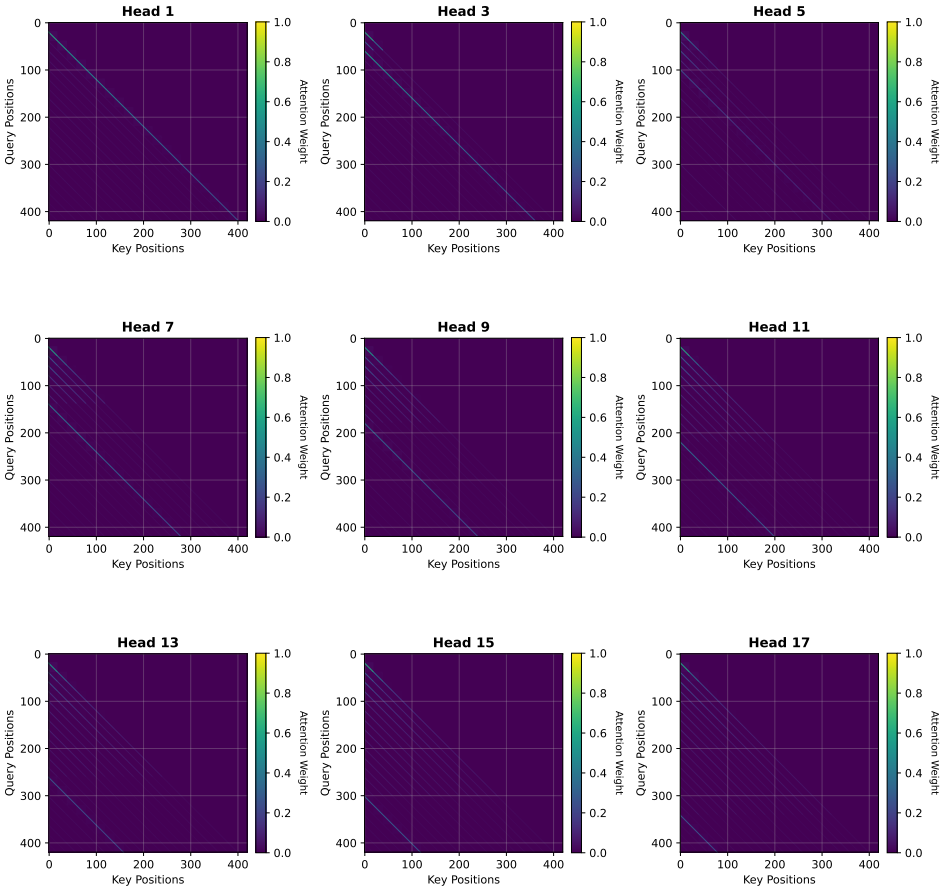


Figure 10: Visualization of 1st-layer Attention $\mathcal{A}^{(1)} \in \mathbb{R}^{T\times T}$. For readability, we visualize only nine of the twenty heads, to better highlight the attention patterns on this long sequence of length 400. The first layer replicates the historical occurrence of the same token. Model trained with $L = 20$ examples, trajectory length $H = 20$, vocabulary size $d = 10$, 20 heads in the first layer, and 2048 training steps. The RPE parameters are initialized with a small positive value (0.5) along the construction direction, and grow to much larger magnitudes after training.

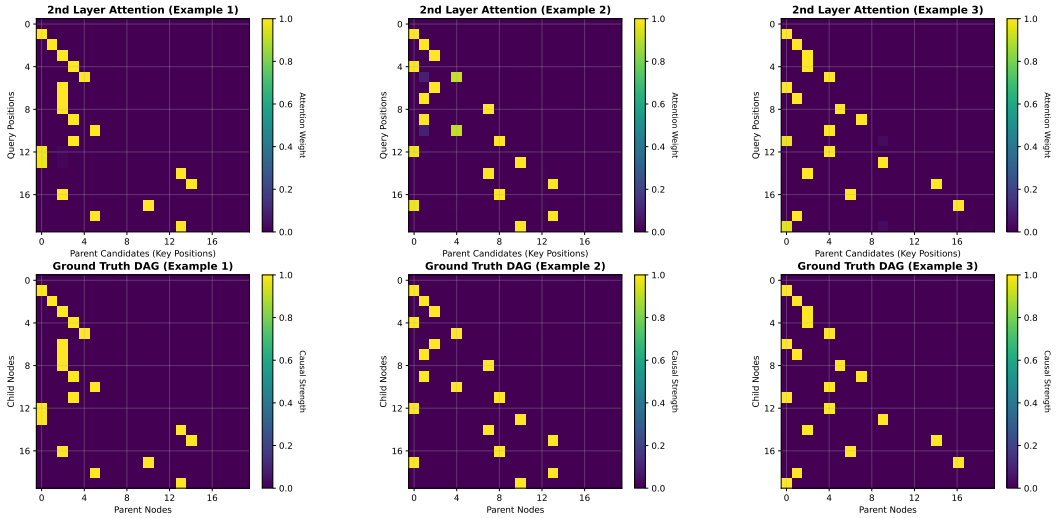


Figure 11: 2nd-Layer Attention $\mathcal{A}^{(2)} \in \mathbb{R}^{H\times H}$ Visualization. Attention patterns matches the groundtruth causal structure in dynamical system setting.

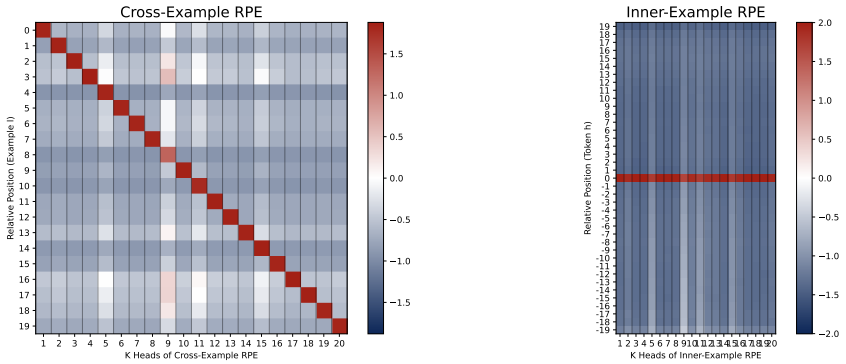


Figure 12: Visualization of first RPE layer. The parameters are consistent with the construction.

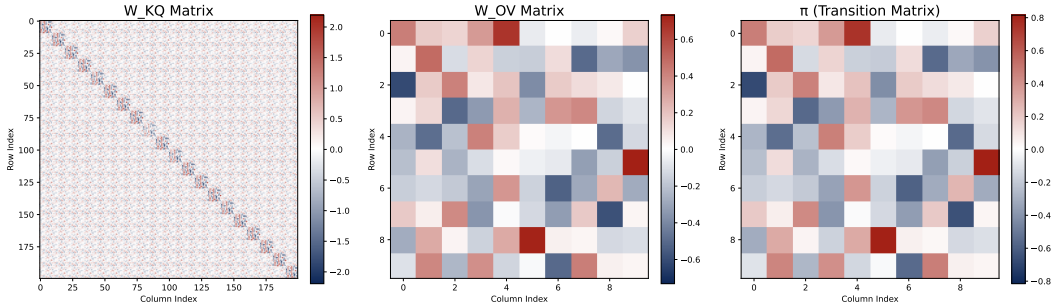


Figure 13: Visualization of 2nd-layer. $\boldsymbol{W}_{KQ} \in \mathbb{R}^{dL\times dL}$ shows noticable non-zero blocks on its diagonal. The occuring block is of size $d \times d$.

