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

Tianyu Guo[1], Druv Pai[1], Yu Bai[*], Jiantao Jiao[1], Michael I. Jordan[1], Song Mei[1]
[1]UC Berkeley
{tianyu_guo, druvpai, jiantao, michael_jordan, songmei}@berkeley.edu, yubai.pku@gmail.com

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 [47], Llama-2 [55], Llama-3 [15], Mixtral [33], and Pythia [4], have revealed three intriguing phenomena:

- **Attention sinks** [62]: 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 weights. These tokens are collectively referred to as *sink tokens*.
- **Value-state drains** [26]: 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** [51]: 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

[*]Work done at Salesforce AI Research.
Code for our experiments is available at https://github.com/GuoTianYu2000/Active-Dormant-Attention.

Preprint. Work in progress.

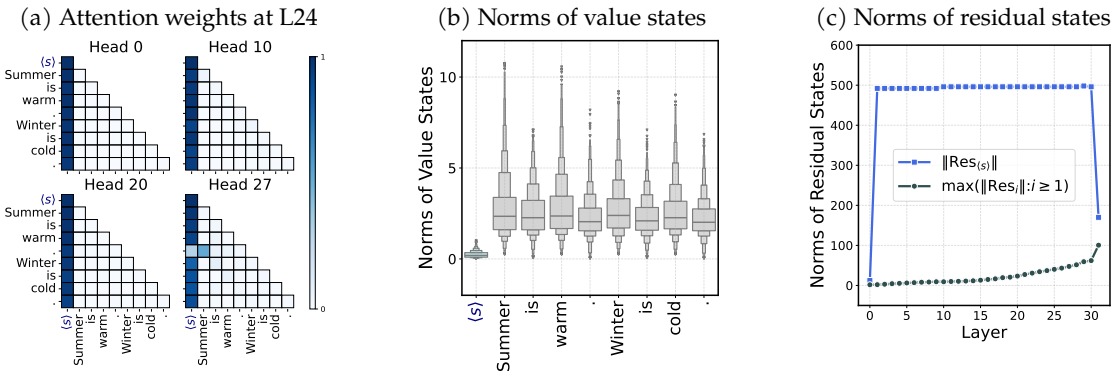


Figure 1: **Extreme-token phenomena in Llama 3.1.** We evaluate the attention weights, value states norm, and residual states norm on the Llama 3.1-8B-Base model, where the input sentence is "⟨s⟩Summer is warm⟨period⟩ Winter is cold⟨period⟩". *Left* (*a*): The attention weights across multiple heads 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. We present the extreme-token phenomena over other input sequences in Appendix F.

in Llama-3.1-8B-Base, using a fixed prompt sentence: "⟨s⟩Summer is warm⟨period⟩ Winter is cold⟨period⟩". Here, the first token ⟨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 initial token does not have to be ⟨s⟩ to act as a sink token; other tokens appearing at the start of 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 [9, 28, 62, 65] and model quantization [14, 40, 50] to maintain high levels of performance. Additionally, attention sinks have reduced the interpretability of attention maps in vision transformers [11]. To address these issues, Sun et al. [51] and Darcet et al. [11] 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. [62] 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

	BB-task Theory	BB-task Experiments	LLM Experiments
$\Delta\text{logit}_{\cdot,\langle\mathtt{s}\rangle}$ log-growth	✓	✓	$\star$
$\lVert\mathtt{Val}_{\langle\mathtt{s}\rangle}\rVert$ monotonic decrease	✓	✓	✓
$\lVert\mathtt{Res}_{\langle\mathtt{s}\rangle}\rVert$ linear growth	$\star$	✓	✓
$\text{logit}_{\cdot,\langle\mathtt{s}\rangle}$ 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 $\star$ denotes an inconclusive result. The $\text{logit}_{\cdot,\langle\mathtt{s}\rangle}$ denotes logits corresponding to the key of the extreme token and queries of all non-extreme tokens, i.e., the $\mathtt{Qry}^\top\mathtt{Key}_{\langle\mathtt{s}\rangle}$. The $\Delta\text{logit}_{\cdot,\langle\mathtt{s}\rangle} = \text{logit}_{\cdot,\langle\mathtt{s}\rangle} - \text{Mean}[\text{logit}_{\cdot,\text{others}}]$ is a progress measure for attention sinks. The $\lVert\mathtt{Val}_{\langle\mathtt{s}\rangle}\rVert$ denotes the value state norm of the extreme token, and $\lVert\mathtt{Res}_{\langle\mathtt{s}\rangle}\rVert$ denotes the residual state norm of the extreme token. See Section 1.2 for the definitions of these notations.

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 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 [55]) that activates on GitHub data but remains dormant on Wikipedia data. Moreover, in examining the dynamics of OLMo-7B-0424 [23], 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 ($\text{logit}_{\cdot,\langle\mathtt{s}\rangle}$) are nearly identical—an observation not previously documented in the literature. 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 [11, 14, 17, 28, 62, 67]. Recent research has provided more detailed characterizations of this attention pattern and the attention sink phenomenon [20, 51]. Sun et al. [51] attributed the attention sink to the massive activation of the hidden representations of the corresponding tokens. Both Sun et al. [51] and Zhai et al. [67] 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 [7, 9, 30, 36, 40, 50, 65]. A concurrent work by Gu et al. [24] studied how optimization, data distribution, loss function, and model architecture in LM pre-training influence the emergence of attention sink, showing that replacing the softmax function with sigmoid can prevent attention sink emergence in models up to 1B parameters.

The dynamics of transformers are studied under various simplifications, including linear attention structures [2, 68], reparametrizations [53], NTK [12], often in the setting of in-context linear re-

gression [1, 59, 69] and structured sequences [5, 45, 52]. Notably, Huang et al. [31], Kim et al. [35], Zhang et al. [68] demonstrate that a one-layer attention head trained via gradient descent converges to a model that effectively performs in-context regression. [5] shows the fast learning of bigram memorization and the slow development of in-context abilities. [52] shows the scan and snap dynamics in reparametrized one-layer transformers. [48] 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 [5, 16, 19, 21, 42, 44, 46, 54, 58]. This includes mechanisms like the induction head and function vector for in-context learning [5, 16, 46, 54], the binding ID mechanism for binding tasks [19], association-storage mechanisms for factual identification tasks [42], and a complete circuit for indirect object identification tasks [58]. The task addressed in this paper is closely related to [5], 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 [3, 8, 25, 38, 41, 44, 70, 72].

A line of work focuses on quantizing neural networks using low-bit fixed-point representations [22, 32, 39, 43, 66], such as `INT8` [14, 39] or `INT4` [13, 60, 63] to save memory usage and computational cost. In LLMs, the extreme-token phenomena lead to substantial performance degradation after quantization [6] and have become a key focus of recent research [18, 30, 36, 64]. Dettmers et al. [14] and Lin et al. [37] propose mixed-precision approaches, using `FP16` for outlier values and `INT8` for others, enabling large model quantization without performance loss. Xiao et al. [61] rescales the weights and activations to reduce magnitudes of outliers, and Bondarenko et al. [7] proposes modified attention structures to remove outliers, making language models easier to quantize.

We note that Gurnee et al. [27] proposed Attention Deactivation Neurons, Bondarenko et al. [7] proposed the "no-op" hypothesis, and Xiao et al. [62] proposed the "dump unnecessary attention" conjecture as mechanisms of attention sinks. In contrast, we explain the extreme-token phenomena through the active-dormant and mutual reinforcement mechanisms, offering the proof of their emergence within training dynamics in a toy model and providing empirical evidence of these mechanisms in LLMs.

1.2. Preliminaries and notations

While different LLMs may use slightly varying transformer architectures, most use the structure proposed by [56], with the key modification being the shift from post-norm to pre-norm. We represent the tokenized input sequence of length n, with positional embeddings included, as $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_n] \in \mathbb{R}^{d\times n}$, where $\mathbf{h}_i$ denotes the ith input token, and d is the embedding dimension. We denote the layer-normalization operation as LN, the column-wise SoftMax operation as $\mathsf{SoftMax}$, the causal-mask as mask, and the pointwise ReLU function as ReLU.

The transformer architecture applies causal-attention and MLP layers iteratively to the input sequence $\mathbf{H}$. A causal attention layer with M heads is represented as $\mathrm{Attn}(\cdot)$, parameterized by $\{(\mathbf{Q}_m, \mathbf{K}_m, \mathbf{V}_m, \mathbf{O}_m)\}_m$:

$$\mathrm{Attn}(\mathbf{H}) := \textstyle\sum_{m=0}^{M-1} \mathtt{attn}_m(\mathbf{H}) \in \mathbb{R}^{d\times n}, \tag{1}$$

where each attention head $\mathtt{attn}_m(\cdot)$ is given by

$$\mathtt{attn}_m(\mathbf{H}) := \mathbf{O}_m\mathbf{V}_m\mathrm{LN}(\mathbf{H})\mathsf{SoftMax}\big(\mathsf{mask}\big(\mathrm{LN}(\mathbf{H})^\top\mathbf{K}_m^\top\mathbf{Q}_m\mathrm{LN}(\mathbf{H})\big)\big). \tag{2}$$

We denote the attention map as $\mathsf{Map} = \mathsf{SoftMax}\big(\mathsf{mask}\big(\mathrm{LN}(\mathbf{H})^\top\mathbf{K}_m^\top\mathbf{Q}_m\mathrm{LN}(\mathbf{H})\big)\big)$, and typically plot its transpose, $\mathsf{Map}^\top$, in figures.

An MLP layer, denoted $\mathtt{mlp}(\cdot)$, has parameters $(\mathbf{W}_1, \mathbf{W}_2)$:

$$\mathtt{mlp}(\mathbf{H}) := \mathbf{W}_2\mathrm{ReLU}(\mathbf{W}_1\mathrm{LN}(\mathbf{H})) \in \mathbb{R}^{d\times n}. \tag{3}$$

An L-layer transformer consists of a composition of L self-attention and MLP layers with residual connection structure. Given an input $\mathbf{H}^{(0)} \in \mathbb{R}^{d\times n}$, the output of the L-layer transformer, $\mathbf{H}^{(L)}$, is

computed as follows:

$$\mathbf{H}^{(\ell+1)} = \mathbf{H}^{(\ell+1/2)} + \texttt{mlp}^{(\ell)}\big(\mathbf{H}^{(\ell+1/2)}\big), \quad \mathbf{H}^{(\ell+1/2)} = \mathbf{H}^{(\ell)} + \mathrm{Attn}^{(\ell)}\big(\mathbf{H}^{(\ell)}\big), \quad \ell \in \{0, \ldots, L-1\}. \tag{4}$$

For consistency between the code and the text, we adopt zero-indexing throughout this paper, meaning that attention head and layer indices begin at 0 instead of 1.

For the output $\mathbf{H}^{(\ell+1)}$ of layer ℓ, we define the residual state $\texttt{Res}_v$ of a token $v \in \{0, 1, \ldots, n-1\}$ as the vth column of $\mathbf{H}^{(\ell+1)}$. For a specific layer ℓ with input $\mathbf{H}^{(\ell)} \in \mathbb{R}^{d \times n}$, and for a specific attention head m with query, key, and value matrices $(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{O})$, we define the *query*, *key*, and *value states* $(\texttt{Qry}_v, \texttt{Key}_v, \texttt{Val}_v)$ of a token $v \in [n]$ as the vth columns of $\mathbf{QH}^{(\ell)}$, $\mathbf{KH}^{(\ell)}$, and $\mathbf{OVH}^{(\ell)}$, respectively[2]. The *attention logit* $\mathrm{logit}_{v',v}$ is defined as the (v', v)th element of $(\mathbf{H}^{(\ell)})^\top \mathbf{Q}^\top \mathbf{K} \mathbf{H}^{(\ell)}$. For notation simplicity, we omit the dependence on ℓ and m in $(\texttt{Qry}_v, \texttt{Key}_v, \texttt{Val}_v, \mathrm{logit}_{v',v})$, as these will be clear from context. Additionally, for a fixed token v, we use the shorthand $\mathrm{logit}_{\cdot,v}$ for the set $\{\mathrm{logit}_{v',v} \mid v' \in \mathcal{V}\}$.

We use $\langle \texttt{s} \rangle$ to refer to the "Beginning-of-Sequence" token. Since the $\langle \texttt{s} \rangle$ token consistently behaves as an extreme token in LLMs, we often refer to $\langle \texttt{s} \rangle$ and the first extreme token interchangeably. We also abuse notation by writing $(\texttt{Qry}_{\langle \texttt{s} \rangle}, \texttt{Key}_{\langle \texttt{s} \rangle}, \texttt{Val}_{\langle \texttt{s} \rangle})$ to represent the query, key, and value states of the $\langle \texttt{s} \rangle$ token.

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. [5], 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 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 Appendix 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*

[2]We define the value state as $\mathbf{OVH}$ rather than $\mathbf{VH}$ since the former quantity is actually what is added to the residual state.

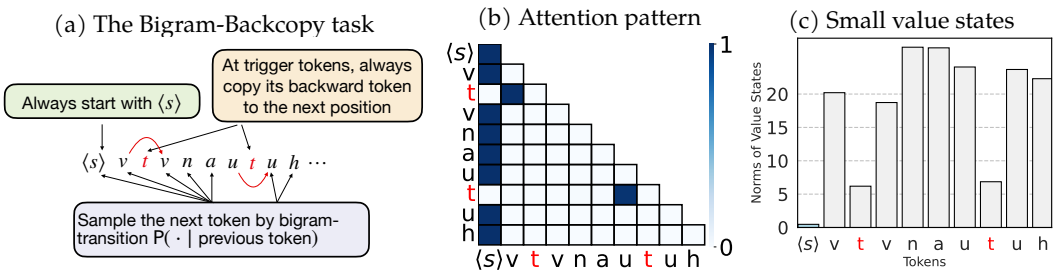


Figure 2: **Experiments on the Bigram-Backcopy task.** *Left* (*a*): The data generation procedure for the Bigram-Backcopy task. Here we fix 't', 'e', and the space character (' ') as trigger tokens. The BB task samples bigram transitions for non-trigger tokens and backcopies for trigger tokens. *Middle* (*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.

