Active-Dormant Attention Heads: Mechanistically Demystifying Extreme-Token Phenomena in LLMs

Tianyu Guo* Druv Pai* Yu Bai† Jiantao Jiao* Michael I. Jordan*
Song Mei*

October 18, 2024

Abstract

Practitioners have consistently observed three puzzling phenomena in transformer-based large language models (LLMs): *attention sinks*, *value-state drains*, and *residual-state peaks*, collectively referred to as *extreme-token phenomena*. These phenomena are characterized by certain so-called "sink tokens" receiving disproportionately high attention weights, exhibiting significantly smaller value states, and having much larger residual-state norms than those of other tokens. These extreme tokens give rise to various challenges in LLM inference, quantization, and interpretability.

We elucidate the mechanisms behind extreme-token phenomena. First, we show that these phenomena arise in very simple architectures—transformers with one to three layers—trained on a toy model, the Bigram-Backcopy (BB) task. In this setting, we identify an *active-dormant mechanism*, where attention heads become sinks for specific input domains while remaining non-sinks for others. Our theoretical analysis of the training dynamics reveals that these phenomena are driven by a *mutual reinforcement mechanism*. Building on these insights, we propose strategies to mitigate extreme-token phenomena during pretraining, including replacing softmax with ReLU and Adam with SGD. Next, we extend our analysis to pretrained LLMs, including Llama and OLMo, showing that many attention heads exhibit a similar *active-dormant mechanism* as in the BB task, and that the *mutual reinforcement mechanism* also governs the emergence of extreme-token phenomena during LLM pretraining. Our results reveal that many of the static and dynamic properties of extreme-token phenomena predicted by the BB task align with observations in pretrained LLMs.

1 Introduction

Recent analyses of transformer-based open-source large language models (LLMs), such as GPT-2 (Radford et al., 2019), Llama-2 (Touvron et al., 2023), Llama-3 (Dubey et al., 2024), Mixtral (Jiang et al., 2023), and Pythia (Biderman et al., 2023), have revealed three intriguing phenomena:

- **Attention sinks** (Xiao et al., 2023): In many attention heads, the initial token consistently attracts a large portion of the attention weights. Other special tokens, such as the delimiter token, can also draw significant attention weight. These tokens are collectively referred to as *sink tokens*.
- **Value-state drains** (Guo et al., 2024): For the attention heads that exhibit attention sinks, the value states of sink tokens are consistently much smaller than those of other tokens.
- **Residual-state peaks** (Sun et al., 2024): The residual states of sink tokens, excluding those from the first and last layers, exhibit significantly larger norms compared to other tokens.

These phenomena often appear together and consistently occur in various pretrained LLMs, which we collectively refer to as the *extreme-token phenomena*. Figure 1 illustrates these phenomena in Llama-3.1-8B-Base, using a fixed prompt sentence: "⟨s⟩Summer is warm⟨period⟩ Winter is cold⟨period⟩". Here, the first token,

*UC Berkeley. Email: {tianyu_guo, druvpai, jiantao, michael_jordan, songmei}@berkeley.edu.
†Work done at Salesforce AI Research. Email: yubai.pku@gmail.com.

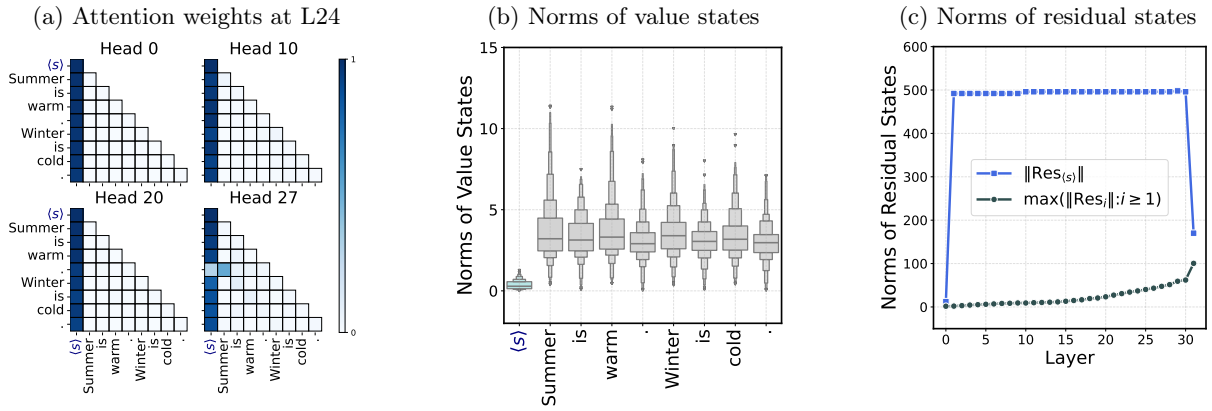


Figure 1: **Extreme-token phenomena in Llama 3.1.** We evaluate the attention weights, value states norm, and residual states norm on the Llama 3.1-8B-Base model, where the input sentence is "⟨s⟩Summer is warm⟨period⟩ Winter is cold⟨period⟩". *Left (a)*: The attention weights across multiple heads at Layer 24. We observe the *attention sink* phenomenon: the ⟨s⟩ token attracts a significant portion of the overall attention weight. *Middle (b)*: The empirical distribution of the norms of value states over all layers and all heads. We exclude 2% of the outlier values to help visualization. We observe the *value-state drain* phenomenon: the value state of the ⟨s⟩ token is much smaller than those of other tokens on average. *Right (c)*: The norm of the residual stream states, measured at the output of each layer. We observe the *residual-state peak* phenomenon: the ⟨s⟩ token's residual states have significantly larger norms than those of other tokens from layers 1 to 30.

⟨s⟩ (the Beginning-of-Sequence token), serves as the sink token. As shown in the figure, the sink token receives disproportionately high attention weights, exhibits significantly smaller value states, and has much larger residual state norms compared to other tokens. It is important to note that the first token does not have to be ⟨s⟩ to act as a sink token; other tokens appearing first in the sequence can also serve this role. Additionally, in models such as Llama-2, a delimiter token can also function as the sink token.

The extreme-token phenomena have posed several challenges for pretrained transformers in downstream tasks. For instance, sink tokens require special treatment during long-context inference (Xiao et al., 2023; Han et al., 2023; Yu et al., 2024; Chen et al., 2024) and model quantization (Dettmers et al., 2022; Liu et al., 2024; Son et al., 2024) to maintain high levels of performance. Additionally, attention sinks have reduced the interpretability of attention maps in vision transformers (Darcet et al., 2023). To address these issues, Sun et al. (2024) and Darcet et al. (2023) propose adding a "special token" to transformers to serve as the sink token, preventing other tokens from becoming sinks. However, even this special token still exhibits extreme-token phenomena. Despite these efforts, no prior work has satisfiably explained the mechanisms behind the extreme-token phenomena. Xiao et al. (2023) proposes a hypothesis for why they occur, suggesting that models tend to dump unnecessary attention values to specific tokens.

This work aims to demystify the extreme-token phenomena in LLMs. We demonstrate that these phenomena arise from an *active-dormant mechanism* in attention heads (cf. Claim 1), coupled with a *mutual-reinforcement mechanism* during pretraining (cf. Claim 2). We support these statements through studies on simplified transformer architectures and tasks, a dynamical theory for these models, and experiments on pretrained LLMs. The structure of the paper and our key contributions are outlined as follows:

1. In Section 2, we train one- to three-layer transformers on a simple task called the *Bigram-Backcopy* (BB) task, which also displays extreme-token phenomena similar to those observed in LLMs. We show that attention sinks and value-state drains are a consequence of the *active-dormant mechanism* (cf. Claim 1). Both theoretically and empirically, we demonstrate that *mutual reinforcement mechanism* (cf. Claim 2) dynamically drives these phenomena: attention sinks and value-state drains reinforce one another, leading to a stable phase where all query tokens generate near identical attention logits for the keys of extreme tokens. Additionally, empirical results reveal that residual-state peaks arise from the interaction between this mutual reinforcement mechanism and the Adam optimization algorithm.

2. In Section 3, we demonstrate the *active-dormant mechanism* in pre-trained LLMs by showing that many

	BB-task Theory	BB-task Experiments	LLM Experiments
$\Delta\text{logit}_{\langle \texttt{s} \rangle}$ log-growth	✓	✓	⋆
$\|\texttt{Val}_{\langle \texttt{s} \rangle}\|$ monotonic decrease	✓	✓	✓
$\|\texttt{Res}_{\langle \texttt{s} \rangle}\|$ linear growth	⋆	✓	✓
$\text{logit}_{\langle \texttt{s} \rangle}$ concentration	✓	✓	✓

Table 1: Consistency of the quantitative properties across the theoretical and empirical results of the Bigram-Backcopy task and empirical results of LLMs. A ✓ denotes a consistent result, while a ⋆ denotes an inconclusive result. The $\text{logit}_{\langle \texttt{s} \rangle}$ denotes logits corresponding to the key of the extreme token and queries of all non-extreme tokens. The $\Delta\text{logit}_{\langle \texttt{s} \rangle} = \text{logit}_{\langle \texttt{s} \rangle} - \mathbb{E}[\text{logit}_{\text{others}}]$ is a progress measure for attention sinks. The $\|\texttt{Val}_{\langle \texttt{s} \rangle}\|$ denotes the value state norm of the extreme token, and $\|\texttt{Res}_{\langle \texttt{s} \rangle}\|$ denotes the residual state norm of the extreme token.

attention heads transition between active and dormant phases based on the input domain. Specifically, we identify an interpretable active-dormant head (Layer 16, Head 25 in Llama 2-7B-Base (Touvron et al., 2023)) that activates on GitHub data but remains dormant on Wikipedia data. Moreover, in examining the dynamics of OLMo-7B-0424 (Groeneveld et al., 2024), we observe the same mutual reinforcement mechanism and stable phase, consistent with those found in the BB task. This demonstrates that the simple BB model captures both the static and dynamic properties of extreme-token phenomena in LLMs and accurately predicts their behavior.

3. Importantly, the quantitative properties of extreme-token dynamics show strong consistency among the theoretical and empirical results of the Bigram-Backcopy task and the empirical performance of OLMo. In particular, we consistently observe the **sink-logits concentration** phenomenon, where the logits corresponding to the key of the extreme token and the queries of all non-extreme tokens ($\text{logit}_{\langle \texttt{s} \rangle}$) are nearly identical. We summarize the aligned results between the theoretical and empirical findings of the Bigram-Backcopy task and the empirical performance of LLMs in Table 1.

4. We propose architectural and optimization modifications to mitigate the extreme-token phenomena. Specifically, we demonstrate that replacing SoftMax with ReLU activations in attention heads eliminates extreme-token phenomena in the BB task, while switching from Adam to SGD removes the residual-state peak phenomenon. We discuss the possibility that similar modifications could mitigate extreme-token phenomena in LLMs.

1.1 Related work

Several studies independently identified the "attention sink" phenomenon in language models and vision transformers, where attention weights were found to be concentrated on a few tokens (Xiao et al., 2023; Darcet et al., 2023; Han et al., 2023; Zhai et al., 2023; Elhage et al., 2023; Dettmers et al., 2022). Recent research has provided more detailed characterizations of this attention pattern and the attention sink phenomenon (Fu, 2024; Sun et al., 2024). Sun et al. (2024) attributed the attention sink to the massive activation of the hidden representations of the corresponding tokens. Both Sun et al. (2024) and Zhai et al. (2023) discussed methods for mitigating the attention sink by modifying the model and training recipes. Additionally, recent studies have leveraged the attention sink phenomenon to develop improved quantization and more efficient inference algorithms (Liu et al., 2024; Chen et al., 2024; Yu et al., 2024; Son et al., 2024).

The dynamics of transformers are studied under various simplifications, including linear attention structures (Zhang et al., 2023; Ahn et al., 2024), reparametrizations (Tian et al., 2023b), NTK (Deora et al., 2023), often in the setting of in-context linear regression (Ahn et al., 2023; Wu et al., 2023; Zhang et al., 2024) and structured sequences (Bietti et al., 2024; Nichani et al., 2024; Tian et al., 2023a). Notably, Zhang et al. (2023); Huang et al. (2023); Kim et al. (2024) demonstrate that a one-layer attention head trained via gradient descent converges to a model that effectively performs in-context regression. Bietti et al. (2024) shows the fast learning of bigram memorization and the slow development of in-context abilities. Tian et al. (2023a) shows the scan and snap dynamics in reparametrized one-layer transformers. Reddy (2023) simplifies the structure of the induction head, showing the connection between the sharp transitions of in-context learning

dynamics and the nested nonlinearities of multi-layer operations.

Mechanistic interpretability is a growing field focused on understanding the internal mechanisms of language models in solving specific tasks (Elhage et al., 2021; Geva et al., 2023; Meng et al., 2022; Nanda et al., 2023; Olsson et al., 2022; Bietti et al., 2024; Wang et al., 2022; Feng and Steinhardt, 2023; Todd et al., 2023). This includes mechanisms like the induction head and function vector for in-context learning (Elhage et al., 2021; Olsson et al., 2022; Todd et al., 2023; Bietti et al., 2024), the binding ID mechanism for binding tasks (Feng and Steinhardt, 2023), association-storage mechanisms for factual identification tasks (Meng et al., 2022), and a complete circuit for indirect object identification tasks (Wang et al., 2022). The task addressed in this paper is closely related to Bietti et al. (2024), who explored synthetic tasks where tokens are generated from either global or context-specific bigram distributions. Several other studies have also employed synthetic tasks to explore neural network mechanisms (Charton, 2022; Liu et al., 2022; Nanda et al., 2023; Allen-Zhu and Li, 2023; Zhu and Li, 2023; Guo et al., 2023; Zhang et al., 2022).

We note that Gurnee et al. (2024) proposed Attention Deactivation Neurons, a concept similar to Dormant Attention Heads. Gurnee et al. (2024) hypothesized that when such a head attends to the first token, it indicates that the head is deactivated and has minimal effect.

1.2 Notation

We denote the SoftMax attention layer with a causal mask as `attn`, the MLP layer as `mlp`, and the transformer block as TF. The query, key, value states, and residuals of a token v are represented as $\texttt{Qry}_v$, $\texttt{Key}_v$, $\texttt{Val}_v$, and $\texttt{Res}_v$, respectively, with the specific layer and head indicated in context. The logit_v denotes logits corresponding to the key of the token v and queries of all other tokens. We use $\langle \texttt{s} \rangle$ to refer to the "Beginning-of-Sequence" token. Since the $\langle \texttt{s} \rangle$ token consistently acts as an extreme token in LLMs, we use $\langle \texttt{s} \rangle$ and extreme tokens interchangeably. Throughout the paper, we adopt zero-indexing (i.e., attention head and layer indices begin at 0 instead of 1) for consistency between the code and the text.

2 Extreme-token Phenomena in the Bigram-Backcopy Task

In this section, we analyze simple transformers trained on the Bigram-Backcopy (BB) task, a simple model that exhibits extreme-token phenomena. We demonstrate the *active-dormant mechanism* (cf. Claim 1) and *mutual reinforcement mechanism* (cf. Claim 2) within the BB task and provide predictions for the behavior of sink tokens, which will be validated through LLM experiments in the following section.

The Bigram-Backcopy task is a data-generation model that consists of two sub-tasks: *Bigram-transition* and *Backcopy*. In this model, each sequence begins with the $\langle \texttt{s} \rangle$ token, followed by tokens sampled according to a pre-determined bigram transition probability P (in other words, a Markov chain). When specific trigger tokens are encountered, instead of sampling according to the transition P, the preceding token is copied to the next position. An illustration of the Bigram-Backcopy task is provided in Figure 2a. Following Bietti et al. (2024), we select the transition P and the vocabulary $\mathcal{V}$ with $|\mathcal{V}| = V = 64$ based on the estimated character-level bigram distribution from the *tiny Shakespeare* dataset. In all experiments, the set of trigger tokens, $\mathcal{T}$, is fixed and consists of the $|\mathcal{T}| = 3$ most frequent tokens from the unigram distribution. Consequently, the non-trigger token set, $\mathcal{V} \setminus \mathcal{T}$, comprises 61 tokens.

2.1 One-layer transformer exhibits attention sinks and value-state drains

On the Bigram-Backcopy task, we pre-train a standard one-layer transformer with a single SoftMax `attn` head and one `mlp` layer. Unless otherwise specified, the model is trained using Adam for $10,000$ steps, achieving near-optimal prediction accuracy. Detailed training procedures are provided in Appendix C.1. Figure 2b shows that the trained transformer exhibits the attention sink phenomenon, where the $\langle \texttt{s} \rangle$ token captures a significant proportion of the attention weights. More importantly, the attention weights display interpretable patterns: all non-trigger tokens exhibit attention sinks, while the attention for trigger tokens is concentrated on their preceding positions. Additionally, Figure 2c reveals a value-state drain phenomenon similar to that

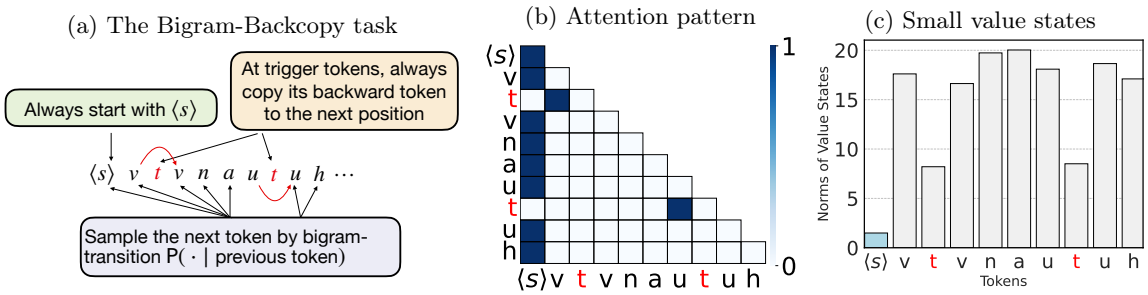


Figure 2: **Experiments on the Bigram-Backcopy task.** *Left (a)*: The data generation procedure for the Bigram-Backcopy task. Here we fix 't', 'e', and the space character (' ') as trigger tokens. The BB task samples bigram transitions for non-trigger tokens and backcopies for trigger tokens. *Middle (b)*: The attention map of a given prompt. Trigger tokens are marked in red. The attention head at non-trigger tokens is dormant and displays attention sinks. *Right (c)*: The value state norms for the prompt. The ⟨s⟩ token has the smallest norm.

observed in LLMs, suggesting that, for non-trigger tokens, the `attn` head contributes minimal value to the residual stream. We provide additional attention patterns on different input sequences in Section C.2.