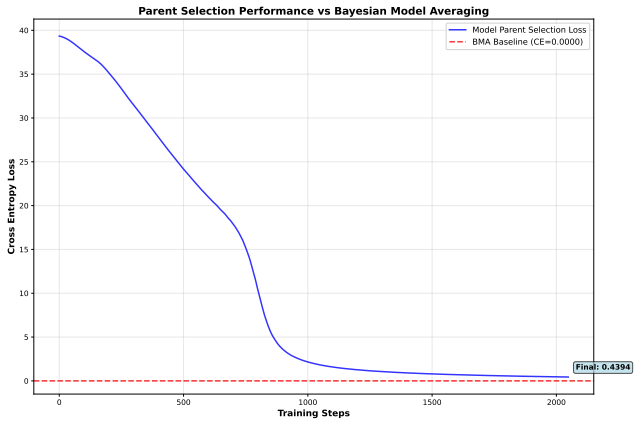


Figure 14: Parent selection loss during training in dynamical system setting.

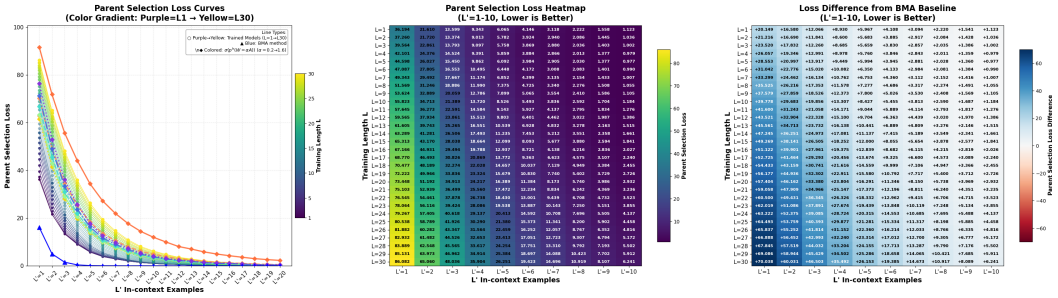


Figure 15: Generalization of parent loss $\{\mathcal{L}_{\text{pa}}^{L'}\}$ for transformers trained with $L \in \{1, \ldots, 30\}$ in dynamical system setting. Trained with $d = 10, H = 15$ and 2048 training steps.

G DISENTANGLED TRANSFORMER WITH ABSOLUTE POSITION EMBEDDING

The disentangled transformer with absolute position embedding (APE) is formulated by:

$$
\begin{aligned}
&\textbf{Embedding Layer:} && \boldsymbol{h}_t^{(0)} = [E(\boldsymbol{w}_t), \mathrm{Pos}(\boldsymbol{w}_t)] = [\boldsymbol{x}_t, \boldsymbol{e}_t], \boldsymbol{H}^{(0)} = [\boldsymbol{h}_1^{(0)}, \ldots, \boldsymbol{h}_T^{(0)}] && \in \mathbb{R}^{d_0} \\
&\textbf{1st Attention (K-head):} && \mathrm{Attn}_t^k(\boldsymbol{H}^{(0)}; \theta) = \sigma\left(\boldsymbol{h}_{1:t-1}^{(0)\top} \boldsymbol{W}_{KQ}^{(1),k} \boldsymbol{h}_t^{(0)}\right)^\top \boldsymbol{h}_{1:t-1}^{(0)\top} \boldsymbol{W}_{OV}^{(1),k} && \in \mathbb{R}^{d}, \\
&\textbf{Disentangled Residual:} && \boldsymbol{h}_t^{(1)} = [\boldsymbol{h}_t^{(0)}, \mathrm{Attn}_t^1(\boldsymbol{H}^{(0)}; \theta), \ldots, \mathrm{Attn}_t^K(\boldsymbol{H}^{(0)}; \theta)] && \in \mathbb{R}^{d_0 + Kd}, \\
&\textbf{2nd Attention (1-head):} && \boldsymbol{f}_{\mathtt{tf}}(\cdot \mid \mathcal{H}_t) = \sigma\left(\boldsymbol{h}_{1:t-1}^{(1)\top} \boldsymbol{W}_{KQ}^{(2)} \boldsymbol{h}_t^{(1)}\right)^\top \boldsymbol{h}_{1:t-1}^{(1)\top} \boldsymbol{W}_{OV}^{(2)} && \in \mathbb{R}^{d}.
\end{aligned} \tag{34}
$$

First, we can see the model parameter $\boldsymbol{W}_{KQ}^{(1),k} \in \mathbb{R}^{d_0 \times d_0}$ where $d_0 = d + T$ and T is the sequence length. The total number of parameters in the first layer is $O(d^2 + H^2 L^2)$ compared to $O(H + L)$ parameters of the model with RPE in Eq. (5). The redundancy of parameters may lead to difficulties of interpreting the mechanism of transformers. Besides, since for disentangled transformer with APE, the embedding dimension is proportional to the length of input sequence, this may make it difficult for us to interpret transformers' mechanism on longer sequence tasks.

As for this transformer, we first provide a theoretical construction which is consistent with our construction for RPE model in Theorem 1. Empirically, we show this transformer can successfully select causal tokens. Besides, we provide results of trainable transformers showing alignments with our construction in attention visualization and parameter verification.

G.1 THEORETICAL CONSTRUCTION

In this section, we provide a construction demonstrating how the proposed two-layer architecture possesses the capacity to implement the specific causal selection mechanism derived in our analysis. Let the input embedding dimension be $d_0 = d+T$, where d is the token dimension, T is the sequence length (due to absolute position embedding) and an input sequence contains $L+1$ examples of length-L chain $T = H(L+1)$. Suppose $\mathcal{N}_{L+1}$ denotes the set of nodes from the last example, i.e., we have $\mathcal{N}_{L+1} = \{t \in T \mid \exists h \in [H], t = HL + h\}$.

G.1.1 LAYER 1: MULTI-HEAD ATTENTION CONSTRUCTION

The first layer consists of K attention heads ($K \le L$). The Query-Key matrix $\boldsymbol{W}_{KQ}^{(1),k}$ will attend to specific predecessor tokens based on position. We construct it as a block matrix where the active interaction terms are confined to the position-embedding subspace:

$$
\boldsymbol{W}_{KQ}^{(1),k} = \left[\begin{array}{c|c} 0_{d\times d} & 0_{d\times T} \\ \hline 0_{T\times d} & \tilde{\boldsymbol{W}}_{KQ}^{(1),k} \end{array}\right], \tilde{\boldsymbol{W}}_{KQ}^{(1),k} = \beta \left[\begin{array}{c|c} 0_{T\times HL} & \begin{array}{c} 0_{H\times H} \\ \vdots \\ I_{H\times H} \\ \vdots \\ 0_{H\times H} \end{array} \end{array}\right] \text{ (k-th block active).} \tag{35}
$$