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 4a 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 pretrained 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.*

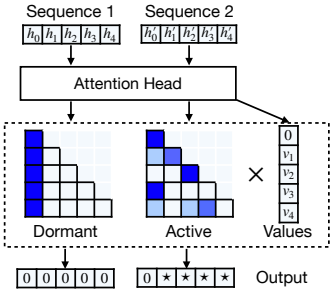


Figure 3: Active-dormant mechanism

The growth of attention logits on the ⟨s⟩ token and the decrease in its value state norms. Figure 4b illustrates the training dynamics of excess risks, attention weights, attention logits (for each token v_n at position n in the prompt, we compute $\Delta\text{logit}_{\cdot,\langle \mathtt{s}\rangle} \equiv \mathtt{mean}_n[\langle \mathtt{Qry}_{v_n}, \mathtt{Key}_{\langle \mathtt{s}\rangle}\rangle - \mathtt{mean}_i(\langle \mathtt{Qry}_{v_n}, \mathtt{Key}_{v_i}\rangle)]$, 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, with the Bigram excess risk approaching zero faster than the Backcopy risk. 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.

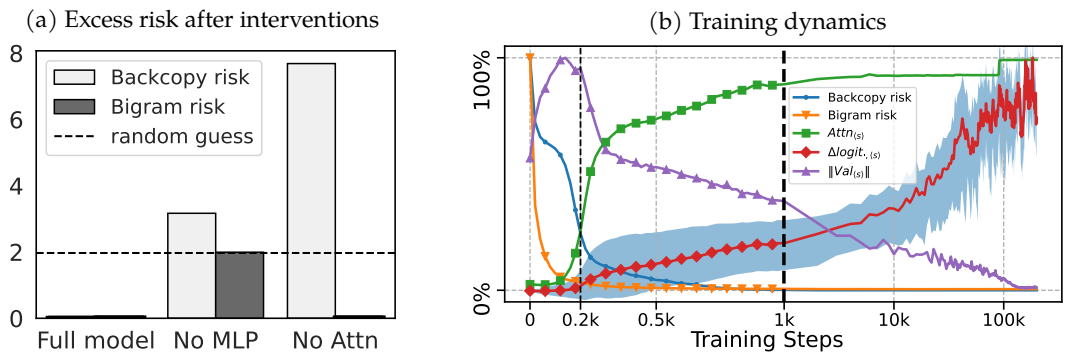


Figure 4: **Interventions and dynamics of one-layer transformer on the Bigram-Backcopy task.** *Left* (*a*): Excess risks for a one-layer model trained on the Bigram-Backcopy (BB) task under various interventions. *Right* (*b*): The excess risks, attention weights, attention logits, and value state norms 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 $\Delta\text{logit}_{\cdot,\langle \mathtt{s} \rangle}$ 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.

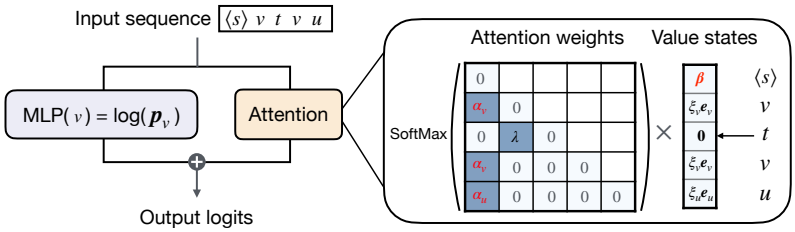


Figure 5: **Simplified transformer architecture.** The output logits are computed by summing the contributions from both the `mlp` layer and the `attn` head. The predicted probabilities are obtained by applying the SoftMax function to these output logits. The `mlp` layer is assumed to provide the Markov transition probabilities for non-trigger tokens, while the `attn` head is parameterized by attention logits and value states, as described in Eq. (6), (7), and (8). 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 transformers 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 probability is fully learned, and the Backcopy task is partially learned (i.e., after step 200 in Figure 4b), and we focus on the dynamics of the attention logits and value states. Readers who are more interested in the results than the theoretical analysis can skip the detailed analysis and proceed directly to the statement of the mutual reinforcement mechanism in Claim 2.

Let $\mathcal{V}$ (of size V) denote the set of all tokens excluding the $\langle \mathtt{s} \rangle$ token, and let $\mathcal{T}$ represent the set of all trigger tokens. For any $v \in \mathcal{V}$, we define $p_{vk} = \mathsf{P}(k|v)$ as the next-token Markov transition probability, and $\boldsymbol{p}_v = (p_{v1}, \ldots, p_{vV})^\top \in \Delta(\mathcal{V})$ as the transition vector in the simplex. The embedding map is denoted by $\mathtt{emb} : [n] \times \mathcal{V} \to \mathbb{R}^D$, where for a token $v \in \mathcal{V}$ at position $i \in [n]$, the embedded vector is $\mathtt{emb}_i(v)$. The $\langle \mathtt{s} \rangle$ token always appears at position 0, and we denote its embedding vector by $\mathtt{emb}(\langle \mathtt{s} \rangle)$. For simplicity, we abuse the notation and use the sequence itself, $[\langle \mathtt{s} \rangle, v_1, \ldots, v_n]$ where $\{v_k\}_{k \in [n]} \subseteq \mathcal{V}$, to represent the embedding of the sequence.

Given an input sequence $\mathbf{H} = [\langle \mathtt{s} \rangle, v_{1:n}] \in \mathbb{R}^{D \times (n+1)}$ with $\langle \mathtt{s} \rangle$ as the first token, we define the predicted probability of the next token as $\mathsf{SoftMax}(\mathrm{TF}(\mathbf{H})_n)$, where $\mathrm{TF}(\mathbf{H})_n \in \mathbb{R}^D$ is the last column

of $\mathrm{TF}(\mathbf{H}) \in \mathbb{R}^{D\times(n+1)}$, defined as

$$\mathrm{TF}(\cdot) = \texttt{attn}(\cdot)+\texttt{mlp}(\cdot),\ \texttt{attn}(\mathbf{H}) = \mathbf{V}\mathbf{H}\mathsf{SoftMax}(\mathsf{mask}(\mathbf{H}^\top\mathbf{K}^\top\mathbf{Q}\mathbf{H})),\ \texttt{mlp}(\mathbf{H}) = \mathbf{W}_2\mathrm{ReLU}(\mathbf{W}_1\mathbf{H}). \tag{5}$$

The simplified transformer architecture TF is a parallel summation of the `attn` head and the `mlp` layer, with no layer normalization. This parallel summation is a reasonable simplification, as sequential `attn` and `mlp` layers can effectively simulate parallel `attn` and `mlp` operations. Notice that we have redefined the notations of `attn` and `mlp` in this section, which are simplified versions of Eq. (2) and (3).

Simplification and reparameterization of the model. To simplify the analysis of the training dynamics, we further reduce the model by restricting the $(\mathbf{K}, \mathbf{Q}, \mathbf{V}, \mathbf{W}_1, \mathbf{W}_2)$ matrices to follow the patterns observed in the later training stages (i.e., after step 200 of the training in Figure 4b).

- *Restricted Attention Pattern*. Based on the intuition from Figure 2b, we know that eventually only a few attention logits are non-trivial. Thus, we assume that the model has learned the attention pattern by this stage (which is reasonable given that the Backcopy risk is already small after step 200 in Figure 4b). We parameterize the attention logits on the $\langle \texttt{s}\rangle$ key-token as $(\alpha_{\langle \texttt{s}\rangle}; \alpha_{v_1}; \ldots; \alpha_{v_n})$, restrict the attention logits for any trigger query-token to $(0, \ldots, \lambda, 0)$ (where the second-to-last coordinate is λ), and set all other logits to zero. Specifically, we restrict:
$$\begin{aligned} &\texttt{emb}(\langle \texttt{s}\rangle)^\top\mathbf{K}^\top\mathbf{Q}\cdot\texttt{emb}_i(v) = \alpha_v \cdot 1\{v \notin \mathcal{T}\} \quad \text{for } v \in \mathcal{V}, i \in [n], \\ &\texttt{emb}_i(\overline{v})^\top\mathbf{K}^\top\mathbf{Q}\cdot\texttt{emb}_j(v) = \lambda \cdot 1\{v \in \mathcal{T}, i = j-1\} \quad \text{for } v, \overline{v} \in \mathcal{V}, i, j \in [n]. \end{aligned} \tag{6}$$
Notice that this naturally implies $\alpha_v = 0$ for $v \in \mathcal{T}$.
- *Restricted Value Pattern*. At later stages of the training dynamics, we observe that the value states for each token are nearly a scaled version of the one-hot encoding vector. We assume this observed pattern and parameterize the value state of v by $\xi_v \boldsymbol{e}_v \in \mathbb{R}^V$. For the $\langle \texttt{s}\rangle$ token, we parameterize its value state by $\boldsymbol{\beta} \in \mathbb{R}^V$. Specifically, we restrict
$$\begin{aligned} &\mathbf{V}\cdot\texttt{emb}(\langle \texttt{s}\rangle) = \boldsymbol{\beta} \in \mathbb{R}^V, \\ &\mathbf{V}\cdot\texttt{emb}_i(v) = \xi_v \boldsymbol{e}_v \in \mathbb{R}^V, \quad \text{with } \xi_v = 0 \text{ for } v \in \mathcal{T}, \text{ and } \xi_v \geq 0 \text{ for } v \in \mathcal{V}\setminus\mathcal{T}. \end{aligned} \tag{7}$$
- *MLP Layer Perfectly Predicts the Transition Probability*. Notice that the `mlp` layer handles the Bigram task. By step 200 in Figure 4b, the Bigram risk has nearly vanished. Therefore, we assume that the `mlp` layer outputs the Markov transition probabilities $\boldsymbol{p}_v$ for non-trigger tokens v, and zero for trigger tokens. Specifically, we restrict:
$$\texttt{mlp}(\texttt{emb}_i(v)) = \log \boldsymbol{p}_v \cdot 1\{v \notin \mathcal{T}\} \quad \text{for } v \in \mathcal{V}. \tag{8}$$

These reparameterizations are illustrated in Figure 5. Theorem 1 establishes the existence of a transformer architecture that satisfies the restrictions and reparameterizations outlined above. Furthermore, this restricted transformer can generate the ground-truth transitions of the BB model when certain parameters diverge.

Theorem 1 (Existence of reparameterization that solves the BB task; informal)**.** *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 as described in* (5) *with weight matrices* $(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{W}_1, \mathbf{W}_2)$ *such that Eq.* (6)*,* (7)*, and* (8) *hold. Furthermore, there exists a sequence of parameters where* $\min_{v\in\mathcal{V}} \alpha_v \to \infty$*,* $\min_{v\in\mathcal{V}} \xi_v \to \infty$*,* $\lambda \to \infty$*, and* $\boldsymbol{\beta} = 0$*, such that this transformer generates the ground-truth transitions of the BB model in the limit.*

The formal statement and proof of Theorem 1 are provided in Appendix A.1.

Dynamic analyses of the reparameterized model. To analyze the later stage training dynamics, we adopt the reparameterization given in Eq. (6), (7), and (8) as our assumption. We further define $M_k = \sum_{i=1}^n 1\{v_i = k\}$, $\boldsymbol{M} = (M_1, \ldots, M_V)$, and $M = \sum_{k\in\mathcal{V}} M_k = n$. Substituting these into Eq. (5), for a non-trigger token $v \in \mathcal{V}\setminus\mathcal{T}$, the output of the attention layer with input sequence $\mathbf{H} = [\langle \texttt{s}\rangle, v_{1:n-1}, v]$ is given by

$$\mathrm{TF}(\mathbf{H})_n = \log \boldsymbol{p}_v + \frac{e^{\alpha_v}}{e^{\alpha_v} + M}\boldsymbol{\beta} + \sum_{k=1}^V \frac{M_k \xi_k}{e^{\alpha_v} + M} \cdot \boldsymbol{e}_k. \tag{9}$$

Therefore, for the non-trigger token v, the cross-entropy loss between the true Markov transition $\boldsymbol{p}_v$ and the predicted transition $\mathsf{SoftMax}(\mathrm{TF}(\mathbf{H})_n)$ is given by

$$\mathsf{loss}_v(\alpha_v, \boldsymbol{\beta}) = \sum_{k=1}^{V} p_{vk} \Big\{ \log \Big[\sum_{i=1}^{V} p_{vi} \exp\Big(\frac{e^{\alpha_v}\beta_i + M_i\xi_i}{e^{\alpha_v} + M} \Big) \Big] - \frac{e^{\alpha_v}\beta_k + M_k\xi_k}{e^{\alpha_v} + M} - \log p_{vk} \Big\}. \tag{10}$$

For simplicity, we neglect the loss on trigger tokens and assume that $(\{M_i\}_{i\in[V]}, M)$ remain fixed across different positions in the input sequences.[3] We then 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

$$\mathsf{loss}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \textstyle\sum_{v\in\mathcal{V}\setminus\mathcal{T}} \pi_v \cdot \mathsf{loss}_v(\alpha_v, \boldsymbol{\beta}). \tag{11}$$

We assume that $\boldsymbol{\xi}$ and λ are fixed, and that $\boldsymbol{\alpha}$ (the attention logits of the $\langle \mathsf{s} \rangle$ token) and $\boldsymbol{\beta}$ (the value state norms of the $\langle \mathsf{s} \rangle$ token) are trainable variables, as we are interested in the dynamics of the attention logits and value state norm for the $\langle \mathsf{s} \rangle$ token. The following theorem illustrates the logarithmic growth of the 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* $\pi_v > 0$ *for any* $v \in \mathcal{V}$, *and* $\{M_i \cdot \xi_i\}_{i\in\mathcal{V}}$ *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*

$$\boldsymbol{\alpha}(t) = \tfrac{1}{2}\log t \cdot \mathbf{1} + \boldsymbol{r}(t). \tag{12}$$

(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* $\overline{B} = V^{-1}[\sum_v M_v\xi_v]$. *Consider the gradient flow over* $\boldsymbol{\beta}$. *As* $t \to \infty$, *we have*

$$\boldsymbol{\beta}(t) \to \boldsymbol{\beta}^\star = [\overline{\beta}(0) + e^{-\alpha}\overline{B}] \cdot \mathbf{1} - e^{-\alpha} \cdot \boldsymbol{M} \circ \boldsymbol{\xi}. \tag{13}$$

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

$$\boldsymbol{\alpha} = \alpha \cdot \mathbf{1}, \quad \boldsymbol{\beta} = c \cdot \mathbf{1} - e^{-\alpha} \cdot \boldsymbol{M} \circ \boldsymbol{\xi}, \quad \alpha, c \in \mathbb{R} \tag{14}$$

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

The proof of Theorem 2 is provided in Appendix 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 [52, 71]; (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 $\langle \mathsf{s} \rangle$ token are small and constant, $\boldsymbol{\beta} = \beta \cdot \mathbf{1}$, Theorem 2(a) shows that the attention logits on the $\langle \mathsf{s} \rangle$ 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 $\langle \mathsf{s} \rangle$ 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 $\langle \mathsf{s} \rangle$ 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).