The active-dormant mechanism of the attention head. Inspired by the interpretable attention weight patterns observed, we propose the *active-dormant mechanism*. For any given token, an attention head is considered *active* if it makes a significant contribution to the residual state, and *dormant* if its contribution is minimal. As illustrated in Figure 2b, when trained on the BB task, the attention head is active for trigger tokens and dormant for non-trigger tokens.

Figure 3a demonstrates that the `mlp` layer is responsible for the Bigram task whereas the `attn` head takes care of the Backcopy task. When the `mlp` layer is zeroed out, the backcopy loss remains significantly better than a random guess, but the bigram loss degrades to near-random levels. Conversely, when the `attn` layer is zeroed out, the backcopy loss becomes worse than a random guess, while the bigram loss remains unaffected. This indicates that on trigger tokens, the `attn` head is active and handles the backcopy task, whereas on non-trigger tokens, the `attn` head is dormant, allowing the `mlp` layer to handle the Bigram task. We summarize the active-dormant mechanism of the `attn` head in Claim 1.

Claim 1 (Active-dormant mechanism)**.** *Attention heads of pre-trained models are often governed by the active-dormant mechanism, exhibiting two phases:*

(1) ***Dormant phase**: On non-trigger tokens, the* `attn` *head assigns dominant weights to the* ⟨s⟩ *token, adding minimal value to the residual stream and having little impact on the model's output.*

(2) ***Active phase**: On trigger tokens, the* `attn` *head assigns dominant attention weights to relevant context tokens, adding substantial value to the residual stream and significantly impacting the model's output.*

The growth of attention logits on the ⟨s⟩ token and the decrease in its value state norms. Figure 3b illustrates the training dynamics of excess risks, attention weights, attention logits (for each token v_n at position n in the prompt, we compute $\Delta\text{logit}_{\langle s\rangle} = \texttt{mean}_n[\langle \texttt{Qry}_{v_n}, \texttt{Key}_{\langle s\rangle}\rangle - \texttt{mean}_i(\langle \texttt{Qry}_{v_n}, \texttt{Key}_{v_i}\rangle)]$, which serves as a progress measure for attention sinks), and value state norms for the ⟨s⟩ token. All values are rescaled to the 0 to 1 range to highlight trends rather than absolute values. Both the Bigram and Backcopy excess risks decrease to nearly zero within the first 1000 steps. As the Backcopy risk decreases, the attention weights on the ⟨s⟩ token begin to increase, suggesting a connection between the formation of attention sinks and the backcopy function in the attention heads. After the first 1000 steps, although both Bigram and Backcopy excess risks have nearly reached zero, the attention logits and weights on the ⟨s⟩ token continue to increase, while the value state norm of the ⟨s⟩ token continues to decrease. While this is an intriguing phenomenon, our next goal is to understand why the attention logits and value state norms continue to evolve toward extreme values.

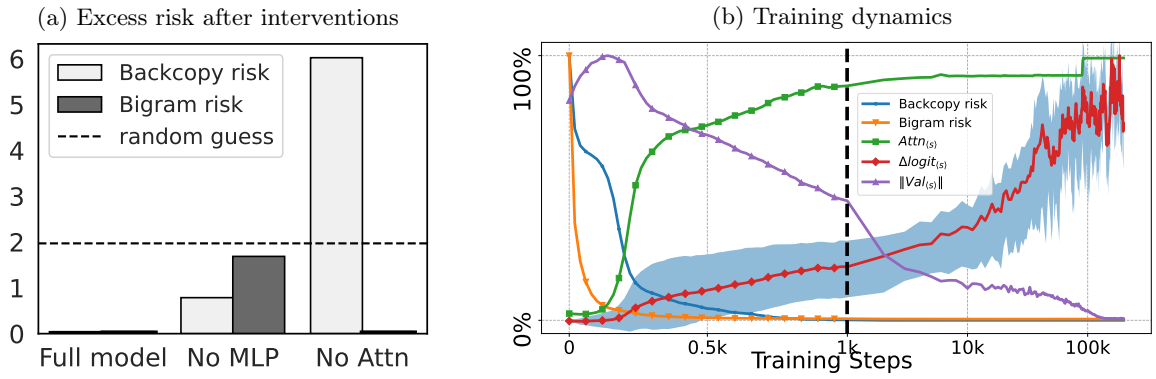


Figure 3: **Interventions and dynamics of one-layer transformer on the Bigram-Backcopy task.** *Left (a)*: Excess risks for a one-layer model trained on the Bigram-Backcopy (BB) task under various interventions. *Right (b)*: The excess risks, attention weights, attention logits, and value state norms for the $\langle \mathtt{s} \rangle$ token throughout the training dynamics. Each curve is rescaled to fall within a 0 to 1 range. On the right side of *(b)*, the horizontal axis is logarithmically scaled. The $\Delta\text{logit}_{\langle \mathtt{s} \rangle}$ curve represents the mean of attention logits from all given non-trigger query tokens v on the $\langle \mathtt{s} \rangle$ token, normalized by the mean of attention logits for other tokens. The shaded area represents the 90% uncertainty interval on the distribution over all non-trigger tokens.

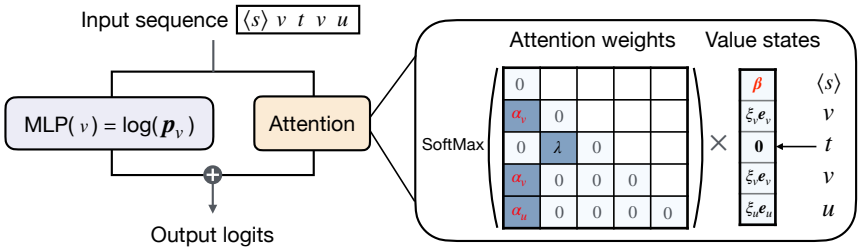


Figure 4: **Simplified transformer architecture.** The output logits are computed by summing the contributions from both the `mlp` layer and the `attn` head. The predicted probabilities are obtained by applying the SoftMax function to these output logits. The `mlp` layer is assumed to provide the Markov transition probabilities for non-trigger tokens, while the `attn` head is parameterized by attention logits and value states, as described in Eq. (1). Additionally, the trainable variables, denoted by $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{R}^V \times \mathbb{R}^V$, represent the attention logits and value states of the $\langle \mathtt{s} \rangle$ token.

2.2 Analysis of a minimally-sufficient transformer architecture

In this section, we analyze the training dynamics of the transformer on the BB task, focusing on a simplified architecture that retains the attention sinks and value-state-drains phenomena. We analyze the regime when the Bigram transition probility is already learned, and the Backcopy task is partially learned (steps 200 to 100k in Figure 3b). We focus on the dynamics of the attention logits and value states.

Let $\mathcal{V}$ (of size V) denote the set of all tokens except the $\langle \mathtt{s} \rangle$ token, and $\mathcal{T}$ denote the set of all trigger tokens. Given any $v \in \mathcal{V}$, we denote $p_{vk} = \mathsf{P}(k|v)$ to be the next token Markov transition probability, and $\boldsymbol{p}_v = [p_{v1}, \ldots, p_{vV}]$ be the row vector in the simplex. We assume that the tokens are embedded into V-dimensional space using one-hot encoding, and for notation simplicity, we abuse v to stand for its one-hot encoding vector $\boldsymbol{e}_v \in \mathbb{R}^V$ which is a row vector. The predicted probability of the $n+1$ token is given by $\mathsf{SoftMax}(\mathrm{TF}([\langle \mathtt{s} \rangle; v_{1:n-1}; v])_n)$, where the transformer architecture is given by $\mathrm{TF}(\cdot) = \mathtt{attn}(\cdot) + \mathtt{mlp}(\cdot)$. Here $\mathtt{attn}(\cdot) = \mathsf{SoftMax}(\mathsf{mask}(\mathtt{Qry}(\cdot)\mathtt{Key}(\cdot)^\top))\mathtt{Val}(\cdot)$ and $(\mathtt{Qry}, \mathtt{Key}, \mathtt{Val})$ are linear maps from $\mathbb{R}^V \to \mathbb{R}^V$. Since the `mlp` layer handles the Bigram task, we assume that `mlp` outputs the Markov transition probabilities $\boldsymbol{p}_v$ on non-trigger tokens v and zero on trigger tokens. For the `attn` head, we assume the attention logits on the $\langle \mathtt{s} \rangle$ key-token are $(\alpha_{v_1}; \ldots; \alpha_{v_n})$, the attention logits on any trigger query-token are $(0, \ldots, \lambda, 0)$ where the second-to-last coordinate is λ, and all other logits are zero. Assume that the value state of $\langle \mathtt{s} \rangle$ is $\boldsymbol{\beta} \in \mathbb{R}^V$, and the value state of each non-trigger token v is a one-hot encoding vector $\boldsymbol{e}_v$ multiplied by $\xi_v \geq 0$. Fig-

ure 4 illustrates this simplified transformer architecture. These assumptions are summarized in the following equations:

$$\begin{aligned}
&\texttt{mlp}(v) = \log \boldsymbol{p}_v \cdot 1\{v \notin \mathcal{T}\} \quad \text{for } v \in \mathcal{V}, \\
&\langle \texttt{Qry}(v), \texttt{Key}(\langle \texttt{s} \rangle) \rangle = \alpha_v \cdot 1\{v \notin \mathcal{T}\} \quad \text{for } v \in \mathcal{V}, \\
&\langle \texttt{Qry}(v), \texttt{Key}(v') \rangle = \lambda \cdot 1\{v \in \mathcal{T}, v' \text{ is the former token of } v\} \quad \text{for } v, v' \in \mathcal{V}, \\
&\texttt{Val}(v) = \xi_v \boldsymbol{e}_v \quad \text{with } \xi_v = 0 \text{ for } v \in \mathcal{T}, \text{ and } \xi_v \geq 0 \text{ for } v \in \mathcal{V} \setminus \mathcal{T}.
\end{aligned} \tag{1}$$

Theorem 1 demonstrates the existence of a transformer architecture of this kind that can generates the ground-truth transitions of the BB model. We provide the proof in Appendix A.1.

Theorem 1. *For any parameters $(\boldsymbol{\alpha} \in \mathbb{R}^V, \boldsymbol{\beta} \in \mathbb{R}^V, \boldsymbol{\xi} \in \mathbb{R}^V, \lambda \in \mathbb{R})$, there exists a one-layer transformer $(\texttt{mlp}, \texttt{Qry}, \texttt{Key}, \texttt{Val})$ such that Eq. (1) holds. This transformer generates the ground-truth transitions of the BB model when $\min_{v \in \mathcal{V}} \alpha_v \to \infty$, $\min_{v \in \mathcal{V}} \xi_v \to \infty$, $\lambda \to \infty$, and $\boldsymbol{\beta} = 0$.*

Throughout, we adopt Eq. (1) as our assumption. We further define $W_k = \sum_{i=1}^n 1\{v_i = k\}$, $\boldsymbol{W} = (W_1, \ldots, W_V)$, and $W = \sum_{k \in \mathcal{V}} W_k = n$. Then for a non-trigger token v, the output of attention layer with input sequence $[\langle \texttt{s} \rangle; v_{1:n-1}; v]$ is given by (denoting $\xi_k = 0$ for $k \in \mathcal{T}$)

$$\mathrm{TF}([\langle \texttt{s} \rangle; v_{1:n-1}; v])_n = \log \boldsymbol{p}_v + \frac{e^{\alpha_v}}{e^{\alpha_v} + W} \boldsymbol{\beta} + \sum_{k=1}^V \frac{W_k \xi_k}{e^{\alpha_v} + W} \cdot \boldsymbol{e}_k.$$

Therefore, on the non-trigger token v, the cross-entropy loss between the true Markov transition $\boldsymbol{p}_v$ and predicted transition $\mathsf{SoftMax}(\mathrm{TF}([v_{1:n-1}; v])_n)$ is given by

$$\mathsf{loss}_v(\alpha_v, \boldsymbol{\beta}) = \sum_{k=1}^V p_{vk} \Big\{ \log \Big[\sum_{i=1}^V p_{vi} \exp\Big(\frac{e^{\alpha_v} \beta_i + W_i \xi_i}{e^{\alpha_v} + W} \Big) \Big] - \frac{e^{\alpha_v} \beta_k + W_k \xi_k}{e^{\alpha_v} + W} - \log p_{vk} \Big\}. \tag{2}$$

For simplicity, we neglect the loss on trigger tokens and assume that $(\{W_i\}_{i \in [V]}, W)$ are fixed across different positions in the input sequences,[1] and consider the total loss as the average of the losses on each non-trigger token, weighted by its proportion in the stable distribution $\{\pi_v\}_{v \in \mathcal{V}}$, given by

$$\mathsf{loss}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \textstyle\sum_{v \in \mathcal{V} \setminus \mathcal{T}} \pi_v \cdot \mathsf{loss}_v(\alpha_v, \boldsymbol{\beta}). \tag{3}$$

We assume $\boldsymbol{\xi}$ and λ are fixed, and that $\boldsymbol{\alpha}$ (the attention logits of the $\langle \texttt{s} \rangle$ token) and $\boldsymbol{\beta}$ (the value states norm of the $\langle \texttt{s} \rangle$ token) are trainable variables, as we are interested in the dynamics of the attention logits and value state norm for the $\langle \texttt{s} \rangle$ token. The theorem below illustrates the logarithmic growth of attention logits $\boldsymbol{\alpha}$, the shrinkage of value states $\boldsymbol{\beta}$, and the stable phase of these two variables.

Theorem 2. *Consider the gradient flow of the loss function $\mathsf{loss}(\boldsymbol{\alpha}, \boldsymbol{\beta})$. Assume $\xi_v \geq 0$ for any v, and $\{W_i \cdot \xi_i\}_{i \in \mathcal{V}}$ are not all equal.*

(a) *(Attention logits grow logarithmically, reinforced by small value states) Fix $\boldsymbol{\beta} = \beta \cdot \mathbf{1}$ for a constant β, and consider the gradient flow over $\boldsymbol{\alpha}$. With any initial value $\boldsymbol{\alpha}(0)$, there exists $\boldsymbol{r}(t)$ with norm uniformly bounded in time, such that*

$$\boldsymbol{\alpha}(t) = \tfrac{1}{2} \log t \cdot \mathbf{1} + \boldsymbol{r}(t). \tag{4}$$

(b) *(Value state shrinks to a small constant vector, reinforced by large attention logits) Fix $\boldsymbol{\alpha} = \alpha \cdot \mathbf{1}$ for a constant α, define $\overline{\beta}(0) = V^{-1}[\sum_v \beta_v(0)]$ and $\overline{B} = V^{-1}[\sum_v W_v \xi_v]$. Consider the gradient flow over $\boldsymbol{\beta}$. As $t \to \infty$, we have*

$$\boldsymbol{\beta}(t) \to \boldsymbol{\beta}^\star = [\overline{\beta}(0) + e^{-\alpha} \overline{B}] \cdot \mathbf{1} - e^{-\alpha} \cdot \boldsymbol{W} \circ \boldsymbol{\xi}. \tag{5}$$

[1] We note that Reddy (2023) makes similar simplification in analyzing induction heads.

(c) *(Stable phase: Sink-logits concentration) Consider the gradient flow over the variables* $(\boldsymbol{\alpha}, \boldsymbol{\beta})$. *Any vector of the following form*

$$\boldsymbol{\alpha} = \alpha \cdot \mathbf{1}, \quad \boldsymbol{\beta} = c \cdot \mathbf{1} - e^{-\alpha} \cdot \boldsymbol{W} \circ \boldsymbol{\xi}, \quad \alpha, c \in \mathbb{R} \tag{6}$$

is a stationary point. These are all global minimizers of $\mathsf{loss}(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

The proof of Theorem 2 is provided in Appendices A.2, A.3, and A.4. We offer two key remarks: (1) As $\alpha_v \to \infty$, a Taylor expansion of the gradient $\partial \mathsf{loss}/\partial \alpha_v$ suggests that $\mathrm{d}\alpha_v/\mathrm{d}t \propto \exp(-2\alpha_v)$, which leads to the logarithmic growth of α_v. Similar logarithmic growth has been reported in the literature under different setups (Tian et al., 2023a; Zhu et al., 2024); (2) The stable phase described in Theorem 2(c) seems to imply that the system can remain stable without attention sinks, as it does not require α to be large. However, in practice, models trained on the BB task tend to converge to a stable phase where α is relatively large.

The formation of attention sinks and value-state drains. Below, we explain how Theorem 2 reveals the *mutual reinforcement mechanism* behind the formation of attention sinks and value-state drains.

(a) When the value states of the $\langle \mathtt{s} \rangle$ token are small and constant, $\boldsymbol{\beta} = \beta \cdot \mathbf{1}$, Theorem 2(a) shows that the attention logits on the $\langle \mathtt{s} \rangle$ token $\boldsymbol{\alpha}(t) \approx \alpha(t)\mathbf{1}$ for $\alpha(t) = (1/2)\log t$, grow logarithmically. This demonstrates that the presence of a small constant value state ($\boldsymbol{\beta} = \beta \cdot \mathbf{1}$) reinforces the formation of attention sinks ($\boldsymbol{\alpha}(t) \approx \alpha(t) \cdot \mathbf{1}$ for $\alpha(t)$ increases logarithmically).