From this construction, if $\beta \to \infty$, the attention score of the first attention layer is given by:

$$
\mathcal{A}_{ij}^{(1),k} = \begin{cases} \frac{1}{i} \mathbf{1}_{[j<i]}, & \text{if } i \notin \mathcal{N}_{L+h}, \\ \mathbf{1}_{[j=kH+h]}, & \text{if } i \in \mathcal{N}_{L+h},\ i = LH + h. \end{cases} \tag{36}
$$

For the value projection, $\boldsymbol{W}_{OV}^{(1),k}$ will propagate the semantic content of the attended tokens:

$$
\boldsymbol{W}_{OV}^{(1),k} = \left[\begin{array}{c} I_{d\times d} \\ \hline 0_{T\times d} \end{array}\right] \in \mathbb{R}^{(d+T)\times d}. \tag{37}
$$

And the output of the first attention will be:

$$
\mathrm{Attn}_i^k(\boldsymbol{H}^{(0)}; \theta) = \mathcal{A}_{i\to\cdot}^{(1),k} \boldsymbol{h}_{1:T}^{(0)T} \boldsymbol{W}_{OV}^{(1),k} = \begin{cases} \bar{\mu}(\boldsymbol{x}_{1:i-1}), & \text{if } i \notin \mathcal{N}_{L+h}, \\ \boldsymbol{x}_h^k, & \text{if } i \in \mathcal{N}_{L+h},\ i = LH + h. \end{cases} \tag{38}
$$

G.1.2 Disentangled Residual Stream

Unlike standard summation residuals, this disentangled transformer employ a concatenation strategy. Nichani et al. (2024) proved this transformer is actually equivalent to a decoder based attention-only transformer (Theorem 3). The output of the first layer is the concatenation of the original input and the outputs of all K heads:

$$\boldsymbol{h}_t^{(1)} = \left[\boldsymbol{h}_t^{(0)}; \text{Attn}_t^1, \ldots, \text{Attn}_t^K \right] \in \mathbb{R}^{d_0 + Kd}. \tag{39}$$

The dimension of the second layer input is $d_1 = d_0 + Kd = d + T + Kd$.

G.1.3 Layer 2: Single-Head Attention Construction

The second layer employs a single attention head to aggregate the evidence collected by the K heads in the previous layer.

$$\boldsymbol{W}_{KQ}^{(2)} = \left[\begin{array}{c|c|c} 0_{d\times d} & 0_{d\times T} & 0_{d\times Kd} \\ \hline 0_{T\times d} & 0_{T\times T} & 0_{T\times Kd} \\ \hline 0_{Kd\times d} & 0_{Kd\times T} & \tilde{\boldsymbol{W}}_{KQ}^{(2)} \end{array}\right], \tilde{\boldsymbol{W}}_{KQ}^{(2)} = \begin{bmatrix} \log \pi & 0_{d\times d} & \cdots & 0_{d\times d} \\ 0_{d\times d} & \log \pi & \cdots & 0_{d\times d} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{d\times d} & 0_{d\times d} & \cdots & \log \pi \end{bmatrix} \tag{40}$$

Finally, the output projection $\boldsymbol{W}_{OV}^{(2)}$ projects the aggregated context back to the semantic space by:

$$\boldsymbol{W}_{OV}^{(2)} = \left[\begin{array}{c} \log \pi \\ \hline 0_{T\times d} \\ \hline 0_{Kd\times d} \end{array}\right] \in \mathbb{R}^{d_1 \times d}. \tag{41}$$

From the Eq. (38) and (39), we can see that $\boldsymbol{h}_t^{(1)} = [\boldsymbol{h}_t^{(0)}; \boldsymbol{x}_h^1, \ldots, \boldsymbol{x}_h^K]$ if $t \in \mathcal{N}_{L+1}, t = LH + h$ else $\boldsymbol{h}_t^{(1)} = [\boldsymbol{h}_t^{(0)}; \bar{\mu}_t, \ldots, \bar{\mu}_t]$. Then, the attention weight of the second layer for any $i \in \mathcal{N}_{L+1}$, $i = LH + h$ of our interests, is given by:

$$\tilde{\mathcal{A}}_{ij}^{(2)} = \begin{cases} \sum_{k=1}^K \log \pi(x_h^k | x_{h'}^k), & \text{for } j \in \mathcal{N}_{L+1},\ j = LH + h', \\ \sum_{k=1}^K \bar{\mu}_j^\top \log \pi\, x_h^k, & \text{for } j \notin \mathcal{N}_{L+1}. \end{cases} \tag{42}$$

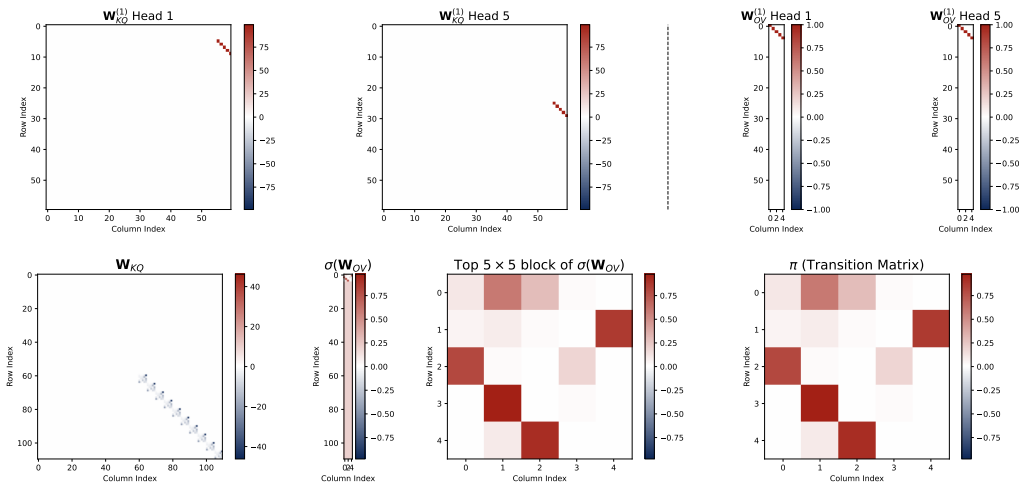
So for $i, j \in \mathcal{N}_{L+1}$, we have $\tilde{\mathcal{A}}_{ij}^{(2)} = \sum_{k\in[K]} \log \pi(\boldsymbol{x}_h^k | \boldsymbol{x}_{h'}^k)$ aligned with Theorem 1.

Furthermore, suppose $K = L$, i.e., we use L examples to infer the causal structure, and the Markov chain is stationary $\boldsymbol{x}_h \sim \mu^\pi$. As $L \to \infty$, for any $j \notin \mathcal{N}_{L+1}$, we have $\tilde{\mathcal{A}}_{ij}^{(2)} \to \sum_{s,s'} \bar{\mu}_j(s') \mu^\pi(s) \log \pi(s|s') \le \sum_s \mu^\pi(s) \log \mu^\pi(s)$. While for the true parent token $t = HL + pa(h)$, we have $\tilde{\mathcal{A}}_{it} \to \sum_{s,s'} \mu^\pi(s') \pi(s|s') \log \pi(s|s')$ which is larger than $\sum_s \mu^\pi(s) \log \mu^\pi(s)$. The above qunatity relation is drawn by non-negativity of KL divergence, whose detailed proof is provided in Appendix C.8. Then, the attention weights of the second layer can select the causal parent token $\mathcal{A}_{i \to pa(i)} \to 1$. And $\boldsymbol{W}_{OV}^{(2)}$ predicts the transition.