[3] We note that [48] makes a similar simplification in analyzing induction heads.

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

Based on these theoretical insights, we summarize the dynamical mechanism underlying attention sinks and value-state drains: 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. This phenomenon is induced by the mutual reinforcement mechanism, as described below:

Claim 2 (Mutual reinforcement mechanism)**.** *Dynamically, attention sinks and value-state drains arise through mutual reinforcement:*

(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.

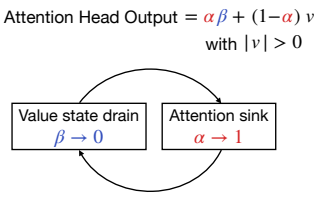


Figure 6: Mutual reinforcement mechanism

Experimental verification of the quantitative prediction. Revisiting Figure 4b, which illustrates the dynamics of a single-layer transformer model trained with Adam on the BB task, we observe that $\Delta\text{logit}_{\cdot,\langle \mathtt{s} \rangle}$ exhibits growth rates consistent with Theorem 2. In this context, $\Delta\text{logit}_{\cdot,\langle \mathtt{s} \rangle}$ 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 $\Delta\text{logit}_{\cdot,\langle \mathtt{s} \rangle}$ 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.

To further validate that Theorem 2 accurately captures the dynamics of the original model, we constructed a simplified model based on Eq. (6), (7), and (8), 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 closely resemble those of the one-layer transformer, also displaying the mutual reinforcement mechanism. A detailed description of the experiment can be found 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 $\langle \mathtt{s} \rangle$ 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 7a presents the training dynamics on the BB task using ReLU instead of SoftMax attention, showing that both the Bigram and Backcopy risk converge

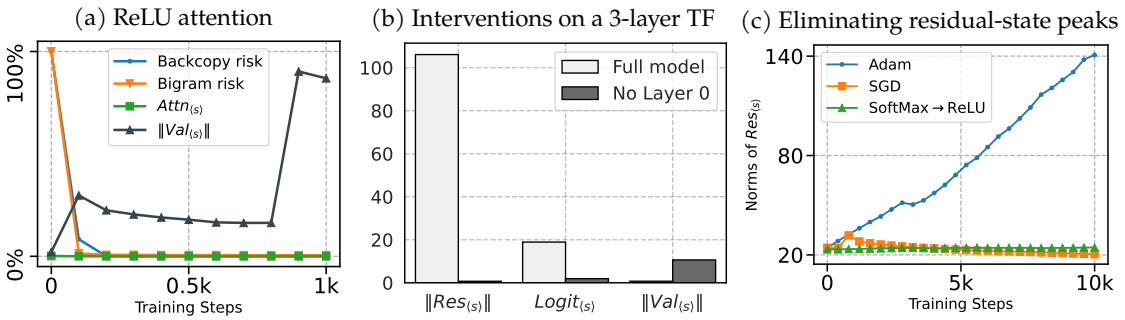


Figure 7: *Left* (*a*): The training dynamics of the single-layer ReLU attention transformer on the BB task. *Middle* (*b*): The intervention results on the `attn+mlp+attn+mlp+mlp` architecture. The attention sink and value-state peak of the middle `attn` layer disappear after zeroing out `attn` + `mlp` 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 the Bayes risk after 200 training steps, but the attention logits of $\langle \mathtt{s} \rangle$ 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. To investigate the relationship between massive residual states and attention sinks, we train on the BB task using the "`attn+mlp+attn+mlp+mlp`" model, which is the minimal structure that shows the massive residual states phenomena. We perform intervention 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 $\|\mathtt{Res}_{\langle \mathtt{s} \rangle}\|$ and $\mathrm{Mean}_v[\|\mathtt{Res}_v\|]$ at the layer 0 output, and compute $\mathrm{logit}_{\cdot,\langle \mathtt{s} \rangle}$ and $\|\mathtt{Val}_{\langle \mathtt{s} \rangle}\|$ 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.

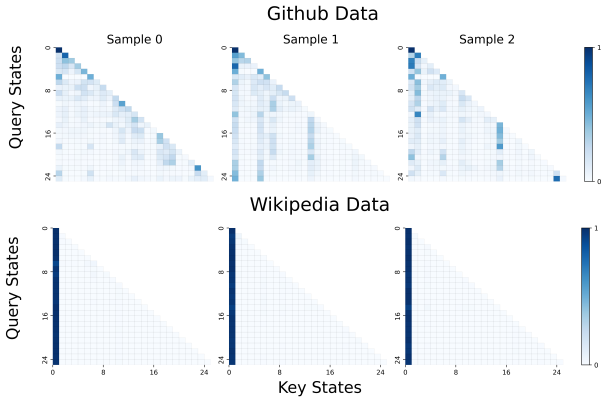
Linear growth of residual-state norm with Adam training. Figure 7c shows the residual-state norms of the $\langle \mathtt{s} \rangle$ token at the layer 0 output of three-layer transformers during pre-training on the BB task. The results indicate that training the transformer with Adam leads to a linear increase in residual state norms.

Switching from Adam to SGD and switching from SoftMax to ReLU attention eliminates the residual-state peaks. Figure 7c 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 [55], confirming the existence of the *active-dormant mechanism* in LLMs. Notably, we identify a specific head that is

(a) Attention weights for GitHub/Wikipedia data



(b) Zero-out-head intervention outcomes

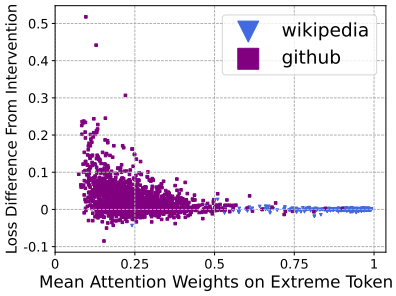


Figure 8: **Active-dormant mechanism of Layer 16 Head 25 (L16H25) of Llama 2-7B-Base.** We observe that L16H25 is active on GitHub data and dormant on Wikipedia data, both sourced from RedPajama-1T [10]. *Left* (*a*): Attention weights of L16H25, prompted by three randomly selected samples from each 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.

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 [23]. We show that the attention logits, value states norm, and residual states norm 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 8, we display a particular attention head—Layer 16 Head 25 (L16H25) of Llama 2-7B-Base [55]—which demonstrates a clear active-dormant distinction across two distinct contexts (e.g., tokens from the GitHub subset versus the Wikipedia subset of RedPajama [10]). 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 8a 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 8b compares the loss difference when L16H25 is zeroed out for prompts from both domains. The results show that zeroing out this head significantly decreases model performance on GitHub sequences, while having minimal impact on Wikipedia

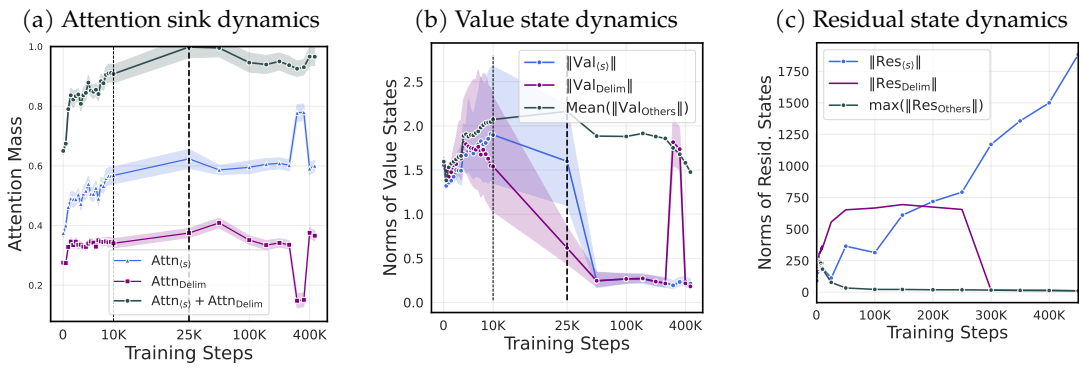


Figure 9: **Attention weights, value state norms, and residual state norms of Layer 24 during the training dynamics of OLMo.** *Left* (*a*): The total attention mass on extreme tokens ⟨s⟩ and "Delim"(⟨period⟩) at Layer 24, averaged across all attention heads. The horizontal axis is logarithmically scaled after step 10k. 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. The horizontal axis is logarithmically scaled after step $10k$. 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 4b under the BB task. *Right* (*c*): The residual state norms of each token at Layer 24 during training. The residual state norm of ⟨s⟩ increases linearly in magnitude throughout training, matching Figure 7c in the BB task.

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

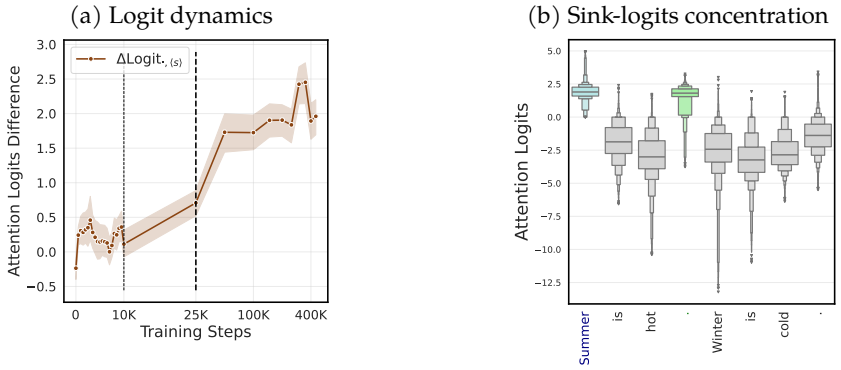


Figure 10: **Attention logits of Layer 24.** *Left* (*a*): Attention logits difference of all tokens' query states against ⟨s⟩'s key state during training. The difference in attention logits is computed as $\Delta\text{logit}_{\cdot,\langle \texttt{s}\rangle} = \texttt{Qry}_{\cdot}^{\top}\texttt{Key}_{\langle \texttt{s}\rangle} - \text{Mean}[\texttt{Qry}_{\cdot}^{\top}\texttt{Key}_{\text{Others}}]$. The horizontal axis is logarithmically scaled after step $10k$. We observe that $\Delta\text{logit}_{\cdot,\langle \texttt{s}\rangle}$ increases approximately in logarithmic scale during training steps 10k to 100k, matching the decreasing phase of the value states in Figure 9b. *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 $\text{logit}_{\cdot,v} = \texttt{Qry}_{\text{test}}^{\top}\texttt{Key}_{v}$ across all heads at Layer 24 and all test tokens. The distribution of $\text{logit}_{\cdot,\langle \texttt{s}\rangle}$ and $\text{logit}_{\cdot,\text{Delim}}$ have considerably small variance compared with other logits, confirming the sink-logits concentration phenomenon.

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 [23] (henceforth just referred to as OLMo), which provides open-sourced weights at various stages of their training.[4] 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).[5] The input we use for this analysis is again "Summer is warm⟨period⟩ Winter is cold⟨period⟩"[6] In this prompt, the "Delim" token, namely "⟨period⟩", also becomes a sink token along with ⟨s⟩. We believe this occurs because the period is not semantically meaningful and is not useful for predicting future tokens (cf. Appendix G.2)

Figure 9 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 (⟨s⟩ and Delim) increases rapidly at the beginning of training before stablizing, while the value state norms of these extreme tokens decrease during training steps 10k-100k. The synchronized evolution of attention weights and value state norms aligns with the prediction of the mutual reinforcement mechanism. Additionally, the residual states of ⟨s⟩ increase linearly, while those of other tokens converge to a small number. Figure 10 provides a more detailed examination of the attention logits in Layer 24 of OLMo. Figure 10a presents the dynamics of the difference in attention logits, showing that $\Delta\text{logit}_{\cdot,\langle s\rangle}$ increase during training steps 10k-100k, matching the decreasing phase of the value states. Figure 10b also demonstrates the *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 4.

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

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

[5] 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.

[6] Note that OLMo does not have a specific reserved token to denote the beginning of a string, but attention sinks still form in the majority of heads. In these heads, the initial token behaves as an attention sink, so in this section we denote it by ⟨s⟩ for consistency with Section 2 (cf. the remark at the end of Section 1.2) We discuss this further in Appendix G.2.

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.

References

[1] 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.

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

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

[4] 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.

[5] 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.

[6] Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Understanding and overcoming the challenges of efficient transformer quantization. *arXiv preprint arXiv:2109.12948*, 2021.

[7] Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems*, 36:75067–75096, 2023.

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

[9] 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.

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

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

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

[13] Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: k-bit inference scaling laws. In *International Conference on Machine Learning*, pages 7750–7774. PMLR, 2023.

[14] 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.

[15] 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.

[16] 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.

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

[18] Angela Fan, Pierre Stock, Benjamin Graham, Edouard Grave, Rémi Gribonval, Herve Jegou, and Armand Joulin. Training with quantization noise for extreme model compression. *arXiv preprint arXiv:2004.07320*, 2020.

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

[20] 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.

[21] 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.

[22] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC, 2022.

[23] 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.

[24] Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*, 2024.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] Jerry Yao-Chieh Hu, Pei-Hsuan Chang, Robin Luo, Hong-Yu Chen, Weijian Li, Wei-Po Wang, and Han Liu. Outlier-efficient hopfield layers for large transformer-based models. *arXiv preprint arXiv:2404.03828*, 2024.

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

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

[33] 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.

[34] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pages 795–811. Springer, 2016.

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

[36] Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *arXiv preprint arXiv:2406.01721*, 2024.

[37] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024.

[38] Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. *arXiv preprint arXiv:2310.08566*, 2023.