(b) When the attention logits of the $\langle \mathtt{s} \rangle$ token are large and constant, $\boldsymbol{\alpha} = \alpha \cdot \mathbf{1}$ for $\alpha \to \infty$, Theorem 2(b) shows that the value states of the $\langle \mathtt{s} \rangle$ token $\boldsymbol{\beta}(t) \to \overline{\beta}(0) \cdot \mathbf{1}$. Starting with a random Gaussian initialization for $\boldsymbol{\beta}(0)$, we have $\|\boldsymbol{\beta}(t)\|_2 \approx \|\overline{\beta}(0) \cdot \mathbf{1}\|_2 \approx \|\boldsymbol{\beta}(0)\|_2/\sqrt{V}$, where V is the vocabulary size, typically large. This indicates that attention sinks ($\boldsymbol{\alpha} = \alpha \cdot \mathbf{1}$ for large α) reinforces the formation of value-state drains ($\boldsymbol{\beta}(t) \to \beta \cdot \mathbf{1}$ for small β).

(c) In the later stages of the dynamics, both the attention logits and value states of the $\langle \mathtt{s} \rangle$ token stabilize, as described in 2(c). The attention logits remain constant at $\boldsymbol{\alpha} = \alpha \cdot \mathbf{1}$ with large α, while the value states become small, $\boldsymbol{\beta} = [\overline{\beta}(0) + e^{-\alpha}\overline{B}] \cdot \mathbf{1} - e^{-\alpha} \cdot \boldsymbol{W} \circ \boldsymbol{\xi}$.

Based on these theoretical insights, we summarize the mutual reinforcement mechanism in plain language:

Claim 2 (Mutual reinforcement mechanism)**.** *For any attention head given a specific prompt, if the model can accurately predict the next token without using the attention head, but adding any value state from previous tokens—except for certain special tokens—worsens the prediction, the attention head will become dormant, forming an attention sink at those special tokens. Dynamically, this arises from a mutual reinforcement between attention sinks and value-state drains:*

(a) *The SoftMax mechanism shifts attention weights towards tokens that exhibit value-state drains, reinforcing these tokens as attention sinks.*

(b) *Attention sinks on these extreme tokens further suppress their value states, reinforcing their role as value-state drains.*

(c) *The mutual reinforcement stabilizes when all non-trigger tokens have large, nearly identical attention logits on the extreme token.*

Due to the causal mask, the training dynamics favor the $\langle \mathtt{s} \rangle$ *token as the extreme token.*

Experimental verification of the quantitative prediction. Revisiting Figure 3b, which illustrates the dynamics of a single-layer transformer model trained with Adam on the BB task, we observe that $\Delta\mathrm{logit}_{\langle \mathtt{s} \rangle}$ exhibits growth rates consistent with Theorem 2. In this context, $\Delta\mathrm{logit}_{\langle \mathtt{s} \rangle}$ corresponds to α, as all other attention logits are assumed to be zero under the assumptions of Theorem 2. When plotted on a logarithmic scale, the $\Delta\mathrm{logit}_{\langle \mathtt{s} \rangle}$ curve grows approximately linearly between 1,000 and 10,000 steps, then accelerates before stabilizing around 100,000 steps. Meanwhile, the norm of the value state $\|\mathtt{Val}_{\langle \mathtt{s} \rangle}\|_2$ decreases monotonically. The simultaneous increase in attention weights and decrease in value-state norms demonstrate the mutual reinforcement mechanism during the training process.

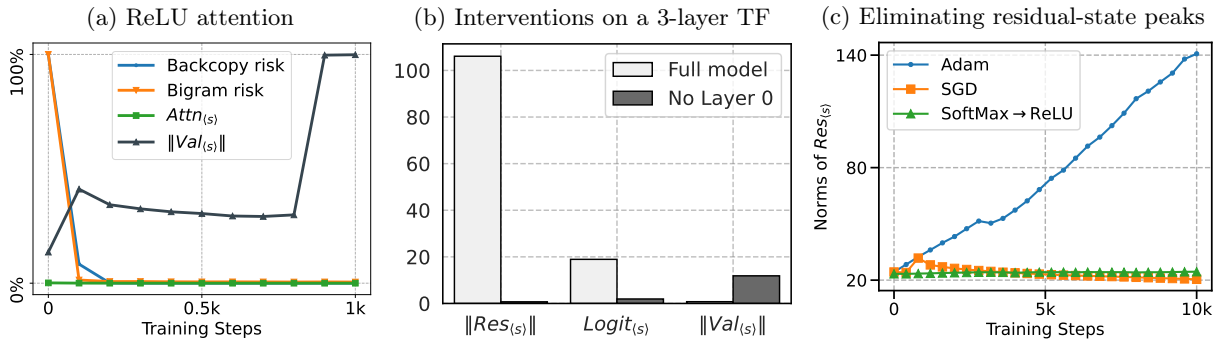


Figure 5: *Left (a)*: The training dynamics of the single-layer ReLU attention transformer on the BB task. *Middle (b)*: The intervention results on the `attn+mlp+attn+mlp+mlp` architecture. The attention sink and value-state peak of the middle `attn` layer disappear after zeroing out `attn+mlp` of layer 0. *Right (c)*: The evolution of massive norms in a three-layer transformer trained with Adam, SGD, and using a ReLU attention transformer. Notably, only the three-layer model with Softax attention trained using Adam results in the formation of residual-state peaks.

To further verify that Theorem 2 captures the dynamics of the original model, we further constructed a simplified model that aligns with Eq. (1) , and trained the parameters ($\boldsymbol{\alpha} \in \mathbb{R}^V, \boldsymbol{\beta} \in \mathbb{R}^V, \boldsymbol{\xi} \in \mathbb{R}^V, \lambda \in \mathbb{R}$) using Adam. The resulting training curves resemble those of a one-layer transformer, also displaying the mutual reinforcement mechanism. We provide a full description in Appendix C.3.

Generality of the theoretical prediction. Although Theorem 2 focuses on a specific BB task with a simplified architecture and loss function, the underlying principles are broadly applicable to more general settings. In particular, we expect that the formation of extreme tokens in LLMs follows a similar mutual reinforcement mechanism. Indeed, Theorem 2 is essentially based on the following two key assumptions: (1) even with a specific attention head `attn` zeroed out, the LLM can still accurately predict the next token, implying that the attention head is better off dormant; and (2) for the attention head `attn`, value states of previous tokens—except for certain special tokens—remain relevant for specific tasks and therefore do not vanish. Under these assumptions, we anticipate the formation of attention sinks and value-state drains for the attention head `attn` and such special tokens. In Section 3, we explore how these phenomena are formed during the training dynamics of LLMs, finding that the empirical results align with the theory.

Replacing SoftMax by ReLU attention removes attention sinks and value-state drains. As a consequence of our theory, we predict that training using ReLU attention in place of SoftMax attention will prevent the mutual reinforcement mechanism. Without SoftMax, the training dynamics no longer push the attention weights toward the ⟨s⟩ token, which remains zero throughout training. In the absence of attention sinks, the dynamics no longer push down the value state norm, and the mutual reinforcement mechanism breaks. Figure 5a presents the training dynamics on the BB task using ReLU instead of SoftMax attention, showing that both the Bigram and Backcopy risk converge to the Bayes risk after 200 training steps, but the attention logits of ⟨s⟩ do not increase, and the value state does not shrink, confirming our prediction.

2.3 The emergence of residual-state peaks

In this section, we experimentally investigate the residual-state peaks phenomenon. We observe that no residual-state peaks occur in the single-layer transformer trained on the BB task. To explore this further, we train slightly deeper transformers on the BB task and track the residual state norm after layer 0. We observe that two-layer models do not exhibit residual-state peaks, while models with three or more layers do. Additional experimental results are provided in Appendix B.1 and B.2.

Massive residual state at layer 0 output induces attention sinks and value-state drains in the middle layer. We perform intervention experiments in the "`attn+mlp+attn+mlp+mlp`" model, by

analyzing how the model's behavior changes after zeroing out layer 0 (the first "attn+mlp" block). Before and after zeroing, we compute the difference in $\|\texttt{Res}_{\langle \texttt{s} \rangle}\|$ and $\text{Mean}_v[\|\texttt{Res}_v\|]$ at the layer 0 output, and compute $\text{logit}_{\langle \texttt{s} \rangle}$ and $\|\texttt{Val}_{\langle \texttt{s} \rangle}\|$ in the middle layer. After zeroing out, the residual state norm becomes non-massive, and attention logits and the value state norm return to a normal level. This confirms that the residual-state peak contributes to the attention sink and value-state-drain phenomena in the middle layer of pre-trained transformers.

Linear growth of residual-state norm with Adam training. Figure 5c shows the residual-state norms of the $\langle \texttt{s} \rangle$ token at the layer 0 output of three-layer transformers during pre-training on the BB task. The results indicate that training the transformer with Adam leads to a linear increase in residual norms.

Switching from Adam to SGD and switching from SoftMax to ReLU attention eliminates the residual-state peaks. Figure 5c also illustrates the dynamics of residual-state norms in other training setups. When switching the training algorithm from Adam to SGD, attention sinks remain, but residual-state peaks disappear. Similarly, switching to ReLU attention, which lacks the mutual reinforcement mechanism, also eliminates residual-state peaks. These findings highlight the dependence of residual-state peaks on SoftMax attention and the Adam optimization algorithm. We propose a potential explanation of this phenomenon in Appendix B.3.

3 Extreme-token Phenomena in pretrained LLMs

In this section, we investigate extreme-token phenomena in open-source pretrained LLMs. In Section 3.1, we analyze the static behavior of these phenomena in Llama 2-7B-Base (Touvron et al., 2023), confirming the existence of the *active-dormant mechanism* in LLMs. Notably, we identify a specific head that is active on GitHub samples but dormant on Wikipedia samples. In Section 3.2, we examine the dynamic behavior of extreme-token phenomena during the pretraining of OLMo-7B (Groeneveld et al., 2024). We show that the attention logits, value state norms, and residual state norms of the sink token(s) in OLMo reflect behavior similar to that of the simpler BB model. Specifically, the simultaneous formation of attention sinks and value-state drains gives evidence for the *mutual reinforcement mechanism*.

3.1 Active-dormant mechanism in LLMs

Our study of the BB model leads to the following prediction with respect to the extreme-token phenomena, which we hypothesize also applies to LLMs:

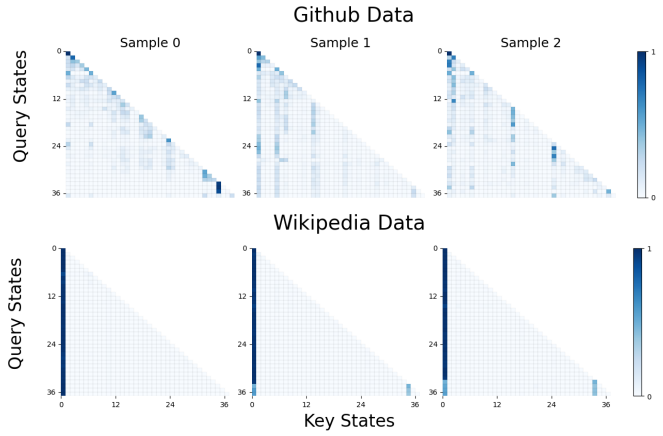
Attention heads are controlled by an active-dormant mechanism (cf. Claim 1). The presence of attention sinks and value-state drains indicates that an attention head is in a dormant phase.

This hypothesis suggests that in LLMs, whether an attention head becomes a sink depends on the context. Specifically, the attention head may become entirely irrelevant for selecting the next tokens in certain contexts or tasks, but not in others. When this irrelevance occurs, the attention head transitions into an attention sink. This hypothesis was confirmed in small transformers and the BB task, as demonstrated in Section 2.

Accordingly, we aim to identify instances of attention heads in pretrained LLMs that exhibit this active-dormant behavior, i.e., heads that are dormant in some domains but active in others. In Figure 6, we display a particular attention head—Layer 16 Head 25 (L16H25) of Llama 2-7B-Base (Touvron et al., 2023)—which demonstrates a clear active-dormant distinction across two distinct contexts (e.g., tokens from the GitHub subset versus the Wikipedia subset of RedPajama (Computer, 2023)). While many attention heads show similar context-dependent behavior (see Appendix D), we focus on this one because the conditions for its activation are straightforward and interpretable, whereas other heads may have more nuanced criteria.

Figure 6a shows the attention maps of L16H25 on samples from both the GitHub and Wikipedia subsets of RedPajama. It demonstrates that L16H26 is *dormant* (i.e., an attention sink) on samples from Wikipedia, which resemble prose, and *active* (i.e., not an attention sink) on samples from GitHub, which resemble code. Additionally, Figure 6b compares the loss difference when L16H25 is zeroed out for prompts from both

(a) Attention weights for GitHub/Wikipedia data



(b) Zero-out-head intervention outcomes

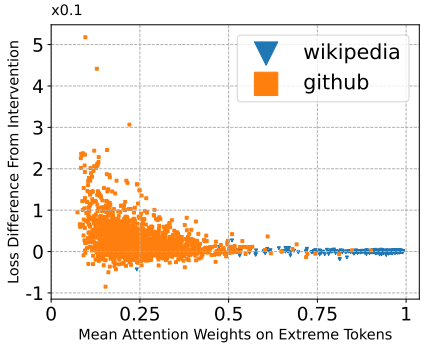


Figure 6: **Active-dormant mechanism of Layer 16 Head 25 (L16H25) of Llama 2-7B-Base.** We observe that L16H25 is active on GitHub data and dormant on Wikipedia data, both sourced from RedPajama-1T (Computer, 2023). *Left (a)*: Attention weights of L16H25, prompted by three randomly selected samples from each domain. *Right (b)*: Results of an intervention study showing the change in cross-entropy loss when the output of L16H25 (specifically, its value states) is set to zero across sequences in both domains. The findings indicate that the model's performance for GitHub data, measured by cross-entropy loss, strongly relies on the output of this attention head.

domains. The results show that zeroing out this head significantly decreases model performance on GitHub sequences, while having minimal impact on Wikipedia sequences. This observation also confirms the head behaves as dormant in some contexts and active in others—in some contexts, removing this head has no effect on model performance, while in others, its removal causes significant performance drops.

3.2 Extreme-token phenomena along training dynamics of LLMs

Our study of the BB model leads to the following prediction about the dynamical behavior of the extreme-token phenomena, which we hypothesize also applies to LLMs:

Attention heads undergo an attention-increasing and value-state-shrinking phase driven by the mutual reinforcement mechanism (cf. Claim 2). This is followed by a stable phase, where all non-trigger tokens have large, nearly identical attention logits on the extreme token. Simultaneously, the residual state norms of the extreme tokens increase linearly during pretraining.

We confirm these predictions below. To observe the training dynamics of a large-scale LLM, we use the setup of OLMo-7B-0424 (Groeneveld et al., 2024) (henceforth just referred to as OLMo), which provides open-sourced weights at various stages of their training.[2] For our analysis, we inspect OLMo at multiple checkpoints: every 500 steps for the first 10,000 steps, then at 25,000 steps, 50,000 steps, and every 50,000 steps up to 449,000 steps (approximately the end of their training).[3] The input we use for this analysis is again "Summer is warm⟨`period`⟩ Winter is cold⟨`period`⟩"[4] In this prompt, token 3, namely "⟨`period`⟩" also becomes a sink token along with token 0. We believe this occurs because the period is not semantically meaningful and is not useful for predicting future tokens (cf. Appendix G.2)

Figure 7 illustrates the dynamics of attention weights, value state norms, and the residual state norms for attention heads in Layer 24 of OLMo. The figure shows that the average attention on extreme tokens (token 0 and token 3) increases rapidly at the beginning of training before stablizing, while the value state norms of these extreme tokens decrease rapidly. Additionally, the residual states of token 0 increase linearly, while

[2]We did not analyze Llama for dynamics, as they do not provide open-source intermediate checkpoints along pretraining.

[3]For the single 150,000-step checkpoint, we observed that its statistics were outliers, which we hypothesize is due to a system failure. We address this by using the average of nearby checkpoints to represent its statistics.

[4]Note that OLMo does not have a ⟨`s`⟩ token, but attention sinks still form in the majority of heads. In particular, the first token always behaves as an attention sink. We discuss this further in Appendix G.2.

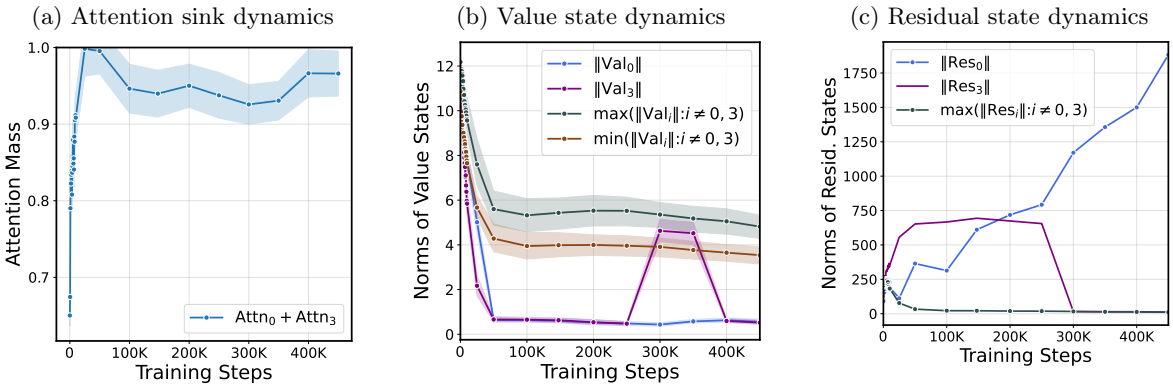


Figure 7: **Attention weights, value state norms, and residual state norms of Layer 24 during the training dynamics of OLMo.** *Left (a)*: The total attention mass on extreme tokens 0 and 3 at Layer 24, averaged across all attention heads. We observe a rapid increase followed by stabilization within the range $[0.9, 1]$ for the rest of training, consistent with our predictions. *Middle (b)*: The value state norms of each token at Layer 24 during training, averaged over all heads. Initially, the value states of all tokens shrink, eventually converging, while the value states of the extreme tokens shrink to significantly lower levels compared to other tokens. Figure *(a)* and *(b)* coincide with the trends in Figure 3b under the BB task. *Right (c)*: The residual state norms of each token at Layer 24 during training. The residual state norm of token 0 increases linearly in magnitude throughout training, matching Figure 5c in the BB task.

those of other tokens converge to a small number. Figure 8 provides a more detailed examination of the attention logits in Layer 24 of OLMo, demonstrating *sink-logits concentration* phenomenon. Specifically, it shows that the sink logits will eventually converge to a stable phase, in which logits corresponding to the key of the sink token and queries of all non-sink tokens are nearly identical. These findings coincide with the dynamical behavior predicted by the BB model, as outlined in Theorem 2(c) and corroborated by the experimental results in Figure 3.