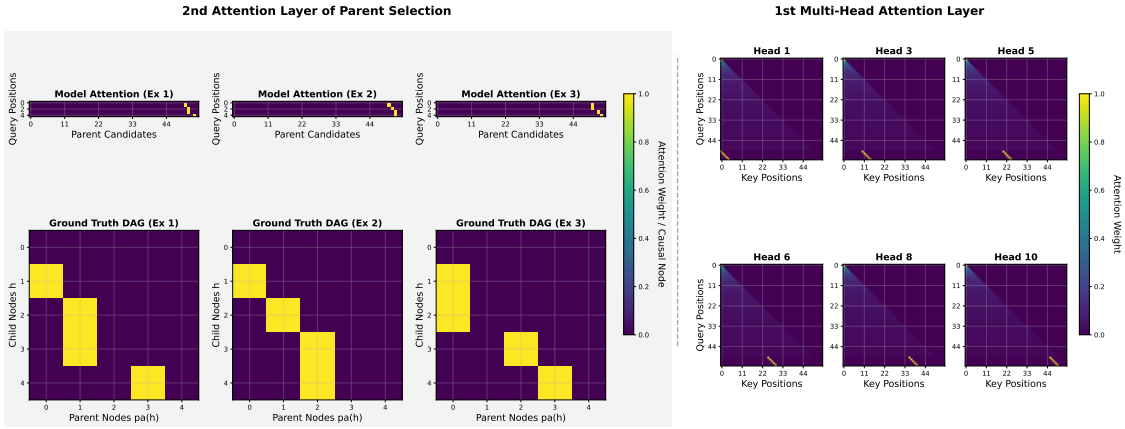
Empirical Verfication. We show the parameter visualization of the construction in Fig. 16a and its empirical attention visualization of parent selection in Fig. 16b. In Fig. 17, the constructed model shows precise parent selection accuracy (cross-entropy loss 0.0706) which is very close to the target algorithm BMA's (ce loss 0.0473).

G.2 Experiments of Trainable Transformers

In the following, we train the standard disentangled transformer formulated by Eq.(34). To show the alignment with theoretical interpretation, we use three strategies to initialize the network: (a)



(a) The parameter visualization of theoretical construction.



(b) Attention pattern visualization of theoretical construction. The first layer (right) copies the previous L examples to the hidden space of the last one $L + 1$. For the second layer (left), the attention weights attend to the correct causal parents which are located in the last 5 columns. The queries don't attend to the keys from first L examples.

Figure 16: Parameter visualization and attention pattern visualization of theoretical construction.

fully random initialization: all the parameters are initialized randomly with Gaussian distribution; (b) block-amplified random initialization: parameters are initialized randomly (of scale 0.1), while the targeted block of the attention projection matrix is assigned a larger magnitude (of scale 0.5) to introduce an inductive bias; (c) direction-consistent initialization: parameters are initialized such that the dominant blocks point in the analytically derived construction direction, still allowing model learning to refine the magnitudes (intial magnitudes: $0.2\times$ optimal parameters).

We first compare the parent token prediction performance of these models during the training process in Fig. 17. The results show that the 2-layer transforemr is fully capable of selecting causal parents in its 2nd-layer attention head.

Then we visualize the attention pattern of the trained model in Fig. 19. For the first attention layer, the figure shows query from the last example $L+1$ mostly attend to one example among L context examples, while some heads demonstrate the degeneration with uniform attention to previous tokens. For the second attention layer, the transformers with different initializations all show their noticable capability of predicting causal parents. Further, we visualize all the parameters of the transformer in Fig. 18. We can see some alignments between the construction in Fig. 16a and the trained parameters. Since the transformer with absolute position embedding has far more parameters of $(\{\boldsymbol{W}_{KQ}^{(1),k}, \boldsymbol{W}_{OV}^{(1),k}\}_k, \boldsymbol{W}_{KQ}^{(2)}, \boldsymbol{W}_{OV}^{(2)})$ than the one with RPE, the full interpretation of its first layer is difficult. For the second layer, the parameter $\boldsymbol{W}_{KQ}^{(2)}$ also shows the diagonal pattern consistent with construction and $\boldsymbol{W}_{OV}^{(2)}$ shows the $\log \pi$ pattern.

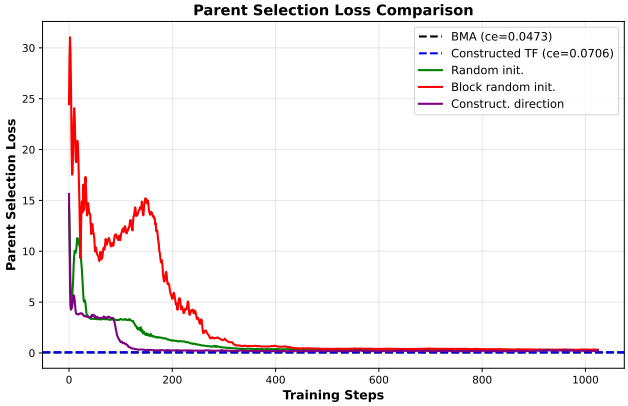
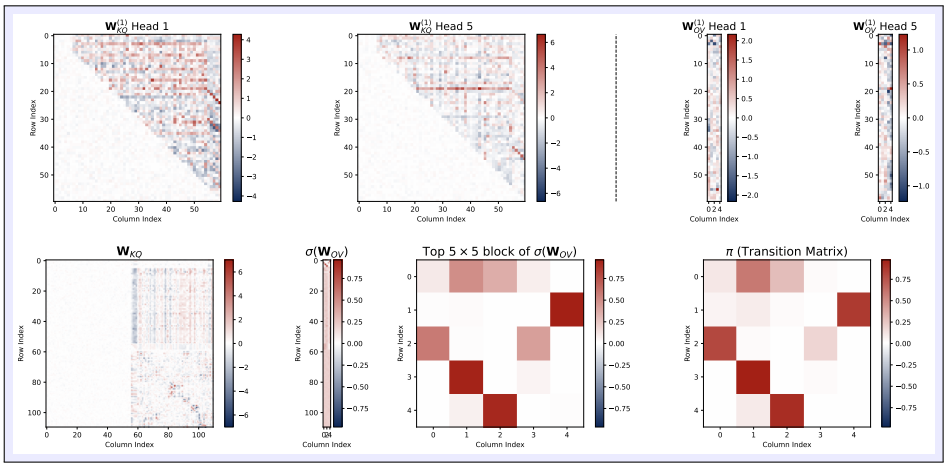
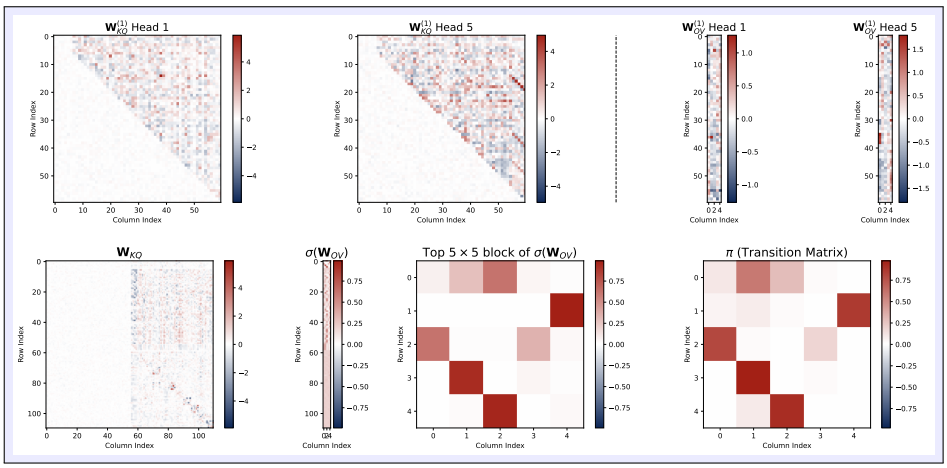