[39] Ye Lin, Yanyang Li, Tengbo Liu, Tong Xiao, Tongran Liu, and Jingbo Zhu. Towards fully 8-bit integer inference for the transformer model. *arXiv preprint arXiv:2009.08034*, 2020.

[40] 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.

[41] 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.

[42] 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.

[43] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

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

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

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

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

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

[53] 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*, 2023.

[54] 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.

[55] 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.

[56] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

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

[58] 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.

[59] 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.

[60] Xiaoxia Wu, Cheng Li, Reza Yazdani Aminabadi, Zhewei Yao, and Yuxiong He. Understanding int4 quantization for language models: latency speedup, composability, and failure cases. In *International Conference on Machine Learning*, pages 37524–37539. PMLR, 2023.

[61] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR, 2023.

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

[63] Z Yao, RY Aminabadi, M Zhang, X Wu, C Li, and Y Zeroquant He. Efficient and affordable post-training quantization for large-scale transformers, 2022. *URL https://arxiv.org/abs/2206.01861*.

[64] Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183, 2022.

[65] 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.

[66] Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. Q8bert: Quantized 8bit bert. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition* (*EMC2-NIPS*), pages 36–39. IEEE, 2019.

[67] 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.

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

[69] 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.

[70] 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.

[71] 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.

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

Contents

1. Introduction **1**
1.1. Related work . . . 3
1.2. Preliminaries and notations . . . 4

2. Extreme-token Phenomena in the Bigram-Backcopy Task **5**
2.1. One-layer transformer exhibits attention sinks and value-state drains . . . 5
2.2. Analysis of a minimally-sufficient transformer architecture . . . 7
2.3. The emergence of residual-state peaks . . . 11

3. Extreme-token Phenomena in pretrained LLMs **11**
3.1. Active-dormant mechanism in LLMs . . . 12
3.2. Extreme-token phenomena along training dynamics of LLMs . . . 13

4. Conclusions **14**

A Proofs of Theorem 1 and 2 **21**
A.1 Proof of Theorem 1 . . . 23
A.2 Proof of Theorem 2(c): Stable phase . . . 24
A.3 Proof of Theorem 2(a): Attention sinks . . . 28
A.4 Proof of Theorem 2(b): Value-state drains . . . 29

B The Linear Growth of the Residual States **31**
B.1. The minimal model structure to recapitulate residual state peak . . . 31
B.2. Additional plots for the three-layer transformer trained on BB task . . . 31
B.3. Potential mechanism for linear growth of the residual state peak in multi-layer models 31

C Ablations **34**
C.1 Experimental details . . . 34
C.2 Additional attention plots of a 1-layer transformer trained on the BB task . . . 34
C.3 Statics and dynamics of the simplified model in Theorem 2 . . . 34
C.4 The Bigram-Backcopy task without the $\langle \mathtt{s} \rangle$ token. . . . 34

D More Attention Heads in Dormant and Active Phase **36**

E Fine-Grained Static Mechanisms for Extreme-Token Phenomena **37**

F. Extreme-Token Phenomena Over Many Samples **39**

G Assorted Caveats **41**
G.1 Multiple attention sinks vs. one attention sink . . . 41
G.2 The role of a fixed $\langle \mathtt{s} \rangle$ token in the Active-Dormant mechanism . . . 41

A. Proofs of Theorem 1 and 2

We introduce new notations that are frequently used in the proofs. Recall that in Eq. (11), we used $\{\pi_v\}_{v\in\mathcal{V}}$ to denote the stable distribution across all tokens. We further define the stable distribution excluding trigger tokens as follows:

$$\tilde{\boldsymbol{\pi}} \in \mathbb{R}^V, \quad \tilde{\pi}_i = \pi_i \mathbf{1}\{i \in \mathcal{V} \setminus \mathcal{T}\}. \tag{15}$$

Section 2.2 defines the bigram transition probability in the Bigram-Backcopy task as $p_{vk} = \mathsf{P}(\mathsf{k} \mid \mathsf{v})$. We further define the bigram transition probability matrix as

$$\mathbf{P} = \begin{pmatrix} p_{11} & \dots & p_{1V} \\ \vdots & \ddots & \vdots \\ p_{V1} & \dots & p_{VV} \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_1^\top \\ \vdots \\ \boldsymbol{p}_V^\top \end{pmatrix}. \tag{16}$$

Given a token v, define the predicted probability at token v as the logit output passed through the softmax activation. Let $\mathbf{H} = [\langle \mathsf{s} \rangle; v_{1:n-1}; v]$. Using the form of $\mathrm{TF}(\mathbf{H})_n$ defined in Eq. (9), we denote

$$\boldsymbol{l}_v = \mathsf{SoftMax}(\mathrm{TF}(\mathbf{H})_n) = (l_{v1}, \dots, l_{vV}), \quad \text{with} \quad l_{vi} = \frac{p_{vi} \exp\left[\frac{M_i \xi_i + e^\alpha \beta_i}{e^\alpha + M}\right]}{\sum_{k=1}^V p_{vk} \exp\left[\frac{M_k \xi_k + e^\alpha \beta_k}{e^\alpha + M}\right]}. \tag{17}$$

Similar to Eq. (16), we define the full output probability matrix as

$$\mathbf{L} = \begin{pmatrix} l_{11} & \dots & l_{1V} \\ \vdots & \ddots & \vdots \\ l_{V1} & \dots & l_{VV} \end{pmatrix} = \begin{pmatrix} \boldsymbol{l}_1^\top \\ \vdots \\ \boldsymbol{l}_V^\top \end{pmatrix}. \tag{18}$$

Using the notation $\boldsymbol{l}_v$ and $\tilde{\pi}_v$, we can rewrite the loss functions defined in Eq. (10) and Eq. (11) as follows:

$$\mathsf{loss}_v(\alpha_v, \boldsymbol{\beta}) = -\sum_{k=1}^V p_{vk} \log l_{vk}, \quad \mathsf{loss}_v(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{v=1}^V \tilde{\pi}_v \mathsf{loss}_v(\alpha_v, \boldsymbol{\beta}). \tag{19}$$

We always have that $\sum_k p_{vk} = 1$ and $\sum_k l_{vk} = 1$. The total variation norm and KL-divergence are then defined as:

$$\|\boldsymbol{p}_v - \boldsymbol{l}_v\|_{\mathrm{TV}} = \sum_k |p_{vk} - l_{vk}|, \quad \mathrm{KL}(\boldsymbol{p}_v \,||\, \boldsymbol{l}_v) = -\sum_k p_{vk} \log(l_{vk}/p_{vk}). \tag{20}$$

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

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

Given any $\boldsymbol{p}_v$ defined in Eq. (16), denote

$$\mathbf{G}_v^{\mathbf{P}} = \mathrm{diag}(\boldsymbol{p}_v) - \boldsymbol{p}_v \boldsymbol{p}_v^\top, \quad \mathbf{G}_v^{\mathbf{L}} = \mathrm{diag}(\boldsymbol{l}_v) - \boldsymbol{l}_v \boldsymbol{l}_v^\top. \tag{21}$$

We now present technical lemmas concerning $\mathbf{G}_v^{\mathbf{P}}$ and $\mathbf{G}_v^{l}$.

Lemma A.1. *The matrices $\mathbf{G}_v^{\mathbf{P}} \in \mathbb{R}^{V\times V}$ and $\mathbf{G}_v^{\mathbf{L}} \in \mathbb{R}^{V\times V}$ are positive semi-definite for any $v \in \mathcal{V}$.*

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

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

$$(\mathbf{G}_v^{\mathbf{L}})_{ii} = l_i - l_i^2 = l_i(\sum_{k\neq i} l_k) \geq \sum_{k\neq i} |(\mathbf{G}_v^{\mathbf{L}})_{ik}|.$$

This shows that both $\mathbf{G}_v^{\mathbf{P}}$ and $\mathbf{G}_v^{\mathbf{L}}$ are diagonally dominant matrices. By Corollary 6.2.27 in Horn and Johnson [29], they are positive semi-definite. □

Lemma A.2. *Suppose that $\tilde{\pi}_v > 0$ for any $v \in \mathcal{V} \setminus \mathcal{T}$. For any $\boldsymbol{\eta} \in \mathbb{R}^V$ with $\boldsymbol{\eta} \perp \mathbf{1}$, there exists $\omega > 0$ such that*

$$\boldsymbol{\eta}^\top \Big[\sum_{k=1}^{V} \tilde{\pi}_k \mathbf{G}_k^{\mathbf{P}} \Big] \boldsymbol{\eta} \geq \omega \|\boldsymbol{\eta}\|_2^2.$$

Proof of Lemma A.2. Denote the null spaces of $\mathbf{G}_v^{\mathbf{P}}$ for $v \in \mathcal{V}$ as $\mathsf{S_v}$. We solve for each $\mathsf{S_v}$. Setting $\mathbf{G}_v^{\mathbf{P}} \boldsymbol{\eta} = 0$ gives that

$$[p_{vj} - p_{vj}(\sum_k p_{vk})]\eta_j = 0 \text{ for any } j \in \mathcal{V}.$$

If $p_{vj} \neq 0$, we divide each side with p_{vj} and get that $\eta_j = \sum_k p_{vk}\eta_k$. As a result, we get that

$$\mathsf{S}_v = \{\boldsymbol{\eta} \mid \eta_j \text{ is constant for } p_{vj} \neq 0\}.$$

Since all $\pi_k > 0$, for any $k \in \mathcal{V} \setminus \mathcal{T}$, there is $v \in \mathcal{V} \setminus \mathcal{T}$ such that $p_{vk} > 0$, we get that

$$\cap_{v \in \mathcal{V} \setminus \mathcal{T}} \mathsf{S}_v = \{c \cdot \mathbf{1} \mid c \in \mathbb{R}\}.$$

Since $\boldsymbol{\eta} \perp \mathbf{1}$, we get that $\boldsymbol{\eta} \perp \cap_{v \in \mathcal{V} \setminus \mathcal{T}} \mathsf{S_v}$. We denote the minimal non-zero eigenvalues of $\mathbf{G}_v^{\mathbf{L}}$ for $v \in \mathcal{V} \setminus \mathcal{T}$ as λ. We get that

$$\boldsymbol{\eta}^\top \Big[\sum_{k=1}^{V} \tilde{\pi}_k \mathbf{G}_k^{\mathbf{P}} \Big] \boldsymbol{\eta} \geq \Big[\min_{v \in \mathcal{V} \setminus \mathcal{T}} \tilde{\pi}_v \Big] \lambda \|\boldsymbol{\eta}\|_2^2.$$

Setting $\omega = \lambda \cdot \min_{v \in \mathcal{V} \setminus \mathcal{T}} \tilde{\pi}_v > 0$, this proves Lemma A.2. □

Lemma A.3. *Given ω defined in Lemma A.2, suppose that*

$$\max_{v,k} |p_{vk} - l_{vk}| = \delta \leq \min\{\omega/(6V), 1\}. \tag{22}$$

For any $\boldsymbol{\eta} \in \mathbb{R}^V$ with $\boldsymbol{\eta} \perp \mathbf{1}$, we have that

$$\boldsymbol{\eta}^\top \Big[\sum_{k=1}^{V} \tilde{\pi}_k \mathbf{G}_k^{\mathbf{L}} \Big] \boldsymbol{\eta} \geq \frac{\omega}{2} \|\boldsymbol{\eta}\|_2^2.$$

Proof of Lemma A.3. Denote $\delta = \max_{v,k} |p_{vk} - l_{vk}|$. Suppose that $\delta \leq 1$. For any $k \in \mathcal{V} \setminus \mathcal{T}$, we can verify that

$$\Big| (\mathbf{G}_k^{\mathbf{P}})_{ij} - (\mathbf{G}_k^{\mathbf{L}})_{ij} \Big| \leq 3\delta,$$

for any $i, j \in [V]$. We denote

$$\mathbf{E} = \sum_{k=1}^{V} \tilde{\pi}_k \mathbf{G}_k^{\mathbf{P}} - \sum_{k=1}^{V} \tilde{\pi}_k \mathbf{G}_k^{\mathbf{L}}.$$

Therefore, $|\mathbf{E}_{ij}| \leq 3\delta$ for any $i, j \in [V]$. This means that

$$\boldsymbol{\eta}^\top \mathbf{E} \boldsymbol{\eta} \leq \|\mathbf{E}\|_2 \|\boldsymbol{\eta}\|_2^2 \leq \|\mathbf{E}\|_F \|\boldsymbol{\eta}\|_2^2 \leq V \cdot 3\delta \cdot \|\boldsymbol{\eta}\|_2^2.$$

As a result, when $\delta \leq \min\{\omega/(6V), 1\}$, we get that

$$\boldsymbol{\eta}^\top \Big[\sum_{k=1}^{V} \tilde{\pi}_k \mathbf{G}_k^{\mathbf{L}} \Big] \boldsymbol{\eta} \geq \omega \|\boldsymbol{\eta}\|_2^2 - \boldsymbol{\eta}^\top \mathbf{E} \boldsymbol{\eta} \geq \frac{\omega}{2} \|\boldsymbol{\eta}\|_2^2.$$

This proves Lemma A.3. □

A.1. Proof of Theorem 1

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

Assumption A. *We have* $\{\texttt{emb}_0(\langle s \rangle)\} \cup \{\texttt{emb}_i(v)\}_{i \in \{0\} \cup [N-1], v \in \mathcal{V}} \subseteq \mathbb{R}^d$*, where the embedding dimension* $d \geq VN + 1$.

Assumption A requires a large embedding dimension $d \geq VN + 1$. This assumption is used to ensure that there are enough orthonormal bases in the embedding space. Given the fact that there are $\mathrm{O}(\exp(d))$ approximately linearly independent vectors for large d [57], it is possible to relax the assumption to be $d \gg \log(VN)$. 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.