4 Conclusions

In this work, we investigated the *extreme-token phenomena*, specifically *attention sinks*, *value-state drains*, and *residual-state peaks*. We analyzed simple transformers trained on the Bigram-Backcopy (BB) task, both theoretically and empirically, demonstrating that these models exhibit the same extreme-token phenomena observed in large language models (LLMs). Building on the insights from the BB task, we made several detailed predictions about the behavior of extreme-token phenomena in LLMs. In particular, we identified the *active-dormant mechanism* governing attention heads in both the BB model and LLMs, with attention sinks and value-state drains serving as indicators of dormant phase, and a *mutual reinforcement mechanism* that induces these phenomena during pretraining. Using insights from these mechanisms, we applied simple modifications to the model architecture and optimization procedure, effectively mitigating the extreme-token phenomena in the BB model. Overall, our work uncovers the underlying mechanisms of extreme-token phenomena and suggests potential pathways to mitigate these issues during LLM pretraining.

We believe the most compelling direction for future work is to explore whether eliminating the extreme-token phenomena is essential or beneficial for building powerful transformer-based LLMs. While it is possible to mitigate these phenomena through simple modifications to the architecture or training algorithms, it remains unclear whether their elimination significantly improves downstream tasks such as inference and quantization. Given the resource-intensive nature of pretraining large-scale LLMs, we anticipate that pretraining a model at the scale of GPT-2 could both provide valuable insight into this issue and help point the way to architectures that can reduce the pretraining burden.

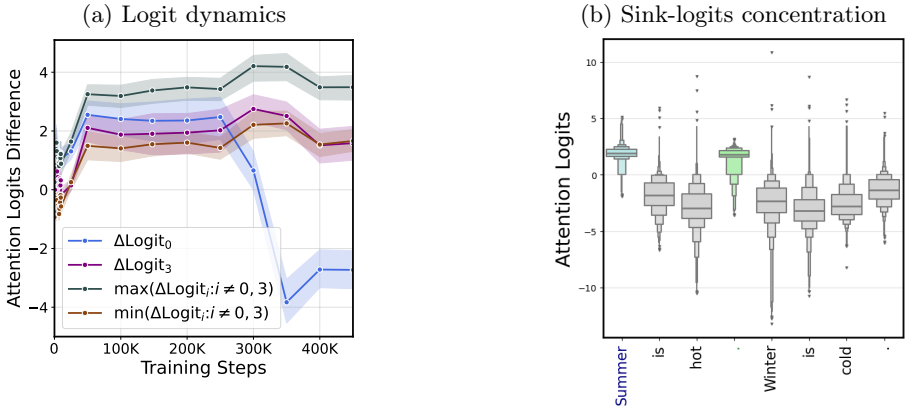


Figure 8: **Attention logits of Layer 24.** *Left (a)*: Attention logits of all tokens' query states against token 0's key state during training. We observe that the logits of all non-extreme tokens' query states against token 0's key state in OLMo's Layer 24 stabilize after an initial period. This behavior aligns with the stable-phase prediction made in the BB model in Theorem 2(c). Note that this prediction does not apply to the logit corresponding to the zeroth query and key token, as its softmax value will be set to 1, making its behavior irrelevant for prediction. The difference in attention logits is computed as $\Delta\text{logit}_i = \langle \texttt{Qry}_i, \texttt{Key}_0 \rangle - \texttt{mean}_j(\langle \texttt{Qry}_i, \texttt{Key}_j \rangle)$. *Right (b)*: Attention logits of the last token's query state against all token's key states for pretrained OLMo. In this experiment, we generate 128 randomly sampled test tokens with IDs from 100 to 50000 in the OLMo tokenizer. We append each token separately to the test phrase "Summer is warm⟨period⟩ Winter is cold⟨period⟩", creating 128 different samples, which we feed to the LLM to examine the model behavior. We plot the distribution of (un-shifted) attention logits $\langle \texttt{Qry}_{\text{test}}, \texttt{Key}_j \rangle$ across all heads at Layer 24 and all test tokens. The distribution of logit_0 and logit_3 have considerably small variance compared with other logits, confirming the sink-logits concentration phenomenon.

Acknowledgements

TG thanks Yaodong Yu, Licong Lin, and Ruiqi Zhang for insightful discussions. YB thanks Caiming Xiong and Huan Wang for the many insightful discussions in the early stages of this work. This project is supported by NSF DMS-2210827, CCF-2315725, CAREER DMS-2339904, ONR N00014-24-S-B001, a UC Berkeley College of Engineering fellowship, an Amazon Research Award, a Google Research Scholar Award, an Okawa Foundation Research Grant, and the European Union (ERC-2022-SYG-OCEAN-101071601).

References

Kwangjun Ahn, Xiang Cheng, Minhak Song, Chulhee Yun, Ali Jadbabaie, and Suvrit Sra. Linear attention is (maybe) all you need (to understand transformer optimization). *arXiv preprint arXiv:2310.01082*, 2023.

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

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 1, context-free grammar. *arXiv preprint arXiv:2305.13673*, 2023.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

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

François Charton. What is my math transformer doing? Three results on interpretability and generalization. *arXiv preprint arXiv:2211.00170*, 2022.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. *arXiv preprint arXiv:2403.06764*, 2024.

Together Computer. RedPajama: An open source recipe to reproduce Llama training dataset, 2023. URL https://github.com/togethercomputer/RedPajama-Data.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

Puneesh Deora, Rouzbeh Ghaderi, Hossein Taheri, and Christos Thrampoulidis. On the optimization and generalization of multi-head attention. *arXiv preprint arXiv:2310.12680*, 2023.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332, 2022.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1:1, 2021.

Nelson Elhage, Robert Lasenby, and Christopher Olah. Privileged bases in the transformer residual stream. *Transformer Circuits Thread*, 2023.

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? *arXiv preprint arXiv:2310.17191*, 2023.

Yao Fu. How do language models put attention weights over long context? *Yao Fu's Notion*, 2024. URL https://yaofu.notion.site/How-Do-Language-Models-put-Attention-Weights-over-Long-Context-10250219d5ce42e8b465087c383a034e?pvs=4.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*, 2023.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

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.

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. Attention score is not all you need for token importance indicator in KV cache reduction: Value also matters. *arXiv preprint arXiv:2406.12335*, 2024.

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal neurons in GPT2 language models. *arXiv preprint arXiv:2401.12181*, 2024.

Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. LM-Infinite: Simple on-the-fly length generalization for large language models. *arXiv preprint arXiv:2308.16137*, 2023.

Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Juno Kim, Tai Nakamaki, and Taiji Suzuki. Transformers are minimax optimal nonparametric in-context learners. *arXiv preprint arXiv:2408.12186*, 2024.

Ruikang Liu, Haoli Bai, Haokun Lin, Yuening Li, Han Gao, Zhengzhuo Xu, Lu Hou, Jun Yao, and Chun Yuan. Intactkv: Improving large language model quantization by keeping pivot tokens intact. *arXiv preprint arXiv:2403.01241*, 2024.

Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *Advances in Neural Information Processing Systems*, 35:34651–34663, 2022.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

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

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

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

Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *The Twelfth International Conference on Learning Representations*, 2023.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024.

Seungwoo Son, Wonpyo Park, Woohyun Han, Kyuyeun Kim, and Jaeho Lee. Prefixing attention sinks can mitigate activation outliers for large language model quantization. *arXiv preprint arXiv:2406.12016*, 2024.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

Yuandong Tian, Yiping Wang, Beidi Chen, and Simon S Du. Scan and Snap: Understanding training dynamics and token composition in 1-layer transformer. *Advances in Neural Information Processing Systems*, 36:71911–71947, 2023a.

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

Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*, volume 47. Cambridge University Press, 2018.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter L Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? *arXiv preprint arXiv:2310.08391*, 2023.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Zhongzhi Yu, Zheng Wang, Yonggan Fu, Huihong Shi, Khalid Shaikh, and Yingyan Celine Lin. Unveiling and harnessing hidden attention sinks: Enhancing large language models without training through attention calibration. *arXiv preprint arXiv:2406.15765*, 2024.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning*, pages 40770–40803. PMLR, 2023.

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

Ruiqi Zhang, Jingfeng Wu, and Peter L Bartlett. In-context learning of a linear transformer block: Benefits of the MLP component and one-step GD initialization. *arXiv preprint arXiv:2402.14951*, 2024.

Yi Zhang, Arturs Backurs, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, and Tal Wagner. Unveiling transformers with LEGO: A synthetic reasoning task. *arXiv preprint arXiv:2206.04301*, 2022.

Hanlin Zhu, Baihe Huang, Shaolun Zhang, Michael Jordan, Jiantao Jiao, Yuandong Tian, and Stuart Russell. Towards a theoretical understanding of the 'reversal curse' via training dynamics. *arXiv preprint arXiv:2405.04669*, 2024.

Zeyuan Allen Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*, 2023.

Contents

A Proofs of Theorem 1 and 2

We provide new notations which are frequently used in the proofs. Define the stable distribution excluding trigger tokens.

$$\tilde{\boldsymbol{\pi}} \in \mathbb{R}^V, \quad \tilde{\pi}_i = \pi_i \mathbf{1}\{i \in \mathcal{V} \setminus \mathcal{T}\}. \tag{7}$$

Define the full bigram transition probability.

$$\mathbf{P} = \begin{pmatrix} p_{11} & \dots & p_{1V} \\ \vdots & \ddots & \vdots \\ p_{V1} & \dots & p_{VV} \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_1^\top \\ \vdots \\ \boldsymbol{p}_V^\top \end{pmatrix}. \tag{8}$$

Given token v, define the predicted probability, which is the logit output passed through the softmax activation

$$\boldsymbol{q}_v = \mathsf{SoftMax}(\mathrm{TF}([\langle \mathsf{s} \rangle; v_{1:n-1}; v])_n). \tag{9}$$

Similarly, define the full output probability matrix.

$$\mathbf{Q} = \begin{pmatrix} q_{11} & \dots & q_{1V} \\ \vdots & \ddots & \vdots \\ q_{V1} & \dots & q_{VV} \end{pmatrix} = \begin{pmatrix} \boldsymbol{q}_1^\top \\ \vdots \\ \boldsymbol{q}_V^\top \end{pmatrix}. \tag{10}$$

Given any vector $\boldsymbol{u} = [u_1; \dots; u_d]$, define the corresponding diagonal matrix as

$$\mathrm{diag}(\boldsymbol{u}) = \begin{pmatrix} u_1 & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & u_d \end{pmatrix}.$$

Define

$$\mathbf{G}_v^{\mathbf{P}} = \mathrm{diag}(\boldsymbol{p}_v) - \boldsymbol{p}_v \boldsymbol{p}_v^\top, \quad \mathbf{G}_v^{\mathbf{Q}} = \mathrm{diag}(\boldsymbol{q}_v) - \boldsymbol{q}_v \boldsymbol{q}_v^\top. \tag{11}$$

Denote

$$\boldsymbol{z} = W \cdot \boldsymbol{\beta} - \boldsymbol{W} \circ \boldsymbol{\xi}. \tag{12}$$

We first present a technical lemma.

Lemma A.1. *The matrices $\mathbf{G}_v^{\mathbf{P}}$ and $\mathbf{G}_v^{\mathbf{Q}}$ are positive semi-definite for any v.*

Proof. Since $\sum_{k=1}^V p_{vk} = 1$ and $\sum_{k=1}^V q_{vk} = 1$ for any v, we have that

$$(\mathbf{G}_v^{\mathbf{P}})_{ii} = p_i - p_i^2 = p_i(\sum_{k \neq i} p_k) \geq \sum_{k \neq i} |(\mathbf{G}_v^{\mathbf{P}})_{ik}|,$$
$$(\mathbf{G}_v^{\mathbf{Q}})_{ii} = q_i - q_i^2 = q_i(\sum_{k \neq i} q_k) \geq \sum_{k \neq i} |(\mathbf{G}_v^{\mathbf{Q}})_{ik}|.$$

This shows that both $\mathbf{G}_v^{\mathbf{P}}$ and $\mathbf{G}_v^{\mathbf{Q}}$ are diagonally dominant matrices. By Corollary 6.2.27 in Horn and Johnson (2012), they are positive semi-definite. □

A.1 Proof of Theorem 1

We denote the hidden dimension as d and the sequence length as N. Suppose that the token v at position i is encoded as $\mathtt{ebd}_i(v)$. We begin with the assumption regarding the transformer's positional embedding:

Assumption A. *The vectors $\{\mathtt{ebd}_i(v)\}_{i \in [N], v \in \mathcal{V}}$ are linearly independent. There exists an orthogonal basis $\{\mathbf{u}_v\}_{v \in \mathcal{V}}$ that is in the null space of the span of $\{\mathtt{ebd}_i(v)\}_{i \in [N], v \in \mathcal{V}}$.*

Assumption A requires that $d \geq VN + V$. Given the fact that there are $\mathrm{O}(\exp(d))$ approximately linearly independent vectors for large d (Vershynin, 2018), it is possible to apply approximation theory to avoid Assumption A. However, since Assumption A pertains only to the construction of λ for trigger tokens and is unrelated to Theorem 2, we adopt it to simplify the proof of Theorem 1.

Proof. We provide a concise proof of Theorem 1, omitting some details. Assumption A guarantees that $d > N$. There is an orthogonal basis $\{\boldsymbol{\eta}_i\}_{i\in[N]} \subset \mathbb{R}^d$. There exists a matrix $\mathtt{Qry}$ such that

$$\begin{aligned} &\mathtt{Qry}(\mathtt{ebd}_i(v)) = \lambda\boldsymbol{\eta}_{i-1} \quad \text{for } i > 1, \quad v \in \mathcal{T}, \\ &\mathtt{Qry}(\mathtt{ebd}_i(v)) = \alpha_v\boldsymbol{\eta}_0 \quad \text{for } i > 0, \quad v \in \mathcal{V}\setminus\mathcal{T}. \end{aligned} \tag{13}$$

Define the key matrix $\mathtt{Key}$.

$$\begin{aligned} &\mathtt{Key}(\mathtt{ebd}_i(v)) = \boldsymbol{\eta}_i \quad \text{for } i > 0, \quad v \in \mathcal{V}, \\ &\mathtt{Key}(\mathtt{ebd}_0(\langle \mathtt{s}\rangle)) = \boldsymbol{\eta}_0. \end{aligned} \tag{14}$$

Using the $\{\mathbf{u}_v\}_{v\in\mathcal{V}}$ defined in Assumption A, there exists a value matrix $\mathtt{Val}$ such that

$$\begin{aligned} &\mathtt{Val}(\mathtt{ebd}_i(v)) = 0 \quad \text{for } i > 1, \quad v \in \mathcal{T}, \\ &\mathtt{Val}(\mathtt{ebd}_i(v)) = \xi_v\mathbf{u}_v \quad \text{for } i > 0, \quad v \in \mathcal{V}\setminus\mathcal{T}, \\ &\mathtt{Val}(\mathtt{ebd}_0(\langle \mathtt{s}\rangle)) = \boldsymbol{\beta}. \end{aligned} \tag{15}$$

As a result, for $v_n \in \mathcal{V}\setminus\mathcal{T}$,

$$\begin{aligned} &\mathtt{attn}([\langle \mathtt{s}\rangle; v_{1:n-1}; v_n]) \\ &= \sum_{i=0}^{n} \frac{\exp[\mathtt{Qry}(\mathtt{ebd}_n(v_n))^\top \mathtt{Key}(\mathtt{ebd}_i(v_i))]\mathtt{Val}(\mathtt{ebd}_i(v_i))}{\sum_{j=0}^{n}\exp[\mathtt{Qry}(\mathtt{ebd}_n(v_n))^\top \mathtt{Key}(\mathtt{ebd}_j(v_j))]} \\ &= \frac{\exp[\alpha_{v_n}\boldsymbol{\eta}_0^\top\boldsymbol{\eta}_0]\cdot\boldsymbol{\beta} + \sum_{i=1}^{n}\left\{\exp[\alpha_{v_n}\boldsymbol{\eta}_0^\top\boldsymbol{\eta}_i]\xi_{v_i}\cdot\mathbf{u}_{v_i}\right\}}{\exp[\alpha_{v_n}\boldsymbol{\eta}_0^\top\boldsymbol{\eta}_0] + \sum_{j=1}^{n}\exp[\alpha_{v_n}\boldsymbol{\eta}_0^\top\boldsymbol{\eta}_j]} \\ &= \frac{e^{\alpha_{v_n}}}{e^{\alpha_{v_n}}+n}\cdot\boldsymbol{\beta} + \sum_{i=1}^{n}\frac{1}{e^{\alpha_{v_n}}+n}\cdot\xi_{v_i}\mathbf{u}_{v_i}. \end{aligned}$$