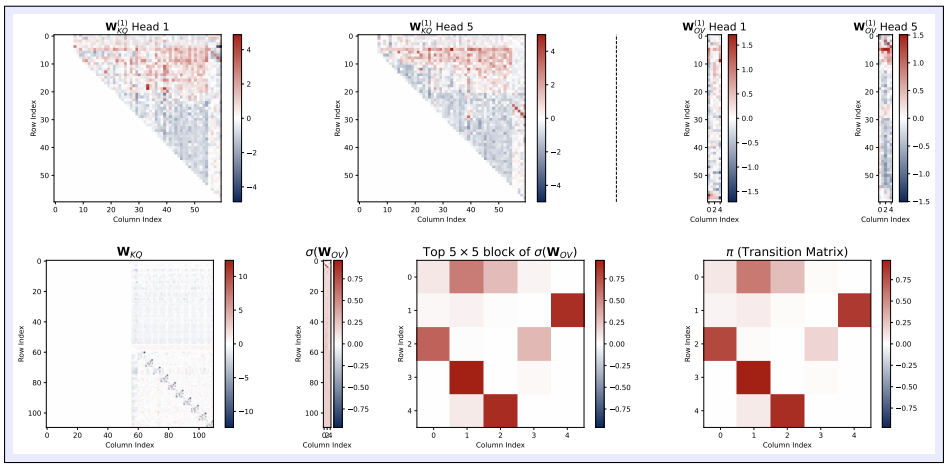
Figure 17: Parent selection loss $\mathcal{L}_{pa}$ of the transformer with absolute position embedding and different initialization strategies.



(a) With Fully Random Initialization. Head 1 and 5 of the first layer $\boldsymbol{W}_{KQ}^{(1)}$ exhibits an identity submatrix (5×5) at the last column.

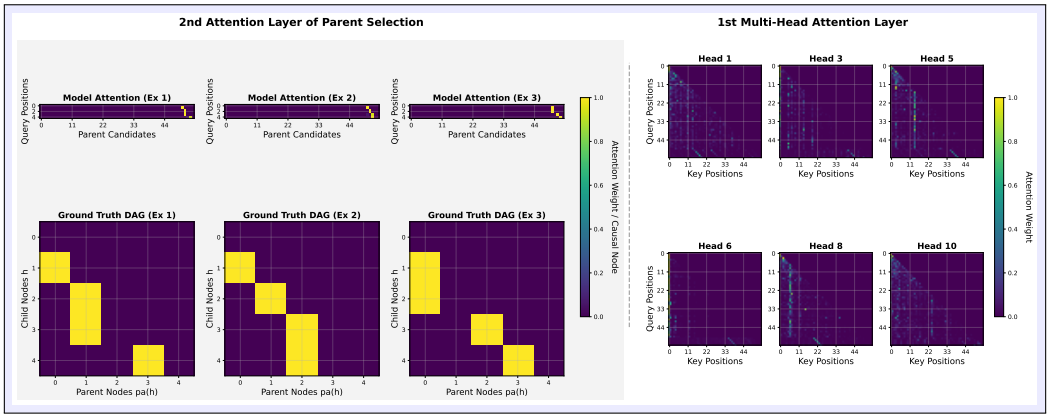


(b) With Block-Amplified Random Initialization. Head 1 of $\boldsymbol{W}_{KQ}^{(1)}$ degenerate which can be verified in attention visualization Fig. 19b (Head 1). Head 5 shows multiple identity submatrices which possibly suggests superposition.

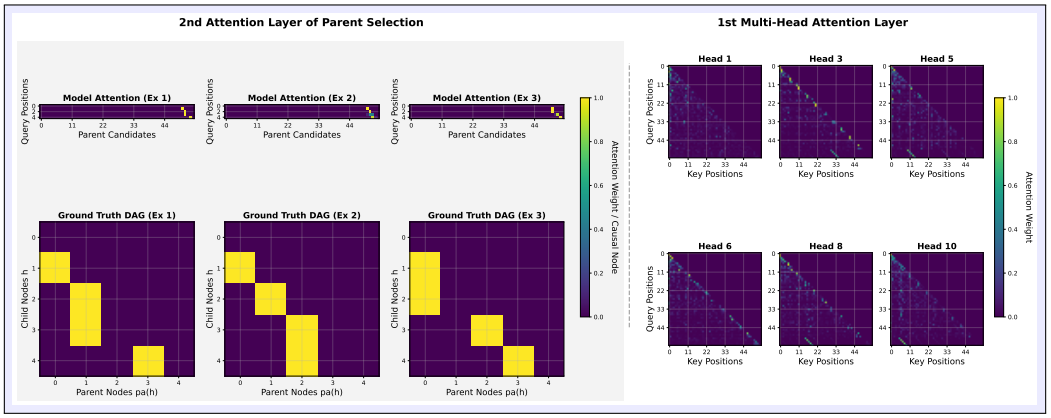


(c) With Direction-Consistent Initialization. Head 1 and 5 of the first layer $\boldsymbol{W}_{KQ}^{(1)}$ exhibits an identity submatrix (5×5) at the last column which is aligned with the theoretical construction.

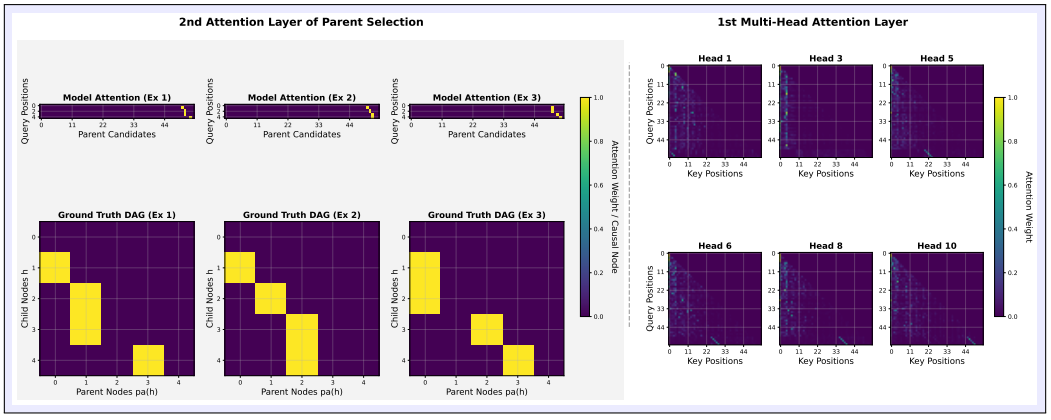
Figure 18: Parameter visualization of trained transformer with absolute position embedding. The second layer shows strong alignment in diagonal patterns of $\boldsymbol{W}_{KQ}$ and $\log \pi$ pattern of $\boldsymbol{W}_{OV}$.