Theorem A.4 (Formal statement of Theorem 1)**.** *Let Assumption A hold. 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* (5) *with weight matrices* $(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{W}_1, \mathbf{W}_2)$ *such that Eq.* (6)*,* (7)*, and* (8) *hold. Consider the Bigram-Backcopy task, where given an input* $\mathbf{H} = [\langle s \rangle; v_{1:n-1}, v]$*, the ground-truth transition gives* $\mathsf{P}(v' \mid \mathbf{H}) = p_{vv'}$ *for* $v \in \mathcal{V} \setminus \mathcal{T}$*, and* $\mathsf{P}(v' \mid \mathbf{H}) = 1\{v' = v_{n-1}\}$ *for* $v \in \mathcal{T}$*. There exists a sequence* $\min_{v \in \mathcal{V}} \alpha_v \to \infty$, $\min_{v \in \mathcal{V}} \xi_v \to \infty$, $\lambda \to \infty$*, and* $\boldsymbol{\beta} = \mathbf{0}$ *such that this transformer generates the ground-truth transition in the limit, i.e.,*

$$\mathsf{SoftMax}(\mathrm{TF}(\mathbf{H})_n) \to \mathsf{P}(\cdot \mid \mathbf{H}). \tag{23}$$

Proof of Theorem A.4.

Step 1. Construction for the attention head. We let $\{\texttt{emb}_0(\langle \mathtt{s} \rangle)\} \cup \{\texttt{emb}_i(v)\}_{i \in \{0\} \cup [N-1], v \in \mathcal{V}} \cup \{\boldsymbol{e}_v\}_{v \in \mathcal{V}}$ to be a set of orthonormal basis in $\mathbb{R}^d$, and denote $\{\boldsymbol{\eta}_i\}_{i \in \{0\} \cup [N-1]} \subseteq \mathbb{R}^d$ by a set of orthonormal basis in $\mathbb{R}^d$ (the existence is guaranteed by Assumption A). Therefore, 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 query matrix $\mathbf{Q} \in \mathbb{R}^{d \times N}$ such that

$$\begin{aligned} \mathbf{Q} \cdot \texttt{emb}_i(v) &= \lambda \boldsymbol{\eta}_{i-1} \quad \text{for } i > 1, \quad v \in \mathcal{T}, \\ \mathbf{Q} \cdot \texttt{emb}_i(v) &= \alpha_v \boldsymbol{\eta}_0 \quad \text{for } i > 0, \quad v \in \mathcal{V} \setminus \mathcal{T}. \end{aligned} \tag{24}$$

Meanwhile, there is a key matrix $\mathbf{K} \in \mathbb{R}^{d \times N}$ such that

$$\begin{aligned} \mathbf{K} \cdot \texttt{emb}_i(v) &= \boldsymbol{\eta}_i \quad \text{for } i > 0, \quad v \in \mathcal{V}, \\ \mathbf{K} \cdot \texttt{emb}_0(\langle \mathtt{s} \rangle) &= \boldsymbol{\eta}_0. \end{aligned} \tag{25}$$

Denote $\{\boldsymbol{e}_v\}_{v \in \mathcal{V}}$ as an orthonormal basis in $\mathbb{R}^V$. There is a matrix $\mathbf{V} \in \mathbb{R}^{d \times V}$ such that

$$\begin{aligned} &\mathbf{V} \cdot \texttt{emb}_i(v) = \xi_v \boldsymbol{e}_v \in \mathbb{R}^V, \quad \text{with } \xi_v = 0 \text{ for } v \in \mathcal{T}, \text{ and } \xi_v \geq 0 \text{ for } v \in \mathcal{V} \setminus \mathcal{T}. \\ &\mathbf{V} \cdot \texttt{emb}_0(\langle \mathtt{s} \rangle) = \boldsymbol{\beta} \in \mathbb{R}^V. \end{aligned} \tag{26}$$

This construction matches Eq. (6) and (7).

As a result, for $v_n \in \mathcal{V} \setminus \mathcal{T}$, by Eq. (5), denoting $\mathbf{H} = [\langle \mathtt{s} \rangle; v_{1:n-1}; v_n]$ and $\texttt{attn}(\mathbf{H})_n$ to be the last column of $\texttt{attn}(\mathbf{H})$, we have

$$\begin{aligned} \texttt{attn}(\mathbf{H})_n &= \sum_{i=0}^{n} \frac{\exp[\texttt{emb}_n(v_n)^\top \mathbf{Q}^\top \mathbf{K} \cdot \texttt{emb}_i(v_i)] \mathbf{V} \cdot \texttt{emb}_i(v_i)}{\sum_{j=0}^{n} \exp[\texttt{emb}_n(v_n)^\top \mathbf{Q}^\top \mathbf{K} \cdot \texttt{emb}_j(v_j)]} \\ &= \frac{\exp[\alpha_{v_n} \boldsymbol{\eta}_0^\top \boldsymbol{\eta}_0] \cdot \boldsymbol{\beta} + \sum_{i=1}^{n} \left\{ \exp[\alpha_{v_n} \boldsymbol{\eta}_0^\top \boldsymbol{\eta}_i] \xi_{v_i} \cdot \boldsymbol{e}_{v_i} \right\}}{\exp[\alpha_{v_n} \boldsymbol{\eta}_0^\top \boldsymbol{\eta}_0] + \sum_{j=1}^{n} \exp[\alpha_{v_n} \boldsymbol{\eta}_0^\top \boldsymbol{\eta}_j]} \\ &= \frac{e^{\alpha_{v_n}}}{e^{\alpha_{v_n}} + n} \cdot \boldsymbol{\beta} + \sum_{i=1}^{n} \frac{1}{e^{\alpha_{v_n}} + n} \cdot \xi_{v_i} \boldsymbol{e}_{v_i}. \end{aligned}$$

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

$$\texttt{attn}(\mathbf{H})_n = \sum_{i=0}^{n} \frac{\exp[\texttt{emb}_n(v_n)^\top \mathbf{Q}^\top \mathbf{K} \cdot \texttt{emb}_i(v_i)] \mathbf{V} \cdot \texttt{emb}_i(v_i)}{\sum_{j=0}^{n} \exp[\texttt{emb}_n(v_n)^\top \mathbf{Q}^\top \mathbf{K} \cdot \texttt{emb}_j(v_j)]}$$

$$= \frac{\exp[\lambda \boldsymbol{\eta}_{n-1}^\top \boldsymbol{\eta}_0] \cdot \boldsymbol{\beta} + \sum_{i=1}^n \{\exp[\lambda \boldsymbol{\eta}_{n-1}^\top \boldsymbol{\eta}_i] \xi_{v_i} \cdot \boldsymbol{e}_{v_i}\}}{\exp[\lambda \boldsymbol{\eta}_{n-1}^\top \boldsymbol{\eta}_0] + \sum_{j=1}^n \exp[\lambda \boldsymbol{\eta}_{n-1}^\top \boldsymbol{\eta}_j]}$$
$$= \frac{1}{e^\lambda + n} \cdot \boldsymbol{\beta} + \sum_{i \neq n-1} \frac{1}{e^\lambda + n} \cdot \xi_{v_i} \boldsymbol{e}_{v_i} + \frac{e^\lambda}{e^\lambda + n} \cdot \xi_{v_{n-1}} \boldsymbol{e}_{v_{n-1}}.$$

Step 2. Construction for the MLP layer. Further, define the weights for the `mlp` layer such that

$$\mathbf{W}_1 \cdot \mathtt{emb}_i(v) = \boldsymbol{e}_v \in \mathbb{R}^V, \quad \mathbf{W}_2 \boldsymbol{e}_v = \log \boldsymbol{p}_v \cdot 1\{v \notin \mathcal{T}\} \in \mathbb{R}^V \quad \text{for } i \in [N], \ \ v \in \mathcal{V}, \tag{27}$$

where $\{\boldsymbol{e}_v\}$ is the eorthonormal basis in $\mathbb{R}^V$ and $\boldsymbol{p}_v \in \mathbb{R}^V$ is defined in Eq. (16). As a result, $\mathtt{mlp}(\mathbf{H})_n = \mathbf{W}_2 \mathrm{ReLU}(\mathbf{W}_1 \mathtt{emb}_n(v)) = \mathbf{W}_2 \boldsymbol{e}_v = \log \boldsymbol{p}_v \cdot \mathbf{1}\{v \notin \mathcal{T}\}$. This matches the Eq. (8).

Step 3. The output of the transformer. By Eq. (5) again, on non-trigger token $v \in \mathcal{V} \setminus \mathcal{T}$, the transformer output gives that

$$\begin{aligned} \mathrm{TF}(\mathbf{H})_n &= \mathtt{mlp}(\mathtt{emb}_n(v)) + \mathtt{attn}(\mathbf{H})_n \\ &= \log \boldsymbol{p}_v + \frac{e^{\alpha_{v_n}}}{e^{\alpha_{v_n}} + n} \cdot \boldsymbol{\beta} + \sum_{i=0}^n \frac{1}{e^{\alpha_{v_n}} + n} \cdot \xi_{v_i} \boldsymbol{e}_{v_i}. \end{aligned}$$

On trigger token $v \in \mathcal{T}$, the transformer output gives that

$$\begin{aligned} \mathrm{TF}(\mathbf{H})_n &= \mathtt{mlp}(\mathtt{emb}_n(v)) + \mathtt{attn}(\mathbf{H})_n \\ &= \frac{1}{e^\lambda + n} \cdot \boldsymbol{\beta} + \sum_{i \neq n-1} \frac{1}{e^\lambda + n} \cdot \xi_{v_i} \boldsymbol{e}_{v_i} + \frac{e^\lambda}{e^\lambda + n} \cdot \xi_{v_{n-1}} \boldsymbol{e}_{v_{n-1}}. \end{aligned}$$

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

$$\mathsf{SoftMax}[\mathrm{TF}(\mathbf{H})_n] \to \boldsymbol{p}_{v_n} \quad \text{for } n > 0, \ \ v_n \in \mathcal{V} \setminus \mathcal{T},$$
$$\mathsf{SoftMax}[\mathrm{TF}(\mathbf{H})_n] \to (1\{v = v_{n-1}\})_{v \in \mathcal{V}} \quad \text{for } n > 0, \ \ v_n \in \mathcal{T}.$$

This proves Eq. (23), indicating that the transformer output matches the ground truth transition. This finishes the proof of Theorem A.4. □

A.2. Proof of Theorem 2(c): Stable phase

We first state Lemma A.5 and Proposition A.6 that are used to prove Theorem 2(c). Lemma A.5 computes the gradients of l_{ik} as defined in Eq. (17).

Lemma A.5. *Given l_{ik} defined in Eq. (17), for any i, k, v, and any value of α_v and β_v, we have that*

$$\frac{\partial l_{ik}}{\partial \alpha_v} = \frac{\mathbf{1}\{i = v\} l_{ik} e^{\alpha_i}}{(e^{\alpha_i} + M)^2} \Big[M\beta_k - M_k \xi_k - \sum_{j=1}^V l_{ij} (M\beta_j - M_j \xi_j) \Big],$$
$$\frac{\partial l_{ik}}{\partial \beta_v} = \frac{e^{\alpha_i}}{e^{\alpha_i} + M} [l_{ik} \mathbf{1}\{k = v\} - l_{ik} l_{iv}].$$

Furthermore, we have

$$\sum_{k=1}^V \frac{\partial l_{ik}}{\partial \alpha_v} = 0 \ \ \textit{for any } i, \ v, \ \boldsymbol{\alpha}, \textit{ and } \boldsymbol{\beta}, \quad \sum_{v=1}^V \frac{\partial l_{ik}}{\partial \beta_v} = 0 \ \ \textit{for any } i, \ k, \ \boldsymbol{\alpha}, \textit{ and } \boldsymbol{\beta}.$$

Proof of Lemma A.5. We repeatedly use the following two facts:

$$\frac{\partial \Big\{ \exp\Big[\frac{M_k \xi_k + e^{\alpha_i} \beta_k}{e^{\alpha_i} + M} \Big] \Big\}}{\partial \alpha_v} = \frac{\mathbf{1}\{i = v\} e^{\alpha_i} (M\beta_k - M_k \xi_k)}{(e^{\alpha_i} + M)^2} \exp\Big[\frac{M_k \xi_k + e^{\alpha_i} \beta_k}{e^{\alpha_i} + M} \Big],$$

$$\frac{\partial\Big\{\exp\Big[\frac{M_k\xi_k+e^{\alpha_i}\beta_k}{e^{\alpha_i}+M}\Big]\Big\}}{\partial\beta_v}=\frac{\mathbf{1}\{k=v\}e^{\alpha_i}}{e^{\alpha_i}+M}\exp\Big[\frac{M_k\xi_k+e^{\alpha_i}\beta_k}{e^{\alpha_i}+M}\Big].$$

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

$$\begin{aligned}\frac{\partial l_{vk}}{\partial\alpha_v}&=l_{vk}e^{\alpha_v}\Big[\frac{M\beta_k-M_k\xi_k}{(e^{\alpha_v}+M)^2}\Big]-\frac{l_{vk}\sum_{i=1}^V p_{vi}e^{\alpha_v}\Big[\frac{M\beta_i-M_i\xi_i}{(e^{\alpha_v}+M)^2}\Big]\exp\Big[\frac{M_i\xi_i+e^{\alpha_v}\beta_i}{e^{\alpha_v}+M}\Big]}{\sum_{i=1}^V p_{vi}\exp\Big[\frac{M_i\xi_i+e^{\alpha_v}\beta_i}{e^{\alpha_v}+M}\Big]}\\&=\frac{e^{\alpha_v}}{(e^{\alpha_v}+M)^2}\Big\{l_{vk}[M\beta_k-M_k\xi_k]-l_{vk}\sum_{j=1}^V l_{vj}(M\beta_j-M_j\xi_j)\Big\},\end{aligned}$$

and

$$\begin{aligned}\frac{\partial l_{ik}}{\partial\beta_v}&=\Big[\frac{e^{\alpha_i}}{e^{\alpha_i}+M}\Big]l_{ik}\mathbf{1}\{k=v\}-\frac{\Big[\frac{e^{\alpha_i}}{e^{\alpha_i}+M}\Big]p_{iv}\exp\Big[\frac{M_v\xi_v+e^{\alpha_i}\beta_v}{e^{\alpha_i}+M}\Big]p_{ik}\exp\Big[\frac{M_k\xi_k+e^{\alpha_i}\beta_k}{e^{\alpha_i}+M}\Big]}{\Big(\sum_{j=1}^V p_{ij}\exp\Big[\frac{M_j\xi_j+e^{\alpha_i}\beta_j}{e^{\alpha_i}+M}\Big]\Big)^2}\\&=\Big[\frac{e^{\alpha_i}}{e^{\alpha_i}+M}\Big][l_{ik}\mathbf{1}\{k=v\}-l_{ik}l_{iv}].\end{aligned}$$

We can verify that

$$\begin{aligned}\sum_{k=1}^V\frac{\partial l_{ik}}{\partial\alpha_v}&=\frac{e^{\alpha_v}}{(e^{\alpha_v}+M)^2}\sum_{k=1}^V\Big\{l_{vk}[M\beta_k-M_k\xi_k]-l_{vk}\sum_{j=1}^V l_{vj}(M\alpha_j-M_j\xi_j)\Big\}\\&=\frac{e^{\alpha_v}}{(e^{\alpha_v}+M)^2}\Big\{\sum_{k=1}^V l_{vk}[M\beta_k-M_k\xi_k]-\sum_{j=1}^V l_{vj}(M\alpha_j-M_j\xi_j)\Big\}\\&=0,\end{aligned}$$

and

$$\begin{aligned}\sum_{v=1}^V\frac{\partial l_{ik}}{\partial\beta_v}&=\Big[\frac{e^{\alpha_i}}{e^{\alpha_i}+M}\Big]\sum_{v=1}^V[l_{ik}\mathbf{1}\{k=v\}-l_{ik}l_{iv}]\\&=\Big[\frac{e^{\alpha_i}}{e^{\alpha_i}+M}\Big][l_{ik}-l_{ik}]\\&=0.\end{aligned}$$

This finishes the proof of Lemma A.5. □

Proposition A.6 computes the gradient of loss with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, giving the ODE of the gradient flow.