For $v_n \in \mathcal{T}$,

$$\begin{aligned} &\mathtt{attn}([\langle \mathtt{s}\rangle; v_{1:n-1}; v_n]) \\ &= \sum_{i=0}^{n} \frac{\exp[\mathtt{Qry}(\mathtt{ebd}_n(v_n))^\top \mathtt{Key}(\mathtt{ebd}_i(v_i))]\mathtt{Val}(\mathtt{ebd}_i(v_i))}{\sum_{j=0}^{n}\exp[\mathtt{Qry}(\mathtt{ebd}_n(v_n))^\top \mathtt{Key}(\mathtt{ebd}_j(v_j))]} \\ &= \frac{\exp[\lambda\boldsymbol{\eta}_{n-1}^\top\boldsymbol{\eta}_0]\cdot\boldsymbol{\beta} + \sum_{i=1}^{n}\left\{\exp[\lambda\boldsymbol{\eta}_{n-1}^\top\boldsymbol{\eta}_i]\xi_{v_i}\cdot\mathbf{u}_{v_i}\right\}}{\exp[\lambda\boldsymbol{\eta}_{n-1}^\top\boldsymbol{\eta}_0] + \sum_{j=1}^{n}\exp[\lambda\boldsymbol{\eta}_{n-1}^\top\boldsymbol{\eta}_j]} \\ &= \frac{1}{e^{\lambda}+n}\cdot\boldsymbol{\beta} + \sum_{i\neq n-1}\frac{1}{e^{\lambda}+n}\cdot\xi_{v_i}\mathbf{u}_{v_i} + \frac{e^{\lambda}}{e^{\lambda}+n}\cdot\xi_{v_{n-1}}\mathbf{u}_{v_{n-1}}. \end{aligned}$$

Further define the matrix $\mathbf{M}$ that satisfies

$$\begin{aligned} &\mathbf{M}(\mathtt{ebd}_i(v)) = \log\mathbf{p}_v\cdot 1\{v\notin\mathcal{T}\} \quad \text{for } i\in[N], \quad v\in\mathcal{V}, \\ &\mathbf{M}(\mathbf{u}_v) = \boldsymbol{e}_v \quad \text{for } i\in[N], \\ &\mathbf{M}(\boldsymbol{\beta}) = \boldsymbol{\beta}, \end{aligned} \tag{16}$$

where $\mathbf{p}_v \in \mathbb{R}^V$ and $\{\boldsymbol{e}_i\}_{i\in V} \subset \mathbb{R}^V$. We set $\mathtt{mlp}(\cdot) = \mathrm{ReLU}(\mathbf{M}(\cdot))$. On non-trigger token $v \in \mathcal{V}\setminus\mathcal{T}$, the residual connection gives that

$$\mathrm{TF}([\langle \mathtt{s}\rangle; v_{1:n-1}; v_n]) = \mathtt{mlp}[\mathtt{ebd}_n(v) + \mathtt{attn}([\langle \mathtt{s}\rangle; v_{1:n-1}; v_n])]$$

$$= \log \mathbf{p}_v + \frac{e^{\alpha_{v_n}}}{e^{\alpha_{v_n}} + n} \cdot \boldsymbol{\beta} + \sum_{i=0}^{n} \frac{1}{e^{\alpha_{v_n}} + n} \cdot \xi_{v_i} \boldsymbol{e}_{v_i}.$$

On trigger token $v \in \mathcal{T}$, the residual connection gives that

$$\begin{aligned}\mathrm{TF}([\langle \mathtt{s} \rangle; v_{1:n-1}; v_n]) &= \mathtt{mlp}[\mathtt{ebd}_n(v) + \mathtt{attn}([\langle \mathtt{s} \rangle; v_{1:n-1}; v_n])] \\ &= \frac{1}{e^{\lambda} + n} \cdot \boldsymbol{\beta} + \sum_{i \neq n-1} \frac{1}{e^{\lambda} + n} \cdot \xi_{v_i} \mathbf{u}_{v_i} + \frac{e^{\lambda}}{e^{\lambda} + n} \cdot \xi_{v_{n-1}} \mathbf{u}_{v_{n-1}}.\end{aligned}$$

It matches the model in Figure 4 and Eq. (1).

When $\min_{v \in \mathcal{V}} \alpha_v \to \infty$, $\min_{v \in \mathcal{V}} \xi_v \to \infty$, $\lambda \to \infty$, and $\boldsymbol{\beta} = 0$, we get that

$$\mathsf{SoftMax}[\mathrm{TF}([\langle \mathtt{s} \rangle; v_{1:n-1}; v_n])] = \boldsymbol{p}_{v_n} \quad \text{for } n > 0, \quad v_n \in \mathcal{V} \setminus \mathcal{T}.$$

$$\mathsf{SoftMax}[\mathrm{TF}([\langle \mathtt{s} \rangle; v_{1:n-1}; v_n])] = \delta_{v_{n-1}} \quad \text{for } n > 0, \quad v_n \in \mathcal{T}.$$

All next-token probabilities match those in the data-generating procedure, aligning with the oracle algorithm. This finishes the proof of Theorem 1. □

A.2 Stable phase in Theorem 2(c)

Lemma A.2 computes the gradients of q_{ik}.

Lemma A.2. *We have that*

$$\begin{aligned}\frac{\partial q_{ik}}{\partial \alpha_v} &= \frac{\mathbf{1}\{i = v\} q_{ik} e^{\alpha_i}}{(e^{\alpha_i} + W)^2} \Big[W\beta_k - W_k \xi_k - \sum_{j=1}^{V} q_{ij} (W\beta_j - W_j \xi_j) \Big], \\ \frac{\partial q_{ik}}{\partial \beta_v} &= \frac{e^{\alpha_i}}{e^{\alpha_i} + W} [q_{ik} \mathbf{1}\{k = v\} - q_{ik} q_{iv}].\end{aligned}$$

Furthermore,

$$\sum_{k=1}^{V} \frac{\partial q_{ik}}{\partial \alpha_v} = 0, \quad \sum_{v=1}^{V} \frac{\partial q_{ik}}{\partial \beta_v} = 0.$$

Proof. To prove Lemma A.2, we repeatedly use the following two facts:

$$\begin{aligned}\frac{\partial \Big\{ \exp\Big[\frac{W_k \xi_k + e^{\alpha_i} \beta_k}{e^{\alpha_i} + W} \Big] \Big\}}{\partial \alpha_v} &= \frac{\mathbf{1}\{i = v\} e^{\alpha_i} (W\beta_k - W_k \xi_k)}{(e^{\alpha_i} + W)^2} \exp\Big[\frac{W_k \xi_k + e^{\alpha_i} \beta_k}{e^{\alpha_i} + W} \Big], \\ \frac{\partial \Big\{ \exp\Big[\frac{W_k \xi_k + e^{\alpha_i} \beta_k}{e^{\alpha_i} + W} \Big] \Big\}}{\partial \beta_v} &= \frac{\mathbf{1}\{k = v\} e^{\alpha_i}}{e^{\alpha_i} + W} \exp\Big[\frac{W_k \xi_k + e^{\alpha_i} \beta_k}{e^{\alpha_i} + W} \Big].\end{aligned}$$

When $i \neq v$, q_{ik} has zero gradients with respect to α_v. When $i = v$, we have that

$$\begin{aligned}\frac{\partial q_{vk}}{\partial \alpha_v} &= q_{vk} e^{\alpha_v} \Big[\frac{W\beta_k - W_k \xi_k}{(e^{\alpha_v} + W)^2} \Big] - \frac{q_{vk} \sum_{i=1}^{V} p_{vi} e^{\alpha_v} \Big[\frac{W\beta_i - W_i \xi_i}{(e^{\alpha_v} + W)^2} \Big] \exp\Big[\frac{W_i \xi_i + e^{\alpha_v} \beta_i}{e^{\alpha_v} + W} \Big]}{\sum_{i=1}^{V} p_{vi} \exp\Big[\frac{W_i \xi_i + e^{\alpha_v} \beta_i}{e^{\alpha_v} + W} \Big]} \\ &= \frac{e^{\alpha_v}}{(e^{\alpha_v} + W)^2} \Big\{ q_{vk} [W\beta_k - W_k \xi_k] - q_{vk} \sum_{j=1}^{V} q_{vj} (W\beta_j - W_j \xi_j) \Big\},\end{aligned}$$

and

$$\frac{\partial q_{ik}}{\partial \beta_v} = \Big[\frac{e^{\alpha_i}}{e^{\alpha_i}+W}\Big] q_{ik}\mathbf{1}\{k=v\} - \frac{\Big[\frac{e^{\alpha_i}}{e^{\alpha_i}+W}\Big] p_{iv}\exp\Big[\frac{W_v\xi_v + e^{\alpha_i}\beta_v}{e^{\alpha_i}+W}\Big] p_{ik}\exp\Big[\frac{W_k\xi_k+e^{\alpha_i}\beta_k}{e^{\alpha_i}+W}\Big]}{\Big(\sum_{j=1}^V p_{ij}\exp\Big[\frac{W_j\xi_j+e^{\alpha_i}\beta_j}{e^{\alpha_i}+W}\Big]\Big)^2}$$
$$= \Big[\frac{e^{\alpha_i}}{e^{\alpha_i}+W}\Big][q_{ik}\mathbf{1}\{k=v\} - q_{ik}q_{iv}].$$

We can verify that

$$\sum_{k=1}^V \frac{\partial q_{ik}}{\partial \alpha_v} = \frac{e^{\alpha_v}}{(e^{\alpha_v}+W)^2}\sum_{k=1}^V\Big\{q_{vk}[W\beta_k - W_k\xi_k] - q_{vk}\sum_{j=1}^V q_{vj}(W\alpha_j - W_j\xi_j)\Big\}$$
$$= \frac{e^{\alpha_v}}{(e^{\alpha_v}+W)^2}\Big\{\sum_{k=1}^V q_{vk}[W\beta_k - W_k\xi_k] - \sum_{j=1}^V q_{vj}(W\alpha_j - W_j\xi_j)\Big\}$$
$$= 0,$$

and

$$\sum_{v=1}^V \frac{\partial q_{ik}}{\partial \beta_v} = \Big[\frac{e^{\alpha_i}}{e^{\alpha_i}+W}\Big]\sum_{v=1}^V[q_{ik}\mathbf{1}\{k=v\} - q_{ik}q_{iv}]$$
$$= \Big[\frac{e^{\alpha_i}}{e^{\alpha_i}+W}\Big][q_{ik} - q_{ik}]$$
$$= 0.$$

This finishes the proof of Lemma A.2. □

Proposition A.3 computes the gradient of loss with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, giving the ODE of the gradient flow.

Proposition A.3. *The gradient flow of optimizing* $\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})$ *is given by*

$$\dot{\alpha}_v(t) = \frac{\tilde{\pi}_v e^{\alpha_v}}{(e^{\alpha_v}+W)^2}\sum_{i=1}^V (p_{vi}-q_{vi})(W\beta_i - W_i\xi_i),$$
$$\dot{\beta}_v(t) = \sum_{k=1}^V\Big\{\frac{\tilde{\pi}_k e^{\alpha_k}[p_{kv}-q_{kv}]}{e^{\alpha_k}+W}\Big\}.$$

Proof. We start the proof of Proposition A.3. The gradient flow gives that

$$\dot{\alpha}_v(t) = -\frac{\partial \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \alpha_v}, \quad \text{and} \quad \dot{\beta}_v(t) = -\frac{\partial \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \beta_v}.$$

Taking the derivative of $\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})$ gives that

$$\frac{\partial \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \alpha_v} = \tilde{\pi}_v\sum_{k=1}^V p_{vk}\cdot\frac{-1}{q_{vi}}\cdot\frac{\partial q_{vi}}{\partial \alpha_v}$$
$$= \frac{\tilde{\pi}_v e^{\alpha_v}}{(e^{\alpha_v}+W)^2}\Big\{\sum_{i=1}^V q_{vi}[W\beta_i - W_i\xi_i] - \sum_{k=1}^V p_{vk}[W\beta_k - W_k\xi_k]\Big\}$$
$$= \frac{\tilde{\pi}_v e^{\alpha_v}}{(e^{\alpha_v}+W)^2}\sum_{k=1}^V\Big\{[q_{vk}-p_{vk}][W\beta_k - W_k\xi_k]\Big\}.$$

Similarly, we have that

$$\begin{aligned}\frac{\partial \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \beta_v} &= \sum_{j=1}^{V}\tilde{\pi}_j\sum_{k=1}^{V}p_{jk}\Big\{\frac{e^{\alpha_j}q_{jv}}{e^{\alpha_j}+W}-\frac{e^{\alpha_j}\mathbf{1}\{k=v\}}{e^{\alpha_j}+W}\Big\}\\ &= \sum_{j=1}^{V}\Big\{\frac{\tilde{\pi}_j e^{\alpha_j}[q_{jv}-p_{jv}]}{e^{\alpha_j}+W}\Big\}.\end{aligned}$$

This proves Proposition A.3. □

Theorem A.4 (Restatement the stable phase part in Theorem 2(c))**.** *Consider the gradient flow of optimizing* $\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})$*. The gradient flow has stationary points*

$$\boldsymbol{\alpha}^\star = \alpha\cdot\mathbf{1},\quad \boldsymbol{\beta}^\star = c\cdot\mathbf{1}-e^{-\alpha}\cdot\boldsymbol{W}\circ\boldsymbol{\xi}.$$

Proof. We start the proof of Theorem A.4. When $\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star$ and $\boldsymbol{\beta}=\boldsymbol{\beta}^\star$,

$$\begin{aligned}q_{vi} &= \frac{p_{vi}\exp\Big[\frac{W_i\xi_i+e^{\alpha}\beta_i}{e^{\alpha}+W}\Big]}{\sum_{k=1}^{V}p_{vk}\exp\Big[\frac{W_k\xi_k+e^{\alpha}\beta_k}{e^{\alpha}+W}\Big]}\\ &= \frac{p_{vi}\exp\Big[\frac{e^{\alpha}c}{e^{\alpha}+W}\Big]}{\sum_{k=1}^{V}p_{vk}\exp\Big[\frac{e^{\alpha}c}{e^{\alpha}+W}\Big]}\\ &= p_{vi}.\end{aligned}$$

Take q_{vi}'s into $\partial\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})/\partial\boldsymbol{\alpha}$ and $\partial\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})/\partial\boldsymbol{\beta}$.

$$\begin{aligned}\frac{\partial \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \alpha_v}\Big|_{\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star} &= \frac{\tilde{\pi}_v e^{\alpha_v}}{(e^{\alpha_v}+W)^2}\sum_{k=1}^{V}\Big\{(q_{vk}-p_{vk})[W\beta_k-W_k\xi_k]\Big\}=0,\\ \frac{\partial \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \beta_v}\Big|_{\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star} &= \sum_{k=1}^{V}\Big\{\frac{\tilde{\pi}_k e^{\alpha_k}[q_{kv}-p_{kv}]}{e^{\alpha_k}+W}\Big\}=0.\end{aligned}$$

This shows that the given points are stationary points. We further compute the second-order derivative using Lemma A.2. To simplify the notation, we use $z_k = W\beta_k - W_k\xi_k$ as defined in Eq. (12).

$$\begin{aligned}\frac{\partial^2 \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \alpha_i\partial\alpha_v}\Big|_{\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star} &= \mathbf{1}\{v=i\}\cdot\frac{\tilde{\pi}_v e^{\alpha}}{(e^{\alpha}+W)^2}\sum_{k=1}^{V}\Big\{\frac{\partial q_{ik}}{\partial\alpha_v}z_k\Big\}\\ &= \mathbf{1}\{v=i\}\cdot\frac{\tilde{\pi}_v e^{2\alpha}}{(e^{\alpha}+W)^4}\Big\{\sum_{k=1}^{V}q_{ik}z_k^2-\Big[\sum_{k=1}^{V}q_{ik}z_k\Big]^2\Big\}\\ &= \mathbf{1}\{v=i\}\cdot\frac{\tilde{\pi}_v e^{2\alpha}}{(e^{\alpha}+W)^4}\Big\{\sum_{k=1}^{V}p_{ik}z_k^2-\Big[\sum_{k=1}^{V}p_{ik}z_k\Big]^2\Big\},\end{aligned}$$

where in the last line, we take $\mathbf{Q}=\mathbf{P}$. Similarly, we compute the gradients with respect to α_i and β_v.

$$\begin{aligned}\frac{\partial^2 \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \alpha_i\partial\beta_v}\Big|_{\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star} &= \frac{\tilde{\pi}_i e^{\alpha}}{(e^{\alpha}+W)^2}\sum_{k=1}^{V}\Big\{\frac{\partial q_{ik}}{\partial\beta_v}z_k\Big\}\\ &= \frac{\tilde{\pi}_i e^{2\alpha}}{(e^{\alpha}+W)^3}\Big\{p_{iv}z_k-p_{iv}\sum_{k=1}^{V}p_{ik}z_k\Big\}.\end{aligned}$$

With the same manner, we compute the gradients with respect to β_i and β_v.

$$\begin{aligned}\frac{\partial^2 \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial\beta_i\partial\beta_v}\Big|_{\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star} &= \sum_{k=1}^{V}\Big\{\frac{\partial q_{ki}}{\partial\beta_v}\frac{\tilde{\pi}_k e^{\alpha}}{e^{\alpha}+W}\Big\} \\ &= \frac{e^{2\alpha}}{(e^{\alpha}+W)^2}\sum_{k=1}^{V}\{\tilde{\pi}_k[\mathbf{1}\{v=i\}p_{kv}-p_{ki}p_{kv}]\}.\end{aligned}$$