(a) With Fully Random Initialization. In the first layer, Head 1, 3, 5, 6 and 8 of KQ matrices copies tokens from previous examples (to the query token of the last example), while Head 10 degenerates showing uniform attention (uniform features are seen as constants eliminated by 2nd softmax attention layer).



(b) With Block-Amplified Random Initialization. In the first layer, Head 3, 5, 8 and 10 of KQ matrices copies tokens from previous examples, while Head 1 degenerates showing uniform attention.



(c) With Direction-Consistent Initialization. In the first layer, Head 1, 5, 6, 8 and 10 of KQ matrices copies tokens from previous examples, while Head 3 degenerates showing uniform attention.

Figure 19: Attention pattern visualization of trained transformer with absolute position embedding.

H Disentangled Transformer with APE Variant

H.1 Model Architecture

For the variant of two types of APE, we consider the disentangled transformer simplified by eliminating some components added to residual stream. The transformer structure we consider below can be seen as substituing the position embedding of sturcture Eq. (34) and simplify the model by assuming zero blocks in model weights.

$$
\begin{aligned}
&\textbf{Embedding Layer:} && \tilde{\boldsymbol{h}}_t^{(0)} = [\mathrm{Pos}_L(\boldsymbol{w}_t), \mathrm{Pos}_H(\boldsymbol{w}_t)] && \in \mathbb{R}^{d_0} \\
&\textbf{1st Attention (K-head):} && \mathrm{Attn}_t^k = \sigma(\tilde{\boldsymbol{h}}_{1:t-1}^{(0)\top} \boldsymbol{W}_{KQ}^{(1),k} \tilde{\boldsymbol{h}}_t)^{(0)\top} \boldsymbol{x}_{1:t-1}^\top \boldsymbol{W}_{OV}^{(1),k} && \in \mathbb{R}^d, \\
&\textbf{Disentangled Residual:} && \tilde{\boldsymbol{h}}_t^{(1)} = [\mathrm{Attn}_t^1, \ldots, \mathrm{Attn}_t^K] && \in \mathbb{R}^{Kd}, \\
&\textbf{2nd Attention (1-head):} && \boldsymbol{f}_{\mathtt{tf}}(\cdot \mid \mathcal{H}_t) = \sigma\left(\tilde{\boldsymbol{h}}_{1:t-1}^{(1)\top} \boldsymbol{W}_{KQ}^{(2)} \tilde{\boldsymbol{h}}_t^{(1)}\right)^\top \boldsymbol{x}_{1:t-1}^\top \boldsymbol{W}_{OV}^{(2)} && \in \mathbb{R}^d,
\end{aligned}
\tag{43}
$$

where we simply assume $\boldsymbol{W}_{OV}^{(1),k} = I_d$. Since the difference lies in the positional embedding, the construction in Appendix G remains valid which can exhibit capabilities in causal token selection empirically. Besides, we train this transformer under the same Markov chain setup as in the transformer with RPE experiments, and obtain consistent results as shown below.

H.2 Experiment Results

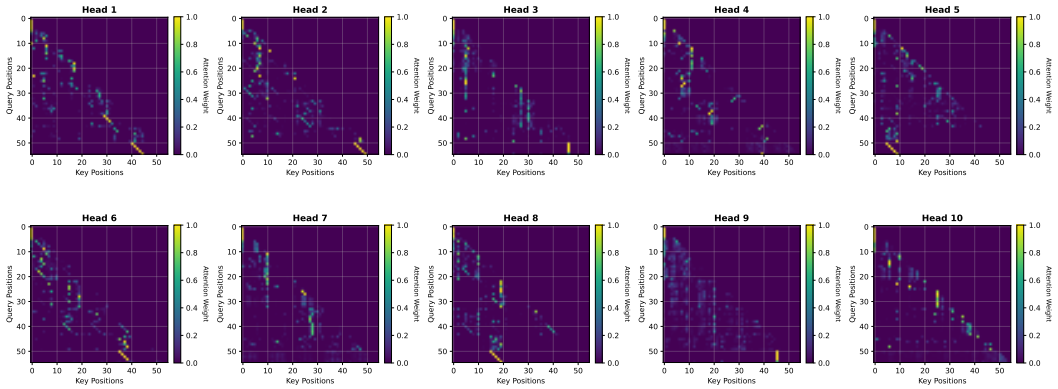


Figure 20: 1st-Layer Attention Visualization of transformers in Eq. (43). Heads 1, 2, 5, 6 and 8 exihibits the diagonal block pattern at the last rows performing the copying mechanism, while Heads 4, 7 and 10 degenerate to uniform attenion. Heads 3 and 9 gives uniform outputs not influencing the 2nd attention layer (eliminated by softmax attention). Trained with $H = 5, L = 10, d = 5$ and 10000 training steps.

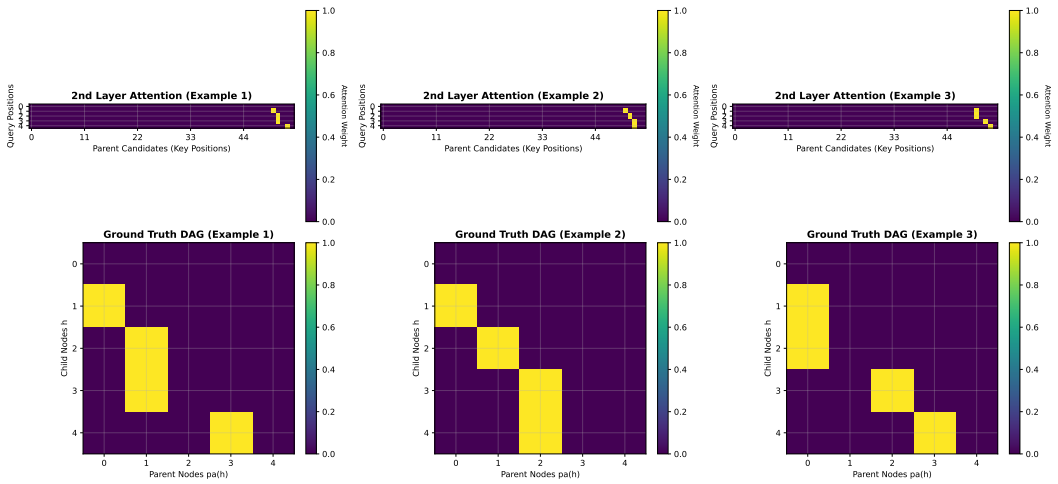


Figure 21: Visualization of 2nd-attention layer. Queries are from the last example $\boldsymbol{x}_{1:H}^{L+1}$. Keys are $\boldsymbol{x}_{1:T} = \boldsymbol{x}_{1:H}^{1:L}$ the whole sequence. Attention layer of disentangled transformer can recognize the causal structure in-context.