Proposition A.6. *Consider the gradient flow of optimizing* $\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})$ *given by*

$$\dot{\boldsymbol{\alpha}}(t)=-\nabla_{\boldsymbol{\alpha}}\mathsf{loss}(\boldsymbol{\alpha}(t),\boldsymbol{\beta}(t)),\quad\dot{\boldsymbol{\beta}}(t)=-\nabla_{\boldsymbol{\beta}}\mathsf{loss}(\boldsymbol{\alpha}(t),\boldsymbol{\beta}(t)).\tag{28}$$

Simplifying the dynamics using Lemma A.5 gives that

$$\begin{aligned}\dot\alpha_v(t)&=\frac{\tilde\pi_v e^{\alpha_v}}{(e^{\alpha_v}+M)^2}\sum_{i=1}^V(p_{vi}-l_{vi})(M\beta_i-M_i\xi_i),\\\dot\beta_v(t)&=\sum_{k=1}^V\Big\{\frac{\tilde\pi_k e^{\alpha_k}[p_{kv}-l_{kv}]}{e^{\alpha_k}+M}\Big\}.\end{aligned}$$

Proof of Proposition A.6. Taking the derivative of $\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})$ gives that

$$\frac{\partial\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial\alpha_v}=\tilde\pi_v\sum_{k=1}^V p_{vk}\cdot\frac{-1}{l_{vi}}\cdot\frac{\partial l_{vi}}{\partial\alpha_v}$$

$$= \frac{\tilde{\pi}_v e^{\alpha_v}}{(e^{\alpha_v}+M)^2}\Big\{\sum_{i=1}^{V} l_{vi}[M\beta_i - M_i\xi_i] - \sum_{k=1}^{V} p_{vk}[M\beta_k - M_k\xi_k]\Big\}$$
$$= \frac{\tilde{\pi}_v e^{\alpha_v}}{(e^{\alpha_v}+M)^2}\sum_{k=1}^{V}\Big\{[l_{vk}-p_{vk}][M\beta_k - M_k\xi_k]\Big\}.$$

Similarly, we have that

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

Plug them in Eq. (28) proves Proposition A.6. □

Theorem A.7 (Restatement the stable phase part in Theorem 2(c))**.** *Assume $\xi_v \geq 0$ for any v, $\pi_v > 0$ for any $v \in \mathcal{V}$, and $\{M_i \cdot \xi_i\}_{i\in\mathcal{V}}$ are not all equal. Consider the gradient flow over the variables $(\boldsymbol{\alpha},\boldsymbol{\beta})$, i.e., $(\dot{\boldsymbol{\alpha}}(t),\dot{\boldsymbol{\beta}}(t)) = -\nabla_{\boldsymbol{\alpha},\boldsymbol{\beta}}\mathsf{loss}(\boldsymbol{\alpha}(t),\boldsymbol{\beta}(t))$. Any vector of the following form*

$$\boldsymbol{\alpha}^\star = \alpha\cdot\mathbf{1}, \quad \boldsymbol{\beta}^\star = c\cdot\mathbf{1} - e^{-\alpha}\cdot\boldsymbol{M}\circ\boldsymbol{\xi}, \quad \alpha, c\in\mathbb{R} \tag{29}$$

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

Proof of Theorem A.7. When $\boldsymbol{\alpha} = \boldsymbol{\alpha}^\star$ and $\boldsymbol{\beta} = \boldsymbol{\beta}^\star$, given l_{vi} defined in Eq. (17) with any v and i, we have that

$$l_{vi} = \frac{p_{vi}\exp\left[\frac{M_i\xi_i + e^\alpha\beta_i}{e^\alpha+M}\right]}{\sum_{k=1}^{V} p_{vk}\exp\left[\frac{M_k\xi_k + e^\alpha\beta_k}{e^\alpha+M}\right]}$$
$$= \frac{p_{vi}\exp\left[\frac{e^\alpha c}{e^\alpha+M}\right]}{\sum_{k=1}^{V} p_{vk}\exp\left[\frac{e^\alpha c}{e^\alpha+M}\right]}$$
$$= p_{vi}.$$

Plug l_{vi} into $\partial\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})/\partial\boldsymbol{\alpha}$ and $\partial\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})/\partial\boldsymbol{\beta}$, we have

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

This shows that $\boldsymbol{\alpha} = \boldsymbol{\alpha}^\star$ and $\boldsymbol{\beta} = \boldsymbol{\beta}^\star$ are stationary points. We further compute the second-order derivative using Lemma A.5. To simplify the notation, we use $z_k = W\beta_k - M_k\xi_k$ and $\boldsymbol{z} = [z_1,\ldots,z_V]$. We have that

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

where in the last line, we plugged in $l_{vi} = p_{vi}$ for any v and i. Similarly, we compute the second order derivatives with respect to α_i and β_v,

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

With the same manner, we compute the second order derivatives with respect to β_i and β_v,

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

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

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

where $\mathbf{G}_k^{\mathbf{P}}$ is defined in Eq. (21). Furthermore, there exists $\boldsymbol{U}$ such that $\boldsymbol{U}\text{Hessian}(\mathsf{loss}(\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star))\boldsymbol{U}^\top = \text{Diag-Hessian}(\mathsf{loss}(\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star))$, with

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

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

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

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

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

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

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

As a result, we have that

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

This shows that $\mathbf{B}$ is positive semi-definite. Therefore, $\text{Hessian}(\mathsf{loss}(\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star))$ is positive semi-definte. This proves Theorem A.7. □

A.3. Proof of Theorem 2(a): Attention sinks

Theorem A.8 (Restatement of the attention sink part in Theorem 2(a))**.** *Assume $\xi_v \geq 0$ for any v, $\pi_v > 0$ for any $v \in \mathcal{V}$, and $\{M_i \cdot \xi_i\}_{i\in\mathcal{V}}$ are not all equal. Fix $\boldsymbol{\beta} = \beta \cdot \mathbf{1}$ for a constant β, and consider the gradient flow of the loss function $\mathsf{loss}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ over $\boldsymbol{\alpha}$, i.e., $\dot{\boldsymbol{\alpha}}(t) = -\nabla \mathsf{loss}(\boldsymbol{\alpha}(t), \boldsymbol{\beta})$. With any initial value $\boldsymbol{\alpha}(0)$, there exists $\boldsymbol{r}(t)$ with norm uniformly bounded in time, such that*

$$\boldsymbol{\alpha}(t) = \tfrac{1}{2}\log t \cdot \mathbf{1} + \boldsymbol{r}(t). \tag{30}$$

Proof of Theorem A.8. We separately analyze each entry of $\boldsymbol{\alpha}$. Focusing on α_v, to simplify the notation, we introduce a random variable φ such that

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

Denote

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

Therefore, using Lemma A.6, we get that

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

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

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

Since both $e^{x/(u+M)}$ 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

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

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

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

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

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

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 bound $\mathrm{d}u/\mathrm{d}t$ and get that

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

By solving the ODE, we get that

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

To give a lower bound, we have that

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

Therefore, $u \geq \sqrt{\tilde{C}t+\tilde{C}_2}$. In conclusion, we have that

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

with r_v bounded. This proves Theorem A.8. □

A.4. Proof of Theorem 2(b): Value-state drains

Theorem A.9 (Restatement of Theorem 2(b))**.** *Assume $\xi_v \geq 0$ for any v, $\pi_v > 0$ for any $v \in \mathcal{V}$, and $\{M_i \cdot \xi_i\}_{i\in\mathcal{V}}$ are not all equal. Fix $\boldsymbol{\alpha} = \alpha \cdot \mathbf{1}$ for a constant α, define $\overline{\beta}(0) = V^{-1}[\sum_v \beta_v(0)]$ and $\overline{B} = V^{-1}[\sum_v M_v\xi_v]$. Consider the gradient flow of the loss function $\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})$ over $\boldsymbol{\beta}$ for fixed $\boldsymbol{\alpha}$, i.e., $\dot{\boldsymbol{\beta}}(t) = -\nabla_{\boldsymbol{\beta}}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}(t))$. As $t\to\infty$, we have*

$$\boldsymbol{\beta}(t) \to \boldsymbol{\beta}^\star = [\overline{\beta}(0) + e^{-\alpha}\overline{B}]\cdot\mathbf{1} - e^{-\alpha}\cdot\boldsymbol{M}\circ\boldsymbol{\xi}. \tag{31}$$

Proof of Theorem A.9. We plug $\boldsymbol{\beta}^\star$ into the loss and get that $\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}^\star) = \sum_{v=1}^V \tilde{\pi}_v \sum_{k=1}^V p_{vk}\log p_{vk}$. Computing $\nabla^2_{\boldsymbol{\beta}}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})$, we get that

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

where $\mathbf{G}_k^{\mathbf{L}}$ is defined in Eq. (21). Lemma A.1 indicates that it is positive semi-definite. Therefore, we have that

$$\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}(t)) \to \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}^\star) \text{ as } t\to\infty.$$

We choose δ as defined in Eq.(22). When t is sufficiently large, we have that

$$\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}(t)) \leq \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}^\star) + \frac{1}{\min_{k\in\mathcal{V}\setminus\mathcal{T}}\tilde{\pi}_k}\cdot 2\delta^2.$$

The convexity further implies that for any $\tilde{\boldsymbol{\beta}} = \theta\boldsymbol{\beta}(t) + (1-\theta)\boldsymbol{\beta}^\star$ ($\theta\in(0,1)$), we have that

$$\mathsf{loss}(\boldsymbol{\alpha},\tilde{\boldsymbol{\beta}}) \leq \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}^\star) + \frac{1}{\min_{k\in\mathcal{V}\setminus\mathcal{T}}\tilde{\pi}_k}\cdot 2\delta^2.$$

Denote $\tilde{\boldsymbol{l}}_v = \boldsymbol{l}_v(\boldsymbol{\alpha},\tilde{\boldsymbol{\beta}})$ as $\boldsymbol{l}$ evaluated on $(\boldsymbol{\alpha},\tilde{\boldsymbol{\beta}})$. Using the definition of the KL-divergence in Eq. (20), we have that

$$\sum_{v=1}^V \tilde{\pi}_v KL(\boldsymbol{p}_v \,||\, \tilde{\boldsymbol{l}}_v) = \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}(t)) - \mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}^\star) \leq \frac{1}{\min_{k\in\mathcal{V}\setminus\mathcal{T}}\tilde{\pi}_k}\cdot 2\delta^2.$$

This further implies that $KL(\boldsymbol{p}_v \,||\, \tilde{\boldsymbol{l}}_v) \leq 2\delta^2$ for any v. Using Pinsker's inequality, we get that

$$\sum_{k=1}^V |p_{vk} - \tilde{l}_{vk}| = \|\boldsymbol{p}_v - \boldsymbol{l}_v\|_{\mathrm{TV}} \leq \sqrt{KL(\boldsymbol{p}_v \,||\, \tilde{\boldsymbol{l}}_v)/2} \leq \delta.$$

Therefore, $\max_{v,k}\left|p_{vk}-\tilde{l}_{vk}\right| \leq \delta$. Lemma A.5 gives that $\sum_{v=1}^{V}\dot{\beta}_v(t) = 0$. Therefore, $\sum_{v=1}^{V}\beta_v(t)/V = \overline{\beta}(0)$. The choice of $\boldsymbol{\beta}^\star$ guarantees that $\overline{\beta}^\star = \overline{\beta}(0)$. This shows that $\boldsymbol{\beta}(t)-\boldsymbol{\beta}^\star \perp \mathbf{1}$. Using Lemma A.3, there exists $\omega > 0$ such that

$$(\boldsymbol{\beta}(t)-\boldsymbol{\beta}^\star)^\top \nabla^2_{\boldsymbol{\beta}}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta})(\boldsymbol{\beta}(t)-\boldsymbol{\beta}^\star) = (\boldsymbol{\beta}(t)-\boldsymbol{\beta}^\star)^\top\Big[\sum_{k=1}^{V}\tilde{\pi}_k\mathbf{G}_k^{\mathbf{L}}\Big](\boldsymbol{\beta}(t)-\boldsymbol{\beta}^\star) \geq \frac{\omega}{2}\|\boldsymbol{\beta}(t)-\boldsymbol{\beta}^\star\|_2^2.$$

Using Taylor expansion, we have that

$$\begin{aligned}\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}^\star)-\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}(t)) =& -\nabla_\beta\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}(t))(\boldsymbol{\beta}(t)-\boldsymbol{\beta}^\star)+\frac{1}{2}(\boldsymbol{\beta}(t)-\boldsymbol{\beta}^\star)^\top\nabla^2_{\boldsymbol{\beta}}\mathsf{loss}(\boldsymbol{\alpha},\tilde{\boldsymbol{\beta}})(\boldsymbol{\beta}(t)-\boldsymbol{\beta}^\star)\\ \geq& -\nabla_\beta\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}(t))(\boldsymbol{\beta}(t)-\boldsymbol{\beta}^\star)+\frac{\omega}{2}\|\boldsymbol{\beta}(t)-\boldsymbol{\beta}^\star\|_2^2\\ \geq& -\frac{1}{2\omega}\|\nabla_\beta\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}(t))\|_2^2.\end{aligned}$$

This shows that $\mathsf{loss}(\boldsymbol{\alpha},\boldsymbol{\beta}(t))$ satisfies the Polyak-Lojasiewicz (PL) condition [34] when t is sufficiently large. This proves Theorem A.9. □

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 11 presents the difference of residual norms between the $\langle \mathtt{s} \rangle$ token and others ($\|\mathtt{Res}_{\langle \mathtt{s} \rangle}\| - \mathbb{E}_{v \neq \langle \mathtt{s} \rangle}[\|\mathtt{Res}_v\|]$), with different combinations of model structures. The $3 \times \mathrm{TF}$ and $2 \times \mathrm{TF} + \mathtt{mlp}$ are the architectures that demonstrate clear evidence of residual state peaks.