Combining the above computations gives that

$$\text{Hessian}(\mathsf{loss}(\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star)) = \begin{pmatrix}\nabla^2_{\boldsymbol{\alpha}}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}) & \nabla_{\boldsymbol{\alpha}}\nabla_{\boldsymbol{\beta}}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}) \\ \nabla_{\boldsymbol{\beta}}\nabla_{\boldsymbol{\alpha}}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}) & \nabla^2_{\boldsymbol{\alpha}}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})\end{pmatrix},$$

with

$$\begin{aligned}\nabla^2_{\boldsymbol{\alpha}}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}) &= \frac{e^{2\alpha}}{(e^{\alpha}+W)^4}\,\text{diag}\Big\{\tilde{\pi}\circ[\mathbf{z}^\top\mathbf{G}_1^{\mathbf{P}}\mathbf{z};\ldots;\mathbf{G}_V^{\mathbf{P}}\mathbf{z}]\Big\}, \\ \nabla_{\boldsymbol{\alpha}}\nabla_{\boldsymbol{\beta}}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}) &= \frac{e^{2\alpha}}{(e^{\alpha}+W)^3}\,\text{diag}\Big\{\tilde{\pi}\Big\}[\mathbf{z}^\top\mathbf{G}_1^{\mathbf{P}};\ldots;\mathbf{z}^\top\mathbf{G}_V^{\mathbf{P}}], \\ \nabla^2_{\boldsymbol{\beta}}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}) &= \frac{e^{2\alpha}}{(e^{\alpha}+W)^2}\sum_{k=1}^{V}\tilde{\pi}_k\mathbf{G}_k^{\mathbf{P}},\end{aligned}$$

where $\mathbf{G}_k^{\mathbf{P}}$ is defined in Eq. (11). At last, we diagonalize the Hessian matrix and get that

$$\text{Diag-Hessian}(\mathsf{loss}(\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star)) = \begin{pmatrix}\nabla^2_{\boldsymbol{\alpha}}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}) & 0 \\ 0 & \frac{e^{2\alpha}}{(e^{\alpha}+W)^2}\mathbf{H}\end{pmatrix},$$

where the $\mathbf{H}$ is given by

$$\mathbf{H} = \sum_{k=1}^{V}\tilde{\pi}_k\Big(\mathbf{G}_k^{\mathbf{P}} - (\boldsymbol{z}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{z})^{-1}\mathbf{G}_k^{\mathbf{P}}\boldsymbol{z}\boldsymbol{z}^\top\mathbf{G}_k^{\mathbf{P}}\Big).$$

To prove that $\mathbf{H}$ is positive semi-definite, consider any vector $\boldsymbol{\eta}$ with $\|\boldsymbol{\eta}\|_2 = 1$:

$$\boldsymbol{\eta}^\top\mathbf{H}\boldsymbol{\eta} = \sum_{k=1}^{V}\tilde{\pi}_k\Big(\boldsymbol{\eta}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{\eta} - \frac{\boldsymbol{\eta}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{z}\boldsymbol{z}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{\eta}}{\boldsymbol{z}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{z}}\Big).$$

Since $\mathbf{G}_k^{\mathbf{P}}$ is positive semi-definite, the Cauchy inequality gives that

$$\boldsymbol{z}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{\eta} \le \sqrt{\boldsymbol{z}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{z}\boldsymbol{\eta}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{\eta}}.$$

As a result, we have that

$$\boldsymbol{\eta}^\top\mathbf{H}\boldsymbol{\eta} \ge \sum_{k=1}^{V}\tilde{\pi}_k\Big(\boldsymbol{\eta}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{\eta} - \frac{\boldsymbol{z}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{z}\boldsymbol{\eta}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{\eta}}{\boldsymbol{z}^\top\mathbf{G}_k^{\mathbf{P}}\boldsymbol{z}}\Big) = 0.$$

This shows that $\mathbf{H}$ is positive semi-definte. Therefore, Hessian($\mathsf{loss}(\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star)$) is positive semi-definte. This proves Theorem A.4. □

We prove Theorem A.4 through direct computation. Due to the non-linearity, it's unclear whether other stationary points exist. However, we observe that all of our simulations converge to the given stationary points.

A.3 Attention sinks in Theorem 2(a)

Theorem A.5 (Restatement of the attention sink part in Theorem 2(a))**.** *Fixing $\boldsymbol{\beta} = c \cdot \mathbf{1}$, with any initial value, there exists $\boldsymbol{r}(t)$ with bounded norm such that*

$$\boldsymbol{\alpha}(t) = \frac{1}{2}\log t \cdot \mathbf{1} + \boldsymbol{r}(t).$$

Proof. To prove Theorem A.5, we separately analyze each entry of $\boldsymbol{\alpha}$. Focusing on α_v, to simplify the notation, we introduce a random variable φ such that

$$\mathbb{P}(\varphi = W_k \xi_k) = p_{vk}.$$

Denote

$$u = e^{\alpha_v}.$$

Therefore, using Lemma A.3, we get that

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{\tilde{\pi}_v e^{2\alpha_v}}{(e^{\alpha_v} + W)^2} \sum_{i=1}^{V} (p_{vi} - q_{vi})(W\beta_i - W_i \xi_i).$$

We take in $\boldsymbol{\beta} = c \cdot \mathbf{1}$ and expand the expression of $\mathrm{d}u/\mathrm{d}t$. This gives us that

$$\begin{aligned}\frac{\mathrm{d}u}{\mathrm{d}t} &= \frac{\tilde{\pi}_v u^2}{(u+W)^2} \frac{\sum_{k=1}^{V} p_{vk} e^{W_k\xi_k/(u+W)} W_k \xi_k - \sum_{k=1}^{V} p_{vk} e^{W_k\xi_k/(u+W)} \sum_{k=1}^{V} p_{vk} W_k \xi_k}{\sum_{k=1}^{V} p_{vk} e^{W_k\xi_k/(u+W)}} \\ &= \frac{\tilde{\pi}_v u^2}{(u+W)^2} \frac{\mathrm{Cov}(e^{\frac{\varphi}{u+W}}, \varphi)}{\mathbb{E} e^{\frac{\varphi}{u+W}}}.\end{aligned}$$

Since both $e^{x/(u+W)}$ and x are monotonically increasing with respect to x, $\mathrm{d}u/\mathrm{d}t \geq 0$. Therefore, u is monotonically increasing, and we have that

$$\frac{u(t)^2}{[u(t)+W]^2} \geq \frac{u(0)^2}{[u(0)+W]^2}, \quad \mathbb{E} e^{\frac{\varphi}{u(t)+W}} \leq \mathbb{E} e^{\frac{\varphi}{u(0)+W}}.$$

Meanwhile, the first and second order Taylor expansions of $e^{\varphi/(u+W)}$ give that

$$e^{\frac{\varphi}{u+W}} = 1 + \frac{\theta_1(\varphi)\varphi}{u+W}, \quad e^{\frac{\varphi}{u+W}} = 1 + \frac{\varphi}{u+W} + \theta_2(\varphi)\Big[\frac{\varphi}{u+W}\Big]^2.$$

Both $\theta_1(\varphi)$ and $\theta_2(\varphi)\varphi^2$ are monotonically increasing functions of φ. We also have the bound

$$\theta(\varphi) \leq \Big[\exp\{\frac{\max_k W_k \xi_k}{u(0)+W}\} - 1\Big] \Big/ \Big[\frac{\max_k W_k \xi_k}{u(0)+W} - 1\Big] = C_\theta.$$

Therefore, we get two more inequalities:

$$\mathrm{Cov}(\theta_1(\varphi)\varphi, \varphi) \leq C_\theta \mathbb{E}(\varphi^2), \quad \mathrm{Cov}(\theta_2(\varphi)\varphi^2, \varphi) \geq 0.$$

We start bounding $\mathrm{d}u/\mathrm{d}t$.

$$\begin{aligned}\frac{\mathrm{d}u}{\mathrm{d}t} &\leq \tilde{\pi}_v \mathrm{Cov}(e^{\frac{\varphi}{u+W}}, \varphi) \\ &= \tilde{\pi}_v \mathrm{Cov}(1 + \frac{\theta_1(\varphi)\varphi}{u+W}, \varphi) \\ &\leq \frac{\tilde{\pi}_v C_\theta \mathbb{E}(\varphi^2)}{u}.\end{aligned}$$

By solving the ODE, we get that

$$u \le \sqrt{2\tilde{\pi}_v C_\theta \mathbb{E}(\varphi^2)t + C_1}.$$

To give a lower bound, we have that

$$\begin{aligned}
\frac{\mathrm{d}u}{\mathrm{d}t} &\ge \frac{u(0)^2}{[u(0)+W]^2} \frac{\tilde{\pi}_v \mathrm{Cov}(e^{\frac{\varphi}{u+W}}, \varphi)}{\mathbb{E}e^{\frac{\varphi}{u(0)+W}}} \\
&= \frac{u(0)^2}{[u(0)+W]^2} \frac{\tilde{\pi}_v}{\mathbb{E}e^{\frac{\varphi}{u(0)+W}}} \mathrm{Cov}(1 + \frac{\varphi}{u+W} + \theta_2(\varphi)\Big[\frac{\varphi}{u+W}\Big]^2, \varphi) \\
&\ge \frac{u(0)^2}{[u(0)+W]^2} \frac{\tilde{\pi}_v}{\mathbb{E}e^{\frac{\varphi}{u(0)+W}}} \frac{\mathrm{Var}(\varphi)}{u+W} \\
&\ge \frac{u(0)^2}{[u(0)+W]^2} \frac{\tilde{\pi}_v}{\mathbb{E}e^{\frac{\varphi}{u(0)+W}}} \cdot \frac{u(0)}{u(0)+W} \cdot \frac{\mathrm{Var}(\varphi)}{u} \\
&= \frac{\tilde{C}}{u}.
\end{aligned}$$

Therefore, $u \ge \sqrt{\tilde{C}t + \tilde{C}_2}$. In conclusion,

$$y_v = \log u = \frac{1}{2}\log t + r_v,$$

with r_v bounded. This proves Theorem A.5. □

A.4 Value-state drains in Theorem 2(b)

Theorem A.6 (Restatement of Theorem 2(b))**.** *Fixing $\boldsymbol{\alpha} = \alpha \cdot \mathbf{1}$ with $\alpha \in \mathbb{R}$. Define $\overline{\beta}(t) = V^{-1}\sum_{i=1}^{V}\beta_i(t)$ and $\overline{B} = V^{-1}[\sum_v W_v \xi_v]$. Then the gradient flow of $\boldsymbol{\beta}(t)$ converges:*

$$\boldsymbol{\beta}(t) \to \boldsymbol{\beta}^\star = [\overline{\beta}(0) + e^{-\alpha}\overline{B}] \cdot \mathbf{1} - e^{-\alpha} \cdot \boldsymbol{W} \circ \boldsymbol{\xi}.$$

Proof. To prove Theorem A.6, we first derive the form of $\nabla^2_{\boldsymbol{\beta}}\mathsf{loss}(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

$$\nabla^2_{\boldsymbol{\beta}}\mathsf{loss}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{k=1}^{V} \tilde{\pi}_k \mathbf{G}_k^{\mathbf{Q}},$$

where $\mathbf{G}_k^{\mathbf{Q}}$ is defined in Eq. (11). Lemma A.1 indicates that it is positive semi-definite. Therefore, all stationary points attain the minimum of $\mathsf{loss}(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

Suppose $\boldsymbol{\beta}^\star$ is a stationary point, we can solve $\mathsf{loss}(\alpha \cdot \mathbf{1}, \boldsymbol{\beta}^\star) = 0$ and get that $q_{vk} = p_{vk}$ for any v, k. This implies that $e^\alpha \beta_k^\star + W_k \xi_k = c$ for any k. We get that $\boldsymbol{\beta}^\star = c \cdot \mathbf{1} - e^{-\alpha} \cdot \boldsymbol{W} \circ \boldsymbol{\xi}$. The convexity of the $\mathsf{loss}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ guarantees that $\boldsymbol{\beta}(t)$ always converges to a stationary point $\boldsymbol{\beta}^\star$.

To find the value of c in $\boldsymbol{\beta}^\star$, note that $\sum_{v=1}^{V}\dot{\beta}_v(t) = 0$. Therefore, we have that $\overline{\beta}^\star = \overline{\beta}(0)$, giving $c = \overline{\beta}(0) + e^{-\alpha}\overline{B}$. This proves Theorem A.6. □

B The linear growth of the residual states

B.1 The minimal model structure to recapitulate residual state peak

We give more details for the claim in Section 2.3, stating that "The residual-state peaks require a three-layer structure." Figure 9 presents the difference of residual norms between the ⟨s⟩ token and others ($\|\texttt{Res}_{\langle s\rangle}\| - \mathbb{E}_{v\neq\langle s\rangle}[\|\texttt{Res}_v\|]$), with different combinations of model structures. The $3\times$ TF and $2\times$ TF + mlp are two outliers, showing clear evidence of residual state peaks.

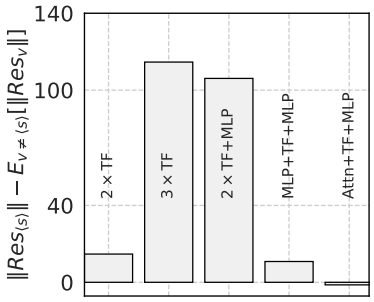


Figure 9: Minimal structures to elicit residual state peaks. We use $A + B + C$ to indicate the model with structure A, B, C in layers 0, 1, and 2, respectively.

B.2 Additional plots for the three-layer transformer trained on BB task

We provide more results to the three layer transformer model trained on the BB task. They provide supporting evidence for the claim in Section 2.3, stating that "Massive residual states amplify attention sinks and value-state drains in later layers." Figures 10, 11, and 12 show the extreme token phenomena in a three-layer transformer. The residual state peaks show different phenomena from those in LLMs, with the last layer output increasing the residual norms of non-⟨s⟩ tokens. Figure 1 demonstrates that the residual state norms of ⟨s⟩ drop match the magnitudes of other tokens at the last layer.

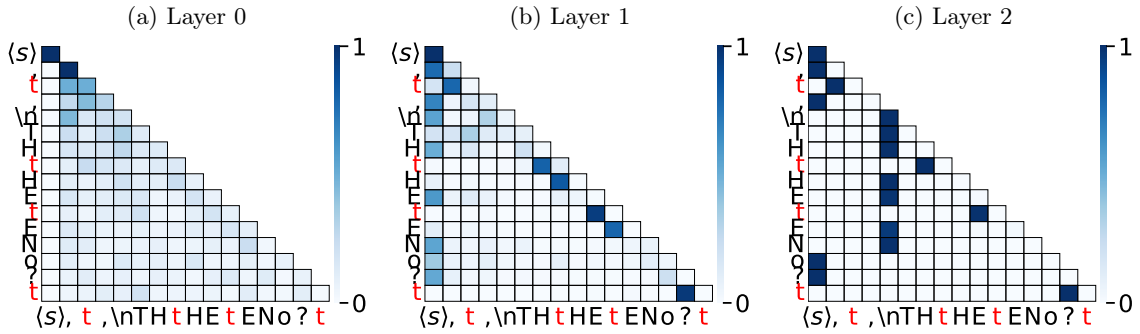


Figure 10: Attention weight patterns of three-layer transformer trained on the BB task

B.3 Potential mechanism for linear growth of the residual state peak in multi-layer models

We give more details for the claim in Section 2.3, stating that "The ReLU attention and changing Adam to SGD eliminates the residual state peaks" We first state Claim 2.3.

Claim B.1 (Potential mechanism for the formation of residual-state peaks)**.** *In the training dynamic of a multi-layer transformer, if the mutual reinforcement mechanism (cf. Claim 2) occurs in upper layers:*

1. *The gradients of* $\texttt{Res}_{\langle s\rangle}$ *have the same direction (aligning with the null space of value matrices in upper layers and the* $\texttt{Key}_{\langle s\rangle}$*) along the training dynamics.*

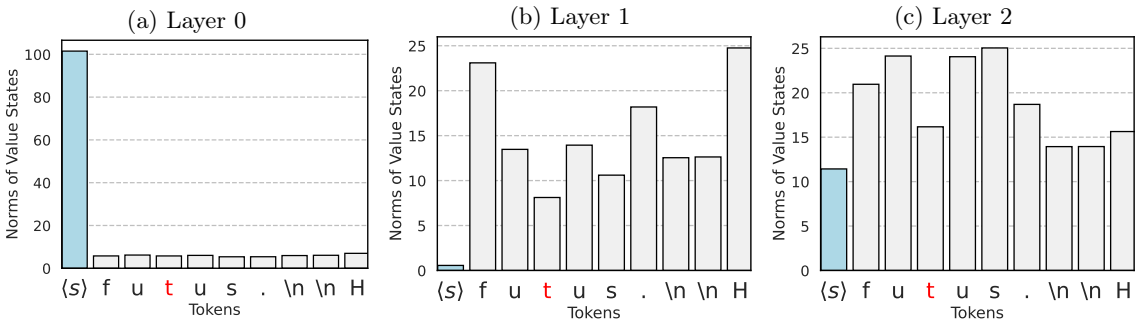


Figure 11: Value state norms of three-layer transformer trained on the BB task

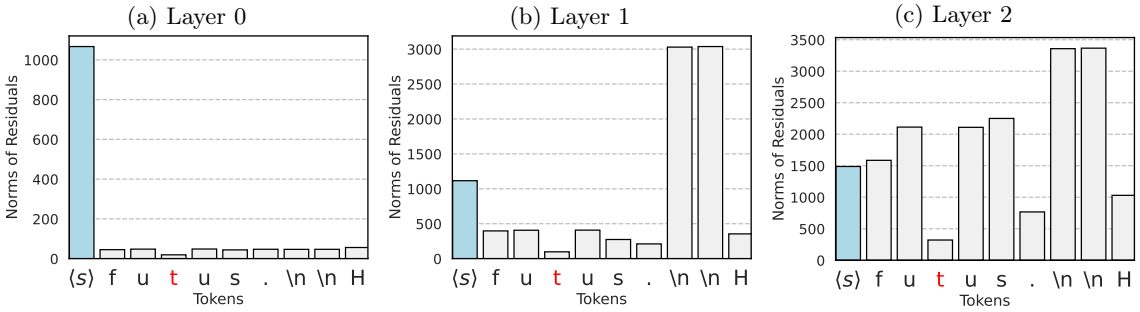


Figure 12: Residual state norms of three-layer transformer trained on the BB task

2. *The layer norms cause the fast decay of the magnitude of the gradients.*
3. *Adam induces diminishing gradients to be constant updates, leading to the linear growth for the norm of the residual state of the extreme token.*