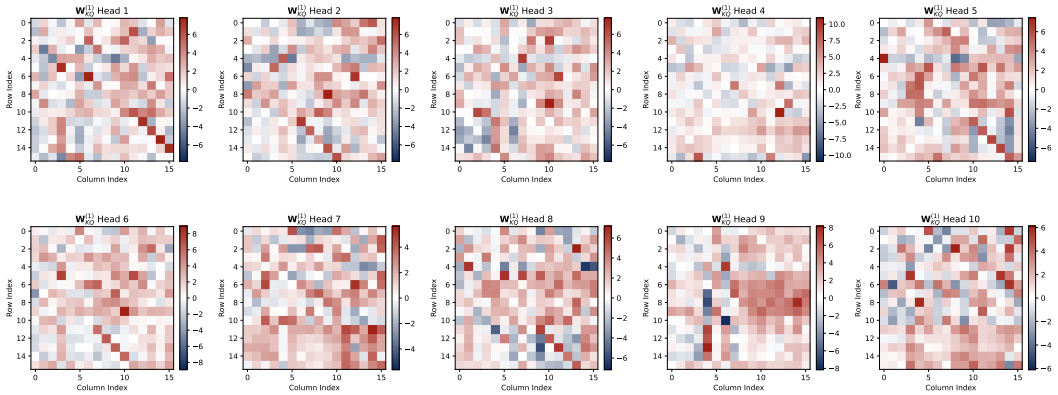


Figure 22: Parameter visualization of the first attention layer $\boldsymbol{W}_{KQ}^{(1),k}$ (10 heads in total). Full interpretation is still challenging for huge parameter space. The attention-level behavior understanding can be referred to Fig. 20.

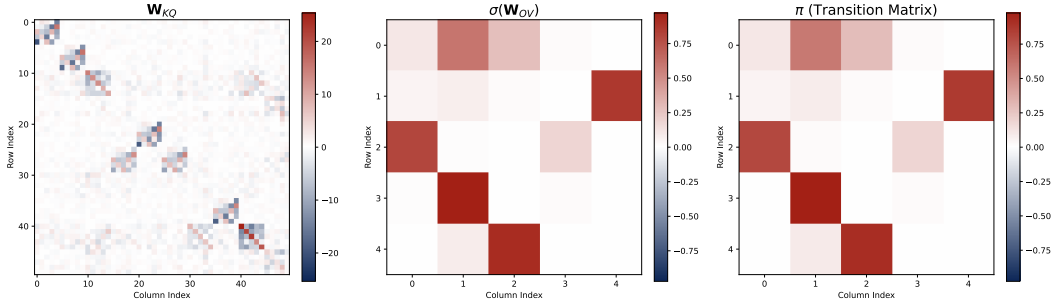


Figure 23: Parameter visualization of the second attention layer $\boldsymbol{W}_{KQ}^{(2)}, \boldsymbol{W}_{OV}^{(2)}$. In the variant of two types of Absolute position embedding, the second layer also shows strong alignment in diagonal patterns of $\boldsymbol{W}_{KQ}$ and $\log \pi$ pattern of $\boldsymbol{W}_{OV}$.

I STANDARD TRANSFORMER WITH FEEDFORWARD NEURAL NETWORK

In this section, we consider a standard 2-layer transformer with FFN layer as follows.

$$\begin{aligned}
&\textbf{Learnable Embedding:} && \boldsymbol{h}_t^{(0)} = \mathbf{Emb_V}(\boldsymbol{w}_t) + \mathbf{Emb_P}(\boldsymbol{w}_t), && \in \mathbb{R}^{d'} \\
&\textbf{MHA Layer \& Residual:} && \tilde{\boldsymbol{h}}_t^{(l)} = \boldsymbol{h}_t^{(l)} + \mathrm{MHA}_t(\boldsymbol{H}^{(l)}; \boldsymbol{W}_{KQ}, \boldsymbol{W}_{OV}), && \in \mathbb{R}^{d'}, \\
&\textbf{FFN Layer \& Residual:} && \boldsymbol{h}_t^{(l+1)} = \tilde{\boldsymbol{h}}_t^{(l)} + \mathrm{FFN}_t(\tilde{\boldsymbol{H}}^{(l)}; \boldsymbol{W}, \boldsymbol{b}) && \in \mathbb{R}^{d'}, \\
&\textbf{Unembedding Layer:} && \boldsymbol{f}_{\mathtt{tf}}(\cdot \mid \mathcal{H}_t) = \boldsymbol{W}_U \boldsymbol{h}_t^{(L)} && \in \mathbb{R}^{d},
\end{aligned} \tag{44}$$

where $\boldsymbol{H}^{(l)} = [\boldsymbol{h}_1^{(l)}, \ldots, \boldsymbol{h}_T^{(l)}]$, the multi-head attention (MHA) is formulated by

$$\mathrm{MHA}_t(\boldsymbol{H}^{(l)}; \theta) = \sum_k \sigma\left(\boldsymbol{h}_{1:t-1}^{(l)\top} \boldsymbol{W}_{KQ}^{(l),k} \boldsymbol{h}_t^{(l)}\right)^\top \boldsymbol{h}_{1:t-1}^{(l)\top} \boldsymbol{W}_{OV}^{(l),k}, \tag{45}$$

and the FFN layer

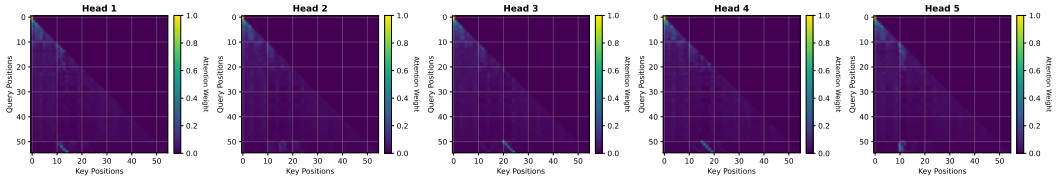
$$\mathrm{FFN}_t(\tilde{\boldsymbol{H}}^{(l)}; \theta) = \boldsymbol{W}_2\, \mathrm{ReLU}(\boldsymbol{W}_1 \tilde{\boldsymbol{h}}_t^{(l)} + \boldsymbol{b}_1) + \boldsymbol{b}_2. \tag{46}$$

We consider the two-layer transformer $L = 2$ with K heads in the first layer and one head in the second.[4] For the task, the input sequence consists of $M = 10$ in-context examples of Length-H Markov chains with $d = 5$ states and the total length $T = H(M + 1)$. We set the hidden dimension as $d' = 128$. For initialization, the parameters $\boldsymbol{W}$ of the transformer is initialized randomly by Gaussian intialization: $\boldsymbol{W}_{ij} \sim \mathcal{N}(0, 1/d_{\boldsymbol{W}})$ where $d_{\boldsymbol{W}}$ is decided by the dimension of $\boldsymbol{W}$. We optimize the model using AdamW with a learning rate of 1×10^{-3} and a weight decay of 1×10^{-4}. Fresh data are sampled at each iteration of training without repetition.