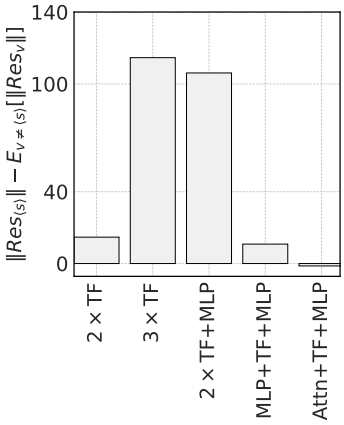


Figure 11: Minimal structures to elicit residual state peaks. We use $A + B + C$ to indicate the model with structure A, B, C in layers 0, 1, and 2, respectively.

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 12, 13, and 14 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-$\langle \mathtt{s} \rangle$ tokens. Figure 1 demonstrates that the residual state norms of $\langle \mathtt{s} \rangle$ drop match the magnitudes of other tokens at the last layer.

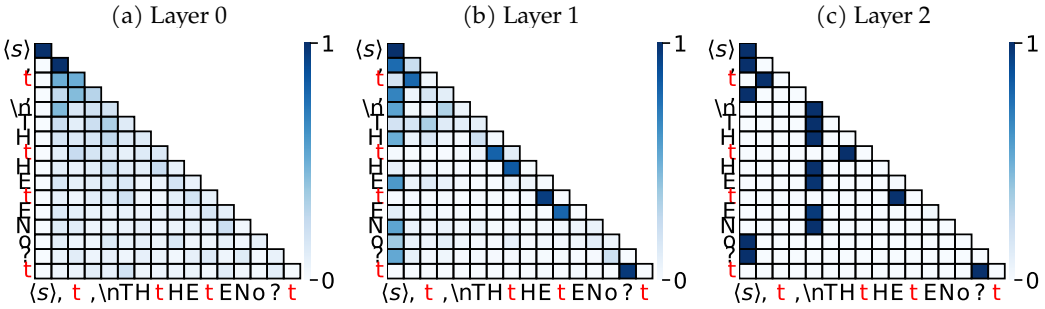


Figure 12: Attention weight patterns of three-layer transformer trained on the BB task

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:*

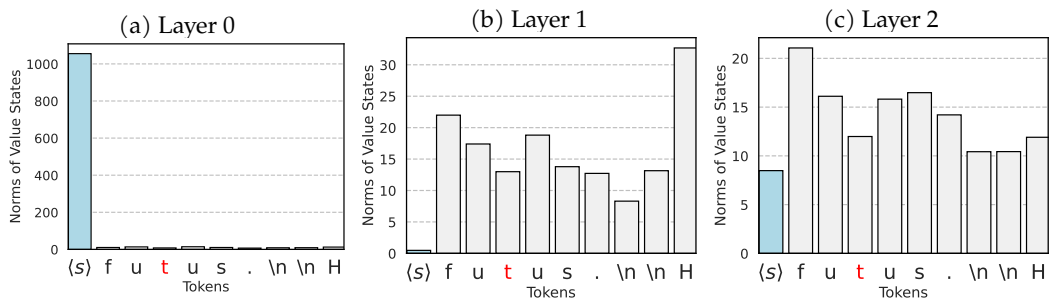


Figure 13: Value state norms of three-layer transformer trained on the BB task

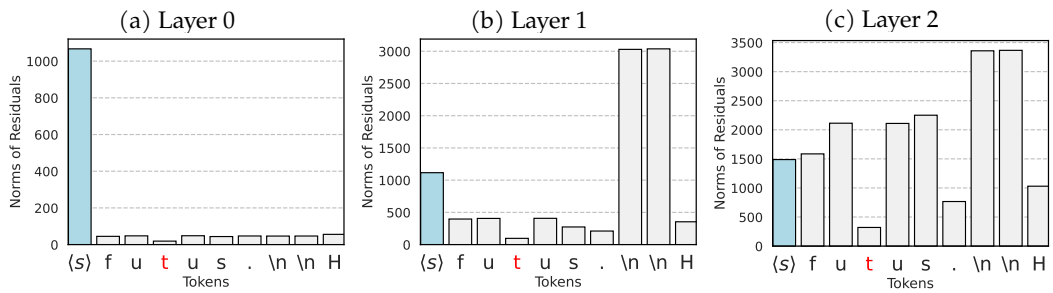


Figure 14: Residual state norms of three-layer transformer trained on the BB task

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.*
2. *The layer-norm operations 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 operation as LayerNorm. Heuristically, we can split the residual state $\texttt{Res}_{\langle \mathtt{s}\rangle}$ to a summation of two directions.

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

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 $\texttt{Key}_{\langle \mathtt{s}\rangle}$ in the original transformer, and $\boldsymbol{\varepsilon}$ corresponds to other directions. Assume that the attention logit from the token v to the $\langle \mathtt{s}\rangle$ token in layer 1 is given by

$$\text{logit}_{v,\langle \mathtt{s}\rangle} = \alpha_v = \tilde{\alpha}_v \boldsymbol{\eta}^\top \text{LayerNorm}(\texttt{Res}_{\langle \mathtt{s}\rangle}) = \tilde{\alpha}_v \cdot \frac{m+\rho}{\sqrt{m^2+2m\rho+1}}. \tag{32}$$

We assume that the scalars m and $\tilde{\alpha}$ are trainable, quantifying the norm of the residual states and magnitude of attention sinks. In the loss function loss_v as defined in Eq. (10), we replace α_v by the expression as in Eq. (32), so that the loss function becomes a function of $(\tilde{\alpha}_v, \boldsymbol{\beta}, m)$, denoted as

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

We then 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

$$\widetilde{\mathsf{loss}}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\beta}, m) = \sum_{v\in\mathcal{V}\setminus\mathcal{T}} \pi_v \cdot \widetilde{\mathsf{loss}}_v(\tilde{\alpha}_v, \boldsymbol{\beta}, m). \tag{33}$$

Proposition B.1. *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 of* $\widetilde{\mathsf{loss}}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\beta}, m)$ *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 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 15 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 7c.

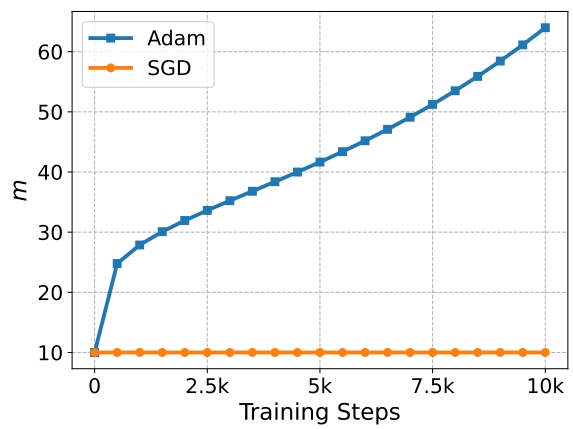


Figure 15: With the gradient formula in Proposition B.1, Adam causes linear growth of m.

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 16 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 16c.

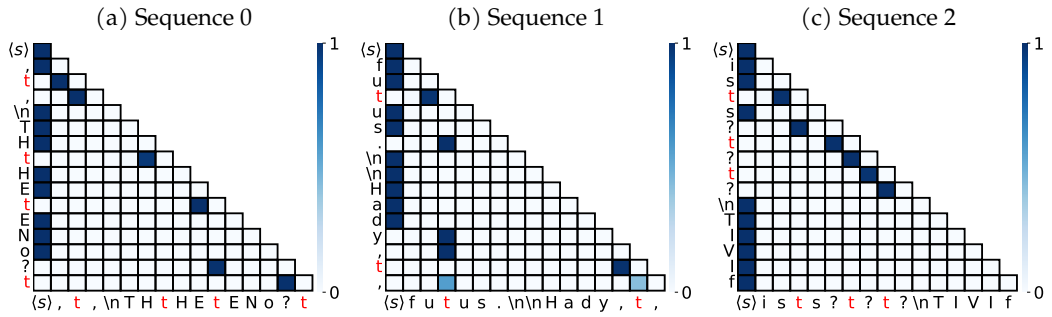


Figure 16: Additional attention plots of the one-layer transformer trained on the Bigram-Backcopy task.

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 5 with several modifications: (1) we use a trainable `mlp`-layer with random Gaussian initialization; (2) we take $\texttt{Val}_{\langle \texttt{s} \rangle} = \mathbf{O}\boldsymbol{\beta}$, 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 $\mathbf{O}$, $\texttt{Val}_{\langle \texttt{s} \rangle}$ 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 17 and 18 present statics and dynamics that match the observations in the one-layer transformer.

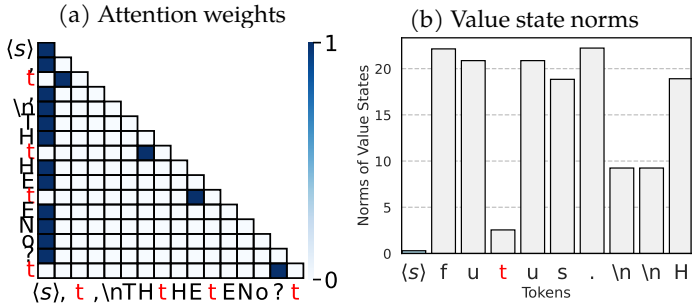


Figure 17: The simplified model structure trained on the BB task.

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

We train a one-layer transformer on the BB task without the ⟨s⟩ token. Figure 19 shows that the initial token is not a sink token. Instead, trigger tokens and delimiter tokens seem to become sink

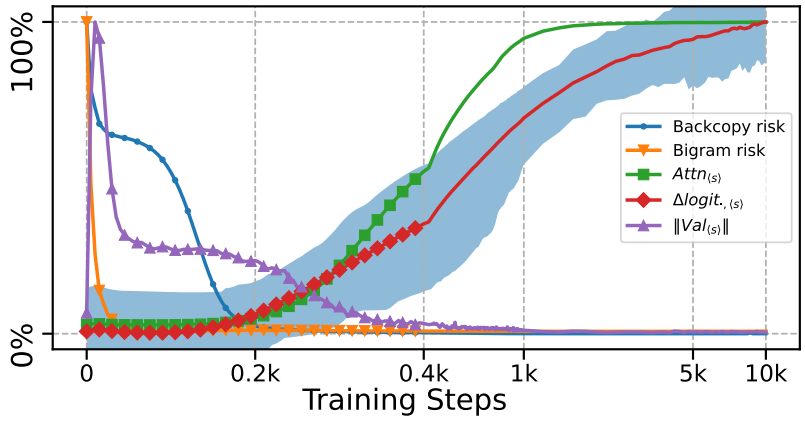


Figure 18: The dynamics of the simplified model structure trained on the BB task. The horizontal axis is logarithmatically scaled after steps 400. The excess risk curves match the one-layer transformer. The logit curve is close to the logarithmic growth predicted in Theorem 2.

tokens. In particular, the observation that delimiter tokens become extreme matches the observation in LLMs that delimiter tokens may also become extreme tokens (cf. Appendix G.2).

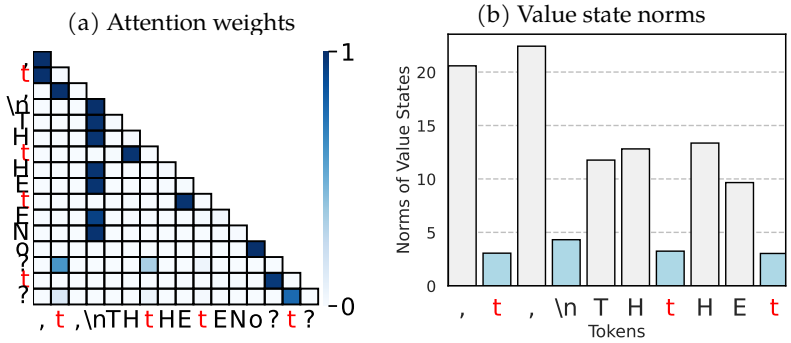


Figure 19: Attention weights and value state norms of a one-layer transformer trained on the BB task without the $\langle s \rangle$ token.

D. More Attention Heads in Dormant and Active Phase

We demonstrate a head with clear *active-dormant mechanism* in Figure 8. In this section, we present two more active-dormant heads in Llama 2-7B-Base, in Figures 20 and 21, which are more difficult to interpret than Layer 16 Head 25, but become dormant on some inputs and remain active on others.