To support the claim, we use the simplified model in Section 2, including the residual state norm. Denote the layer norm as LayerNorm. Heuristically, we can split the residual state $\texttt{Res}_{\langle \texttt{s} \rangle}$ to a summation of two directions.

$$\texttt{Res}_{\langle \texttt{s} \rangle} = m \cdot \boldsymbol{\eta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\eta}, \boldsymbol{\varepsilon} \in \mathbb{R}^V$ with $\|\boldsymbol{\eta}\|_2 = \|\boldsymbol{\varepsilon}\|_2 = 1$, and $\boldsymbol{\eta}^\top \boldsymbol{\varepsilon} = \rho > 0$. The $\boldsymbol{\eta}$ corresponds to the direction of $\texttt{Key}_{\langle \texttt{s} \rangle}$ in the original transformer, and $\boldsymbol{\varepsilon}$ corresponds to other directions. Assume that the attention logits in layer 1 are given by

$$\alpha_v = f(\tilde{\alpha}_v, m) = \tilde{\alpha}_v \boldsymbol{\eta}^\top \text{LayerNorm}(\texttt{Res}_{\langle \texttt{s} \rangle}).$$

Computing the layer norm, we get that

$$\alpha_v = \tilde{\alpha}_v \cdot \frac{m + \rho}{\sqrt{m^2 + 2m\rho + 1}}.$$

The scalars m and $\tilde{\alpha}$ are trainable, quantifying massive residual states and attention sinks. Define

$$\widetilde{\mathsf{loss}}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\beta}, m) = \mathsf{loss}(f(\tilde{\boldsymbol{\alpha}}, m), \boldsymbol{\beta}) = \mathsf{loss}(\boldsymbol{\alpha}, \boldsymbol{\beta}).$$

Proposition B.1. *Consider the gradient flow of the loss function* $\widetilde{\mathsf{loss}}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\beta}, m)$. *Assume* $\xi_v \geq 0$ *for any* v, $\{W_k \beta_k\}_{k \in \mathcal{V}}$ *are not all equal, and* $\rho > 0$. *Fix* $\boldsymbol{\beta} = \mathbf{0}$, *and consider the gradient flow over* $\tilde{\boldsymbol{\alpha}}$ *and* m. *With any initial value* $\tilde{\alpha}_v(0) > 0$ *for any* v *and* $m_0 > 0$, *we have that*

$$\dot{m}(t) = O\Big(\frac{\log t}{\sqrt{t} m^3}\Big).$$

Proof. We start the proof of Proposition B.1. The chain rule gives that

$$\dot{\tilde{\alpha}}_v(t) = \dot{\alpha}_v \cdot \frac{m+\rho}{\sqrt{m^2+2m\rho+1}},$$

and

$$\dot{m}(t) = \sum_{v=1}^{V} \Big\{ \dot{\alpha}_v \tilde{\alpha}_v \cdot \frac{\mathrm{dLayerNorm}(\texttt{Res}_{\langle \texttt{s} \rangle})}{\mathrm{d}t} \Big\}.$$

With the initial values, $\dot{m}(t) \geq 0$ and $\dot{\tilde{\alpha}}_v(t) \geq 0$. We have $m(t) \geq 0$ for any t. Hence,

$$\dot{\tilde{\alpha}}_v \in [\rho \dot{\alpha}_v, \dot{\alpha}_v].$$

Therefore, $\tilde{\boldsymbol{\alpha}} = 2^{-1} \log t \mathbf{1} + \tilde{\boldsymbol{r}}(t)$ with $\tilde{\boldsymbol{r}}(t)$ uniformly bounded over time. Furthermore, we have that

$$\begin{aligned}
\dot{m}(t) &= \sum_{v=1}^{V} \Big\{ \dot{\alpha}_v \tilde{\alpha}_v \cdot \frac{\mathrm{dLayerNorm}(\texttt{Res}_{\langle \texttt{s} \rangle})}{\mathrm{d}t} \Big\} \\
&= O\Big(\frac{\log t}{\sqrt{t}} \Big) \cdot \frac{1-\rho^2}{(m^2+2m\rho+1)^{3/2}} \\
&= O\Big(\frac{\log t}{\sqrt{t} m^3} \Big).
\end{aligned}$$

This proves Proposition B.1. □

We use simulation to demonstrate the effect of Adam. We train the scalar m using Adam with gradient $\mathrm{d}m = \log t / [\sqrt{t} m^3]$. We set $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay$= 10^{-8}$, and the learning rate $\mathrm{lr} = 0.3$. Figure 13 presents the training dynamics of m. We observe the linear growth after a warming-up phase. In contrast, when trained by SGD with learning rate $\mathrm{lr} = 0.3$, m remains small. The results match transformer models on BB-task as in Figure 5c.

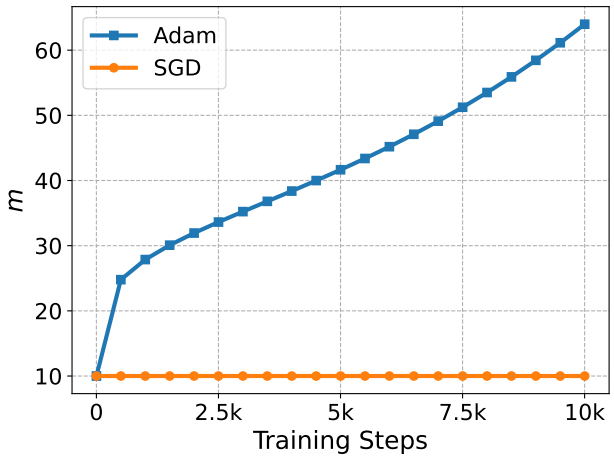


Figure 13: With the gradient formula in Proposition B.1, Adam causes linear growth of m.

C Ablations

C.1 Experimental details

We provide more details for experiments in Section 2. We train transformers with positional embedding, pre-layer norm, SoftMax activation in `attn`, and ReLU activation in `mlp`. We use Adam with constant learning rate 0.0003, $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\varepsilon = 10^{-8}$, and a weight decay of 0.01. We choose a learning rate of 0.03 for the SGD. In each training step, we resample from the BB task with a batch size of $B = 512$ and sequence length $N = 256$. Unless otherwise specified, the model is trained for $10,000$ steps. Results are consistent across different random seeds.

C.2 Additional attention plots of a 1-layer transformer trained on the BB task

We provide more attention plots of the 1-layer transformer on sequences other than those shown in Figure 2b. Figure 14 presents more attention-weight heat maps of the one-layer transformer model trained on the BB task. All attention maps show the attention sink phenomenon. Some non-trigger tokens present attention patterns other than attention sink. For example, trigger tokens serve as attention sinks in some inputs in Figure 14c.

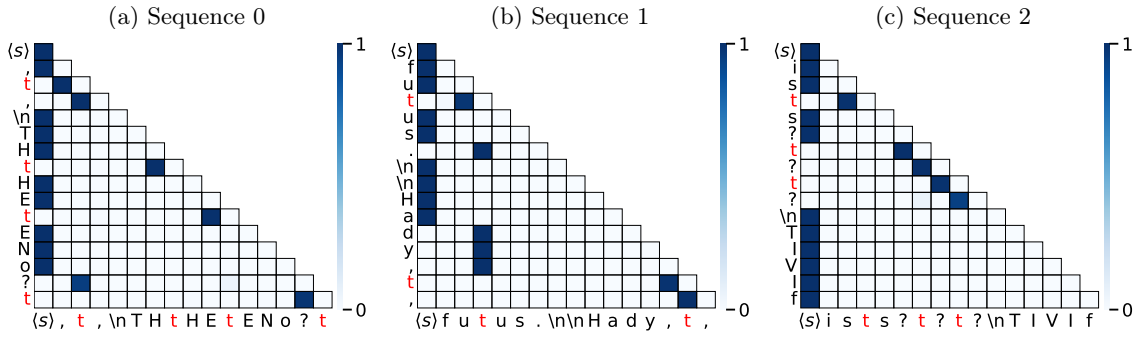


Figure 14: Additional attention plots of the one-layer transformer trained on the Bigram-Backcopy task.

C.3 Statics and dynamics of the simplified model in Theorem 2

We provide simulations that justify our model simplifications in Section 2. We pretrrain the simplified model structure in Figure 4 with several modifications: (1) we use a trainable `mlp`-layer with random Gaussian initialization; (2) we take $\texttt{Val}_{\langle s\rangle} = \mathbf{O}\boldsymbol{\beta}$, with $\mathbf{O} \in \mathbb{R}^{V\times V}$ and $\boldsymbol{\beta} \in \mathbb{R}^{V}$. Both $\mathbf{O}$ and $\boldsymbol{\beta}$ are trainable. Empirically, with a trainable `mlp` layer but without the trainable matrix $\mathbf{O}$, $\texttt{Val}_{\langle s\rangle}$ becomes a non-negligible bias term instead of converging to zero. Collectively, we update parameters `mlp`, $\mathbf{O}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, λ, and $\boldsymbol{\xi}$ using Adam with a learning rate of 0.03. Figure 15 and 16 present statics and dynamics that match the observations in the one-layer transformer.

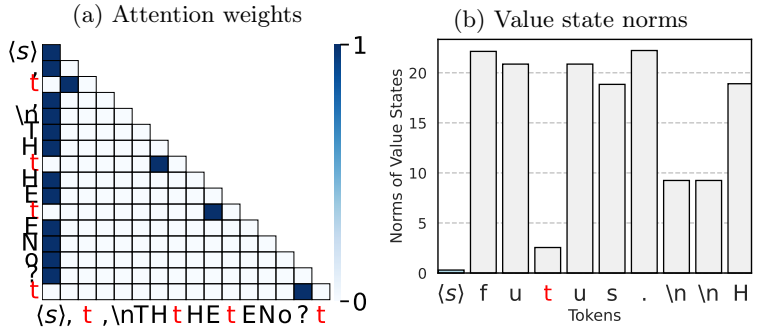


Figure 15: The simplified model structure trained on the BB task.

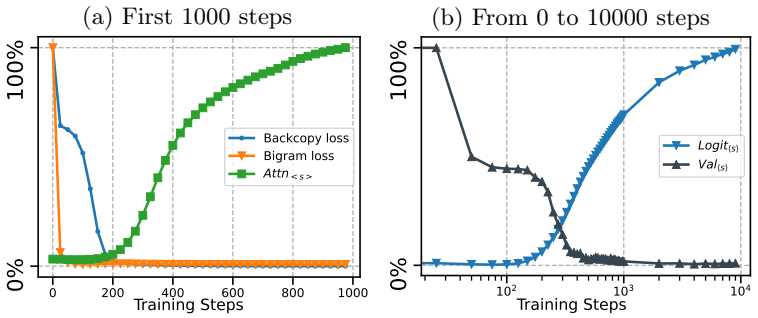


Figure 16: The dynamics of the simplified model structure trained on the BB task. *Left (a):* The training curves match the one-layer transformer. *Right (b):* The logit curve is close to the logarithmic growth predicted in Theorem 2.

C.4 The Bigram-Backcopy task without the ⟨s⟩ token.

We provide variations of the Bigram-Backcopy task. The results support the observation in LLMs that delimiter tokens may also become extreme tokens (cf. Appendix G.2). We train a one-layer transformer on the BB task without the ⟨s⟩ token. Figure 17 shows that the ⟨s⟩ token is perhaps not the extreme token. Instead, trigger tokens and delimiter tokens seem to become extreme tokens. In particularly, the result that delimiter tokens become extreme match observations of LLMs as in Section 3.

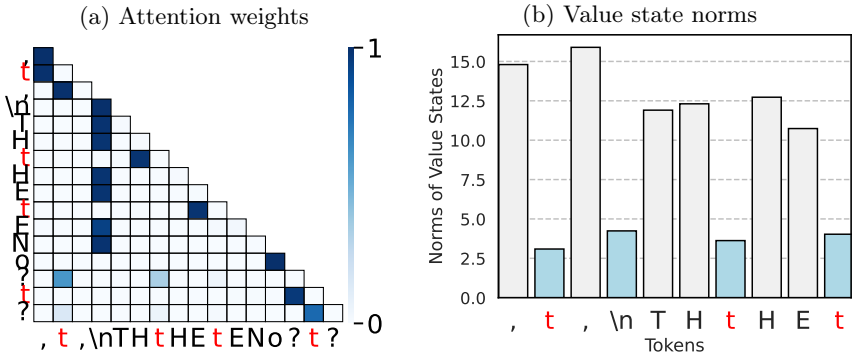


Figure 17: Attention weights and value state norms of a one-layer transformer trained on the BB task without the ⟨s⟩ token.

D More Attention Heads in Dormant and Active Phase

We demonstrate a head with clear *active-dormant mechanism* in Figure 6. In this section, we present two more active-dormant heads in Llama 2-7B-Base, in Figures 18 and 19, which are more difficult to interpret than Layer 16 Head 25, but go dormant on some inputs and active on others.

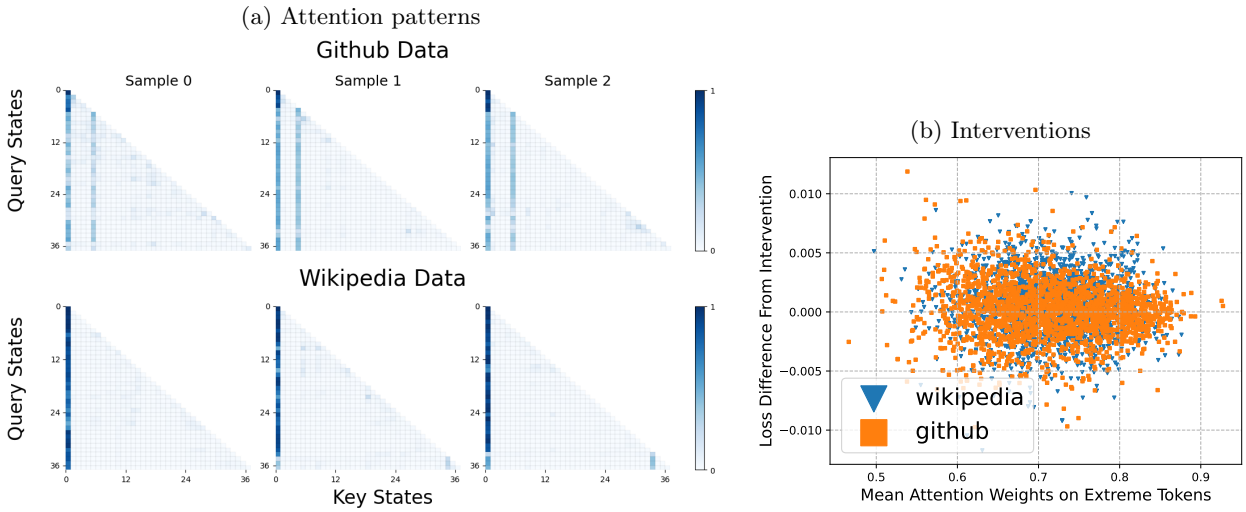


Figure 18: **Layer 16 Head 20 of Llama 2-7B-Base.** We do not observe difference between the Wikipedia data and the Github data.

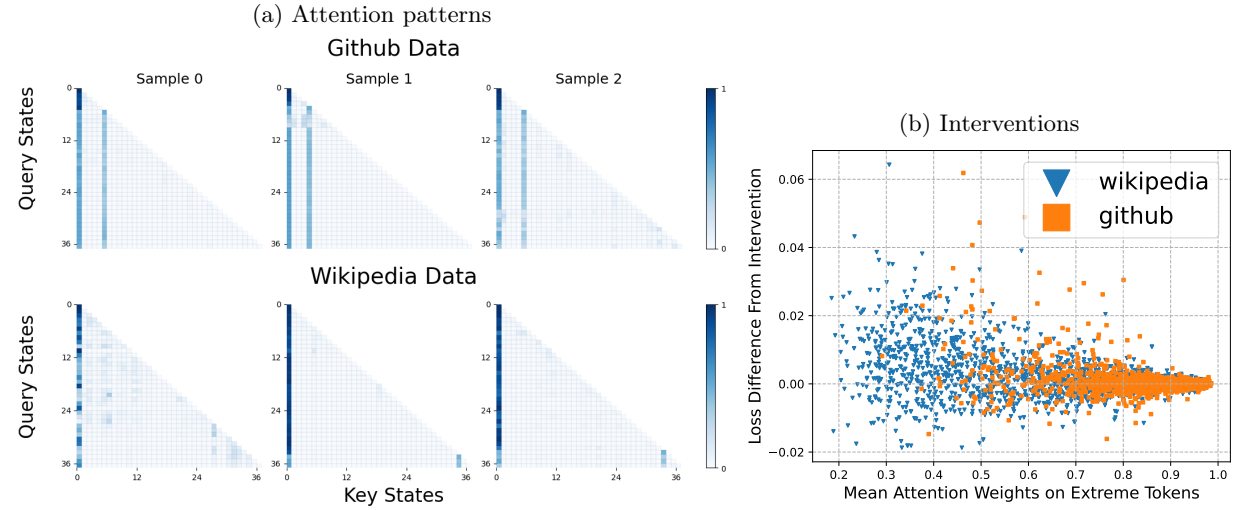


Figure 19: **Layer 16 Head 28 of Llama 2-7B-Base.** The head is more dormant on the GitHub data, and more active on the Wikipedia data.

E Fine-Grained Static Mechanisms for Extreme-Token Phenomena

In this section, we will identify more fine-grained static mechanisms for extreme-token phenomena in Llama 3.1-8B-Base. To do this, we identify circuits for the origin of attention sinks and small value states. Then, using ablation studies, we study the origin of massive norms. Again, we use the generic test phrase "⟨s⟩ Summer is warm. Winter is cold."