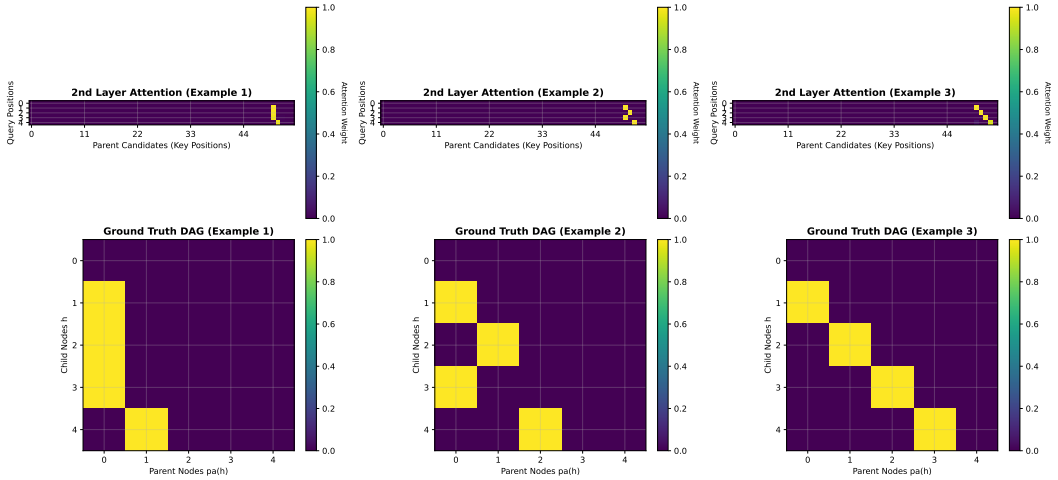
Experiment Results. We train two transformers with 5000 steps and $K = 5$ or 10 heads in the first layer. We observe the attention weights of the first layer visualized in Fig. 24a and Fig. 25a implement the copying mechanism where the features of one context example are copied to the position of last example $M + 1$: the heads of the first layer show a diagonal submatrix occuring at the last several rows of example $M + 1$. Except for these, the remainings mainly show degenerated attention patterns at the rows of the last example $M + 1$. In the visualization of the second layer, we find that the trained standard transformer with MLPs can recognize the causal parents in its attention weights of the 2nd layer. The aligned attention pattern and graph groundtruth in Fig. 24b and Fig. 25b supports our construction of how transformers can handle with in-context causal learning.

Quantitive Results. We provide the results regarding how accurate transformers during training can select random parents in its second attention layer in Fig. 26. We use the cross-entropy loss as the evaluation metric for the accuracy and compare the trained transformers with BMA. We observe during training process, standard transformers gradually acquire the capability of in-context causal learning and approximate the loss of BMA.

[4] Our implementation is based on the codebase provided by Nichani et al. (2024).

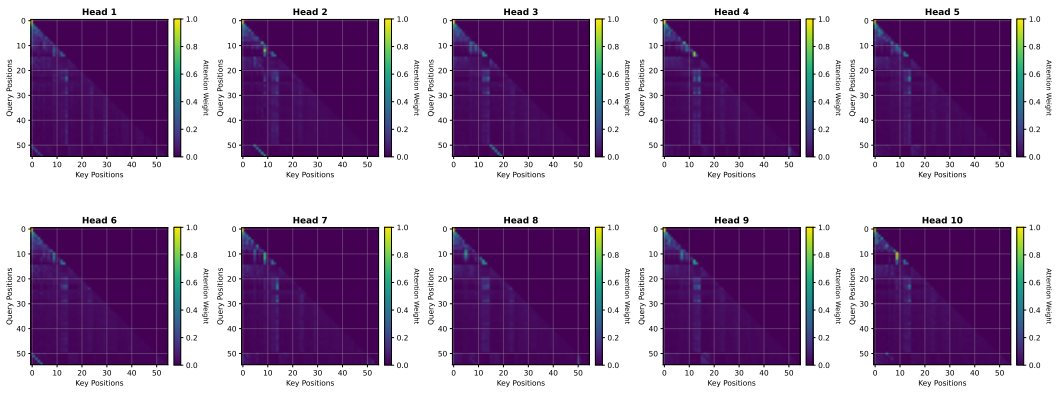


(a) Visualization of first multi-head attention layer. Heads 1, 3 and 4 show the diagonal block at the rows of the last example. Information from previous examples is copied to the hidden space of the last example.

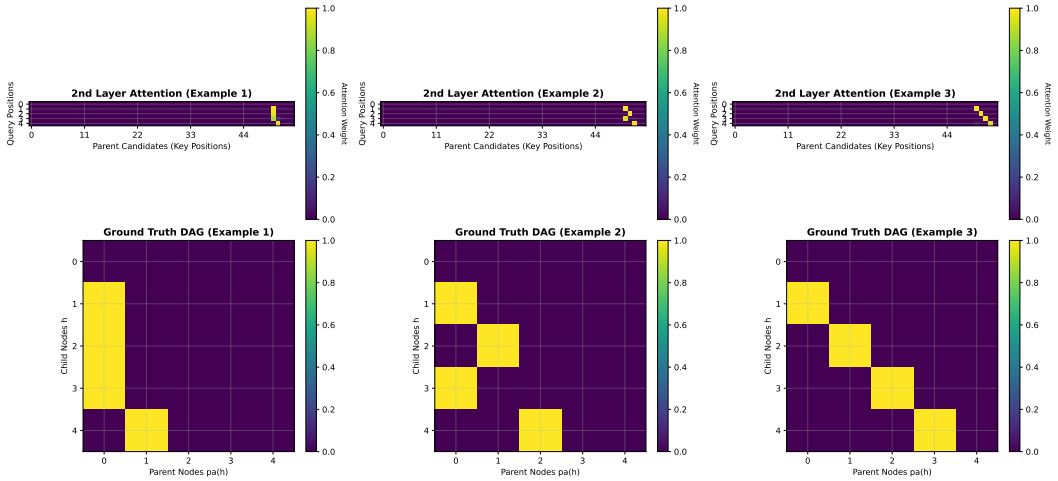


(b) Visualization of second attention layer. Queries are from the last example $\boldsymbol{x}_{1:H}^{L+1}$. Attention layer of standard transformer can recognize the causal structure in-context.

Figure 24: Attention visualization of standard transformer with MLP (5 heads).



(a) Visualization of first multi-head attention layer. Heads 1, 2, 3 and 6 show the diagonal block at the rows of the last example. Information from previous examples is copied to the hidden space of the last example.



(b) Visualization of second attention layer. Queries are from the last example $\boldsymbol{x}_{1:H}^{L+1}$. Attention layer of standard transformer can recognize the causal structure in-context.

Figure 25: Attention visualization of standard transformer with MLP (10 heads).

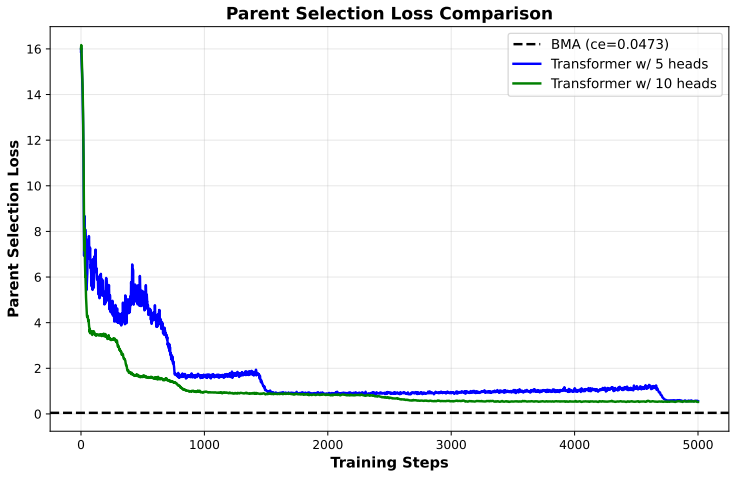


Figure 26: Parent selection loss $\mathcal{L}_{pa}$ of the standard transformer with learnable position embedding and MLP (5 or 10 heads in the first layer). During training, standard transformers gradually acquire the capability of in-context causal learning and approximate the loss of BMA.