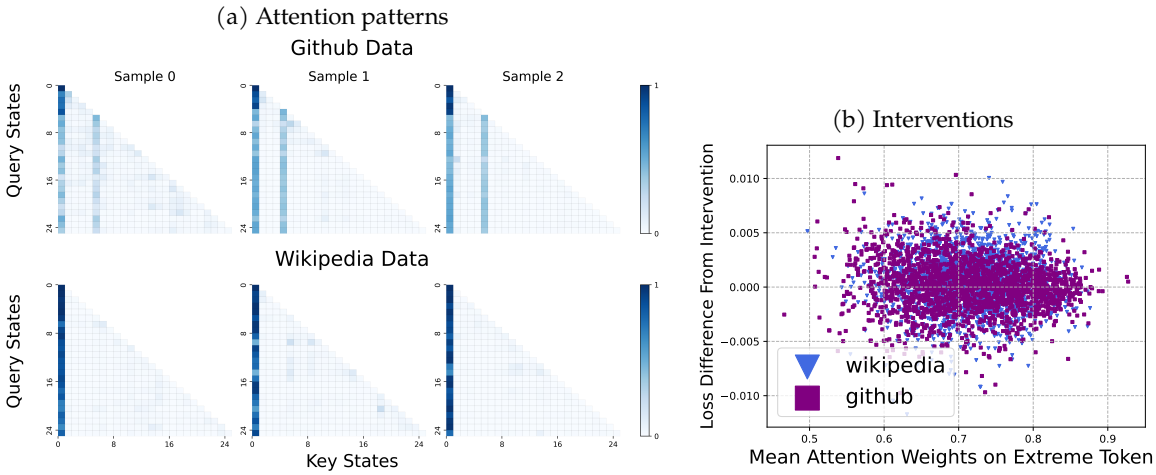


Figure 20: **Layer 16 Head 20 of Llama 2-7B-Base.** We do not observe difference between the Wikipedia data and the Github data.

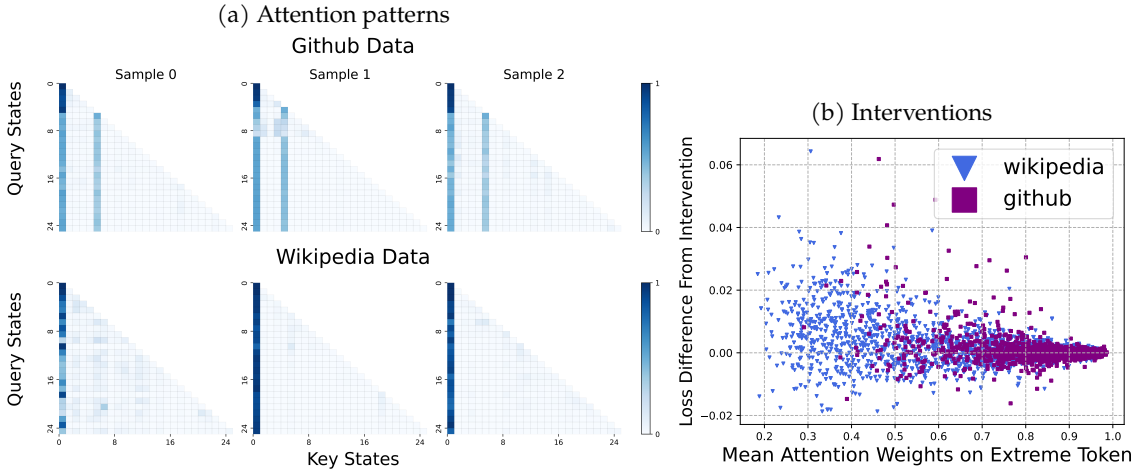


Figure 21: **Layer 16 Head 28 of Llama 2-7B-Base.** The head is more dormant on the GitHub data, and more active on the Wikipedia data.

E. Fine-Grained Static Mechanisms for Extreme-Token Phenomena

In this section, we 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."

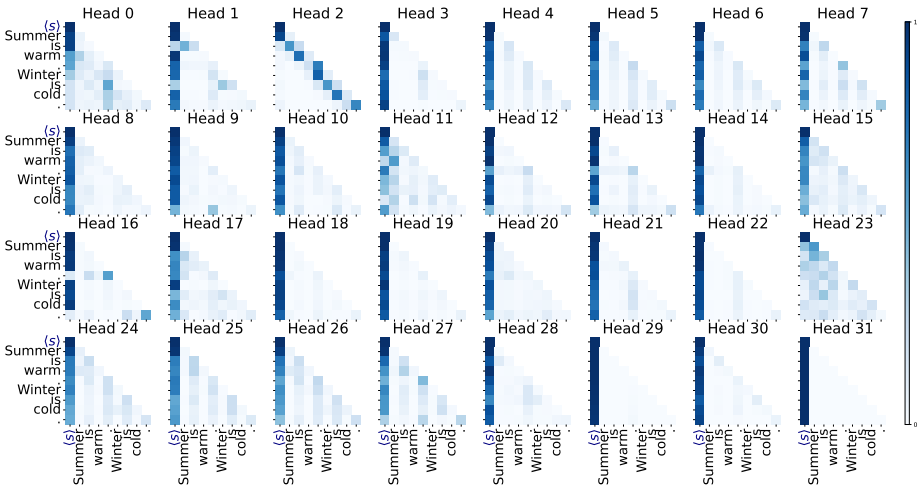


Figure 22: **A visualization of attention heads at Layer 0 of Llama 3.1-8B-Base.** Notice that many heads have the attention sink property, even at Layer 0 without any cross-token interaction. As usual, the test phrase is "Summer is warm. Winter is cold." The most clear attention sink is Head 31.

(a) Correlations between key and query states.

(b) Correlations between key states.

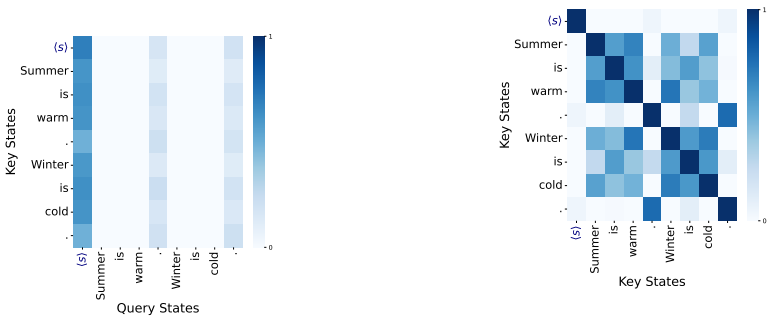


Figure 23: **Correlations between query states and key states at Layer 0 Head 31 of Llama 3.1-8B-Base.** We observe that the key state of ⟨s⟩ have low correlation with other key states, but high correlation with other query states. Meanwhile, all semantically meaningful (i.e., not delimiter) tokens have highly correlated key states.

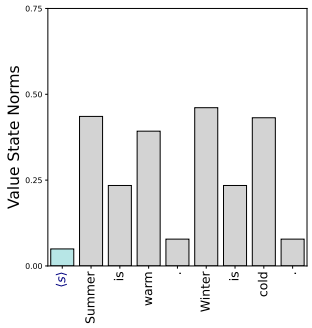
Attention sinks and global contextual semantics. There are many attention heads that exhibit attention sinks at layer 0, and the ⟨s⟩ token is always the sink token (see Figure 22). From now on until the end of this section, we restrict our attention to Head 31 of Layer 0, which is an attention sink. These attention sinks are caused by two linear-algebraic factors, demonstrated in Figure 23.

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.

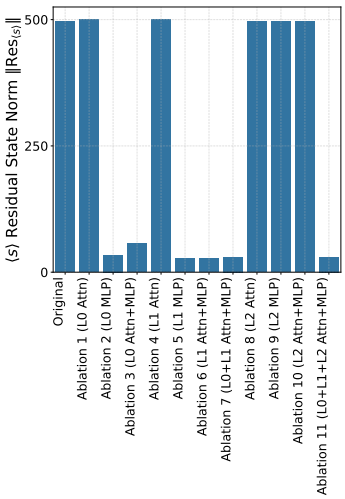


Figure 24: *Left* (*a*): Value-state drains at Layer 0 Head 31 of Llama 3.1-8B-Base. We observe that the value state associated with ⟨s⟩ is already much smaller than every other semantically meaningful token, and still smaller than the delimiter tokens in the same sentence. *Right* (*b*): Ablation study 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 MLP at layer 0 or the MLP at layer 1 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 24a. 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 24b, 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 in prompts other than "Summer is warm. Winter is cold." To this end, we use 128 samples from the Wikipedia dataset, each truncated to 8 tokens. Figure 25 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 26 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 9 and Figure 10.

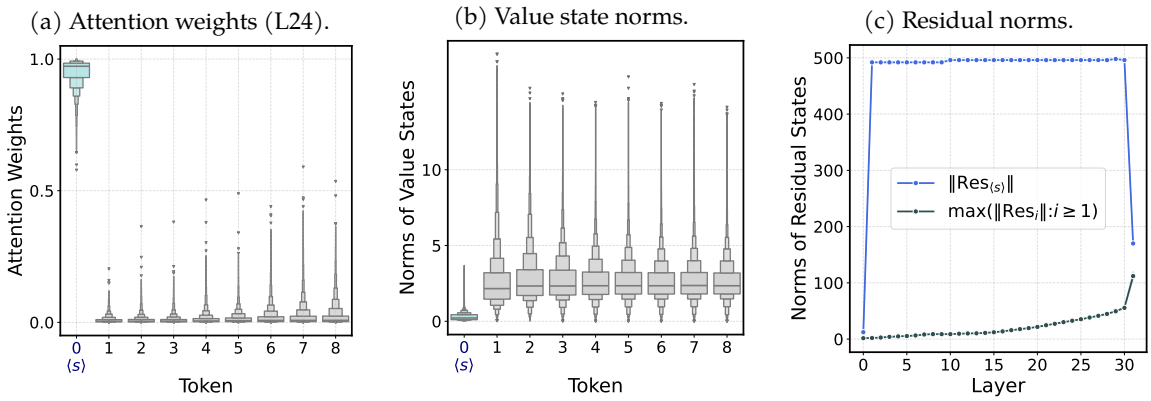


Figure 25: **Extreme token phenomena over many samples in Llama 3.1-8B-Base.** *Left* (a)*:* Let A be the attention weight tensor, of shape (batch size=128, # heads=32, # tokens=8, # tokens=8) at Layer 24 of Llama 3.1-8B-Base. We calculate the tensor $\overline{A}$, of shape (batch size=128, # heads=32, # 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 ⟨s⟩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.

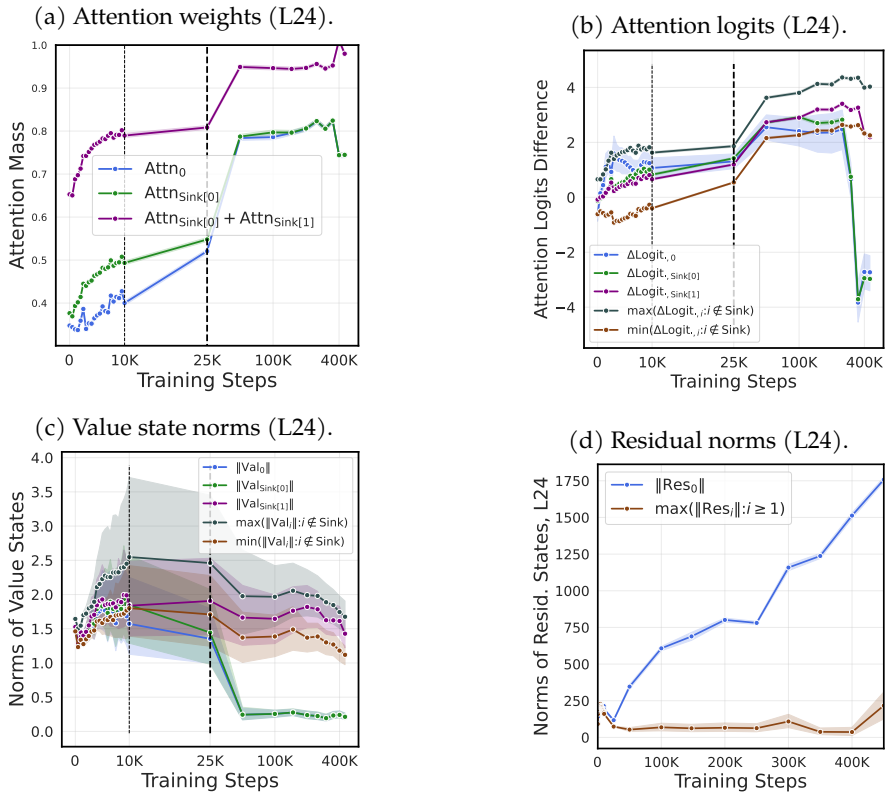


Figure 26: **Dynamics of extreme-token phenomena in layer 24 over many samples in the training trajectory of OLMo-7B.** For this experiment, as in Section 3.2, for each sample and attention head we designate two attention sink tokens as the two tokens with the largest average attention mass $\overline{A}_{bhj}$ (see Figure 25 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 sink tokens. That is, when heads in these models are dormant, they tend to have two sink tokens. For the LLMs in this group, at least on prose data, the ⟨s⟩ token as well as the initial 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 [49], which has around 411B coding tokens. Meanwhile, Llama 2 used at most (2T $\times$ 0.08 =) 0.16T coding tokens. Finally, Llama 3.1 used around (15.6T $\times$ 0.17 =) 2.6T coding tokens [15]. 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 initial 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⟨period⟩ Winter is cold⟨period⟩" 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.

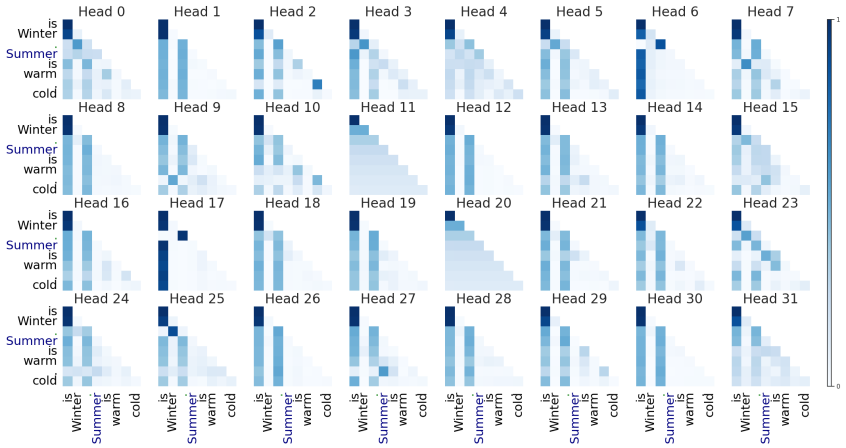


Figure 27: **Attention sinks with shuffled input in Layer 24 of OLMo.** In order to understand the impact of positional encodings when there is no ⟨s⟩ token, we shuffle the input of the test string "Summer is warm. Winter is cold." in OLMo. We observe that there is still an attention sink on token 0, 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.