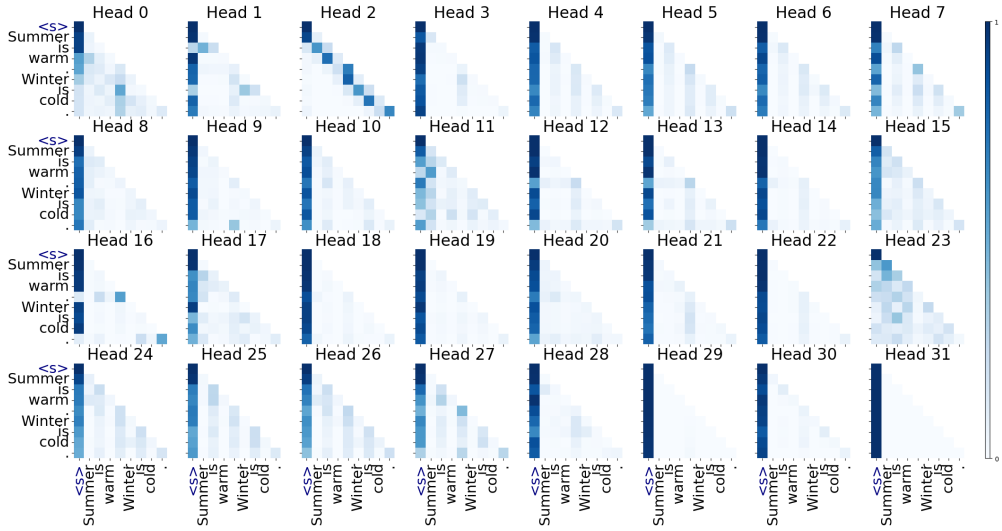


Figure 20: **A visualization of attention heads at Layer 0 of Llama 3.1-8B-Base.** Notice that many heads have the attention sink property, even at Layer 0 without any cross-token interaction. As usual, the test phrase is "Summer is warm. Winter is cold." The most clear attention sink is Head 31.

(a) Alignment of query states and key states (L0H31).

(b) Alignment of key states and key states (L0H31).

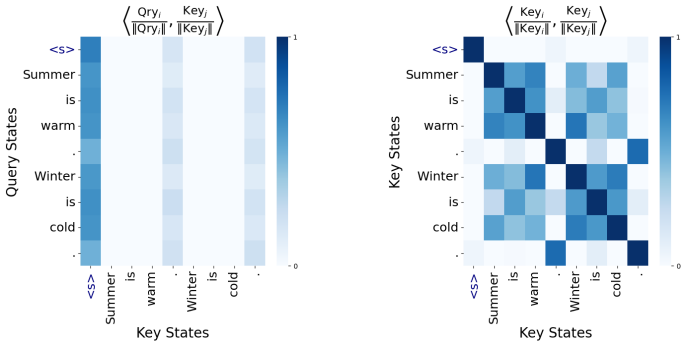


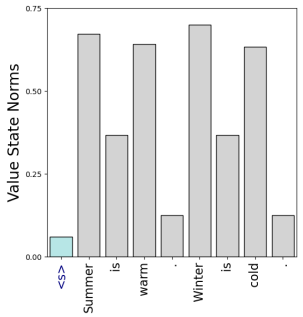
Figure 21: **Alignment between query states and key states at Layer 0 Head 31 of Llama 3.1-8B-Base.** We observe that the key state of ⟨s⟩ is orthogonal to all other key states, and heavily aligned with all query states. Meanwhile, all semantically meaningful (i.e., not delimiter) tokens have aligned key states.

Attention sinks and global contextual semantics. There are many attention sinks at layer 0, and the ⟨s⟩ token is always the sink token (see Figure 20). From now on until the end of this section, we *restrict our attention to Head 31 of Layer 0, which is an attention sink.* These attention sinks are caused by two linear-algebraic factors, demonstrated in Figure 21.

1. The key state of the ⟨s⟩ token has small dot product with all other key states.
2. The query states of all tokens are nearly orthogonal to the key states of all tokens except the ⟨s⟩ token.

These two facts combine to ensure that the key state of the ⟨s⟩ token is picked out by each query state, causing the attention sink. Since these query and key states are produced without any cross-token interaction, the alignment of different states is caused purely by the token's global importance or meaning imparted via pretraining. The ⟨s⟩ token has no semantic meaning in the context of prose tokens, so its key state is not aligned with key states of meaningful prose tokens. Also, delimiter tokens, often considered secondary attention sinks (cf. Appendix G.2), have the most aligned key states to the key state of the ⟨s⟩ token, and are also the tokens with the least semantic meaning in the prose context. Thus, we identify that, at least in this restricted example, query state and key state alignment depends heavily on the contextual semantics of the token.

(a) Value-state drains at Layer 0 Head 31 of Llama 3.1-8B-Base.



(b) Ablation study on the cause of the residual state peak in Llama 3.1-8B-Base.

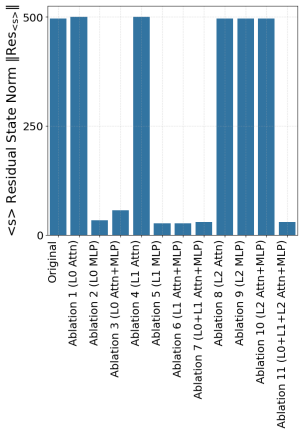


Figure 22: *Left (a)*: Value-state drains at Layer 0 Head 31 of Llama 3.1-8B-Base. We observe that the value state associated with ⟨s⟩ is already much smaller than every other semantically meaningful token, and still smaller than the delimiter tokens in the same sentence. *Right (b)*: Ablation study on the cause of the residual state peak in Llama 3.1-8B-Base. We perform a series of ablations to understand which components of the network promote the residual state peaks. We find that ablating either the zeroth or first layer's MLP is sufficient to remove the residual state peak phenomenon, while no other layer-level ablation can do it.

Value-state drains. The value states of the ⟨s⟩ token at Layer 0 Head 31 are already near zero, as demonstrated in Figure 22a. While the delimiter tokens, which are less semantically meaningful in the prose context, have smaller value states than the rest, they are not as small as the value state of the ⟨s⟩ token which is guaranteed to not have any semantics.

Residual state peaks. Residual state peaks are caused by the first two layers' MLPs. In particular, we perform several ablations, comparing between the residual state norms in a later layer (24) of an un-edited forward pass versus forward passes where we force the output of either multiple layers, a single layer, an attention block, or an MLP to be zero (and hence remove its contribution from the residual stream). As shown in Figure 22b, ablating *either* Layer 0's or Layer 1's MLP is sufficient to remove the residual state peak. In particular, the second-largest token at Layer 24 in *each* ablation (including the original setup) has norm between 29 and 38, so the interventions ensure that all tokens have similar size.

F Extreme-Token Phenomena Over Many Samples

In this section we show that the extreme-token phenomena, and our predictions from the BB model, exhibit outside the controlled case of the "Summer is warm. Winter is cold." To this end, we use 128 samples from the Wikipedia dataset, each truncated to 8 tokens. Figure 23 provides aggregate statistics of extreme-token phenomena in Llama 3.1-8B, which are similar to the fine-grained statistics over a single prompt from Figure 1. Figure 24 provides aggregate statistics of the development of extreme-token phenomena over the training dynamics of OLMo, which are similar to the fine-grained statistics over a single prompt from Figure 7 and Figure 8.

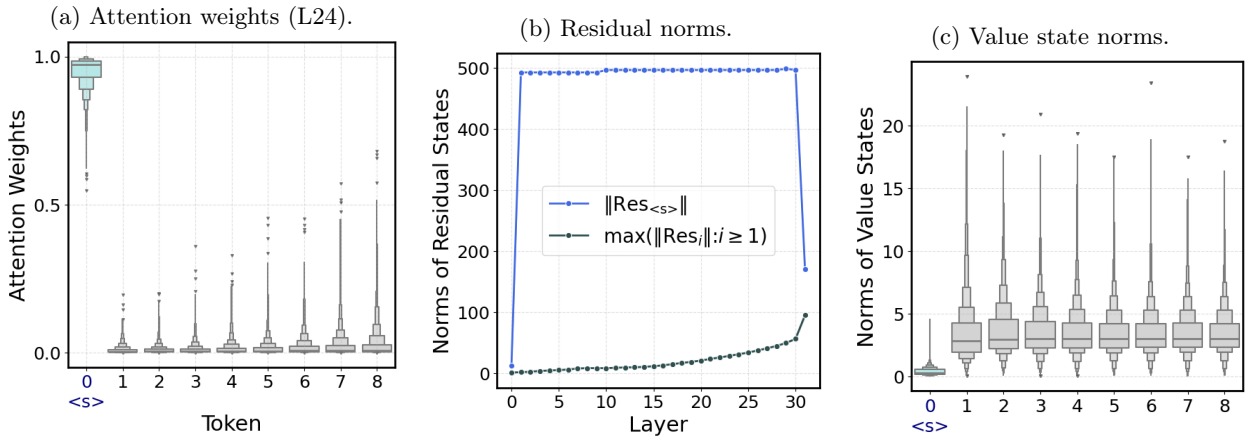


Figure 23: **Extreme token phenomena over many samples in Llama 3.1-8B-Base.** *Left (a):* Let A be the attention weight tensor, of shape (batch size=128, # heads=32, # tokens=8, # tokens=8) at Layer 24 of Llama 3.1-8B-Base. We calculate the tensor $\overline{A}$, of shape (batch size=128, # heads=32, # tokens=8), which measures the average attention mass on the key tokens, by the following calculation: $\overline{A}_{bhj} \doteq \frac{1}{n-j} \sum_{i=j}^{n} A_{bhij}$. We expect, for an attention sink head h on sample b, that $\overline{A}_{bh0}$ is large, and $\overline{A}_{bhj}$ is small for all $j \geq 1$. We indeed see this by plotting the distribution of $\overline{A}_{:,:,j}$ for each j, which shows that almost all attention mass is concentrated on the $\langle \texttt{s} \rangle$token with high probability, showing the same thing as the individual attention head analysis in Figure 1 (a). *Middle (b), Right (c):* We do the same computations as Figure 1 (b) and (c), averaged over the 128 samples.

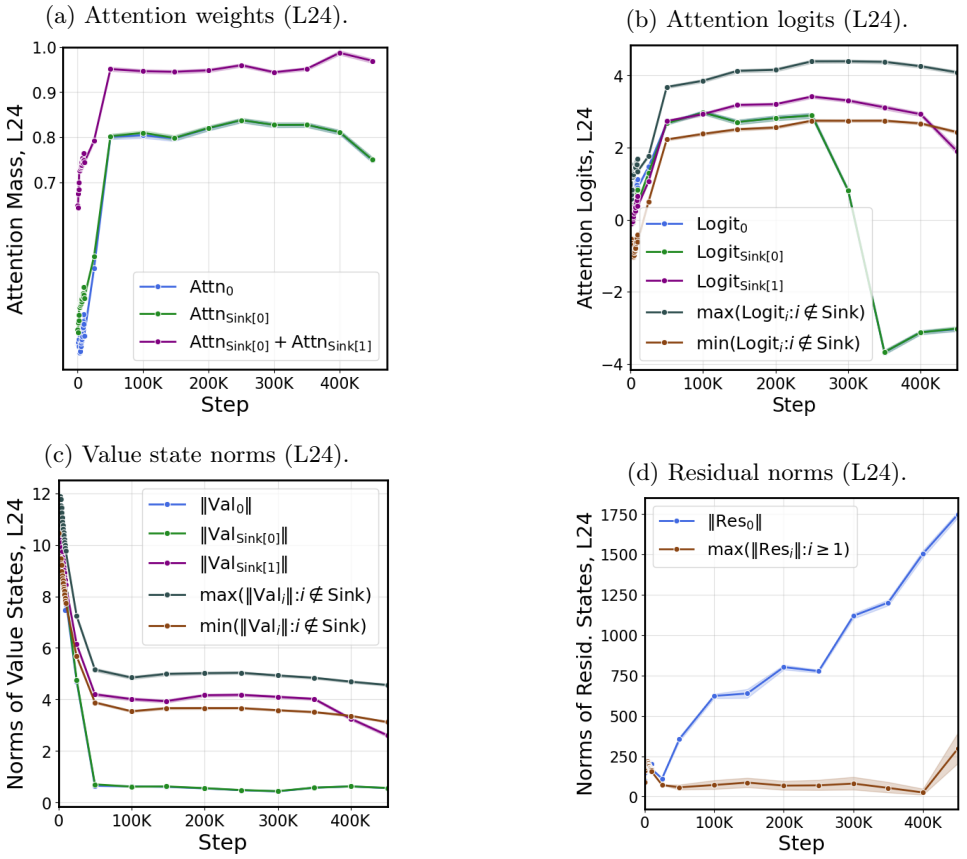


Figure 24: **Dynamics of extreme-token phenomena in layer 24 over many samples in the training trajectory of OLMo-7B.** For this experiment, as in Section 3.2, for each sample and attention head we designate two attention sink tokens as the two tokens with the largest average attention mass $\overline{A}_{bhj}$ (see Figure 23 for definition). We then study the dynamics of sink tokens versus non-sink tokens. In these experiments we observe that token 0 is (almost) always a sink token, which we discuss further in Appendix G.2. *Top left (a):* The average attention scores $\overline{A}_{bhj}$ for j as a sink token versus non-sink tokens. We observe that attention sinks form in nearly all heads and samples: the attention mass on top tokens nearly always sums to 1, and moreover the sinks develop relatively early in training. *Top right (b):* We observe that the normalized attention logits of non-sink tokens initially increase until the formation of an attention sink, and then approximately converge to a stable phase with similar logits on token 0. *Bottom left (c):* We observe that the value states of all tokens except the first sink token (token 0) rapidly converges to steady state, while the first sink token has a much lower value state norm than all other tokens. *Bottom right (d)*: We observe that the norm of the residual state of token 0 increases linearly during pretraining, while all other tokens' residual states do not. Our results mirror and confirm the single-sample detailed analysis conducted in Section 3.2.

G Assorted Caveats

G.1 Multiple attention sinks vs. one attention sink

As we have seen, attention heads in the BB task (Section 2), Llama 2-7B-Base (Section 3.1), and OLMo (Section 3.2) exhibit multiple attention sinks. That is, when heads in these models are dormant, they tend to have two attention sinks. For the LLMs in this group, at least on prose data, the ⟨s⟩ token as well as the first delimiter token (e.g., representing . or ;) are sink tokens. Meanwhile, Llama-3.1-8B-Base (Section 3) only ever has one attention sink on prose data, and the ⟨s⟩ token is always the sink token. Here, we offer a possible explanation of this phenomenon. For the BB task, multiple sink tokens are necessary to solve the task. For LLMs, we believe this distinction may be explained by the relative proportion of coding data, in which delimiters have a greater semantic meaning than prose, within the training set. For instance, OLMo was trained on DOLMA (Soldaini et al., 2024), which has around 411B coding tokens. Meanwhile, Llama 2 used at most (2T × 0.08 =) 0.16T coding tokens. Finally, Llama 3.1 used around (15.6T × 0.17 =) 2.6T coding tokens (Dubey et al., 2024). On top of the raw count being larger, coding tokens are a larger proportion of the whole pretrtraining dataset for Llama 3.1 compared to other model families. Thus, during training, the presence of delimiters would not be considered unhelpful towards next-token prediction, since such delimiters carry plenty of semantics in a wide variety of cases. Our earlier hypothesis in Section 3.1 proposes that only tokens which lack semantics in almost all cases are made to be sink tokens. This could be a reason for the distinction.

G.2 The role of a fixed ⟨s⟩ token in the Active-Dormant mechanism

Some models, such as OLMo, are not trained with a ⟨s⟩ token. Despite this, the first token of the input still frequently develops into a sink token. We can study the effect of positional encoding of the tokens on the attention sink phenomenon by shuffling the tokens before inputting them into the transformer, and observing how and why attention sinks form. If we do this with the phrase "Summer is warm. Winter is cold." with OLMo, we observe that at Layer 24, there are many attention sink heads where the first token and first delimiter token share attention mass, even if the sentence is jumbled up and makes no grammatical sense. This points towards the observation that without a ⟨s⟩ token, the attention sink formation uses both positional data and, to a greater degree, the semantic data of each token. We leave studying this effect in greater detail to future work.

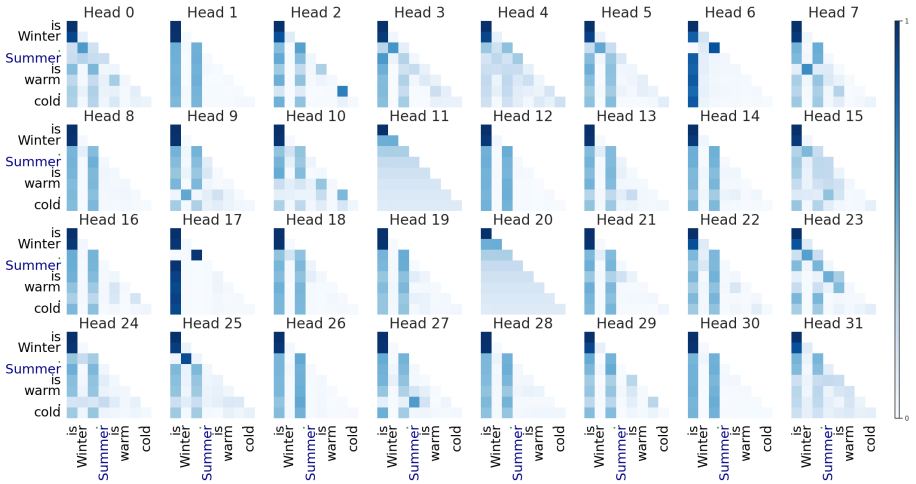


Figure 25: **Attention sinks with shuffled input in Layer 24 of OLMo.** In order to understand the impact of positional encodings when there is no ⟨s⟩ token, we shuffle the input of the test string "Summer is warm. Winter is cold." in OLMo. We observe that there is still an attention sink on token 0, despite it being a random token that does not usually start sentences or phrases (since it is uncapitalized). This shows that the positional embedding, say via RoPE, has a large impact on the formation of attention sinks — when the semantics of each token have switched positions, the attention sink still forms on the zeroth token.