Garbage Attention in Large Language Models: <BOS> Sink Heads and Sink-aware Pruning

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are known to contain significant redundancy, yet a systematic explanation for why certain components, particularly in higher layers, are more redundant has remained elusive. In this work, we identify the <BOS> sink phenomenon as a key mechanism driving this layer-wise sensitivity. We show that attention heads with high <BOS> sink scores are strongly associated with functional redundancy: such heads, especially in deeper layers, contribute little to predictive performance and effectively serve as *dumping grounds* for superfluous attention weights. This provides a concrete functional explanation for the structural redundancy reported in prior studies. Leveraging this insight, we introduce a simple pruning strategy that removes high-<BOS> sink heads. Experiments on Gemma-3, Llama-3.1, and Qwen3 demonstrate that this approach identifies redundant transformer components more reliably than weight- or activation-based criteria, while maintaining performance close to dense baselines even under aggressive pruning. Moreover, we find that the behavior of sink heads remains stable across different sequence lengths. Overall, our results suggest that structural properties of attention offer a more intuitive and robust basis for model compression than magnitude-based methods.

1 Introduction

The advent of Transformers (Vaswani et al., 2017) has enabled Large Language Models (LLMs) to achieve unprecedented performance on various tasks (Brown et al., 2020; Zhao et al., 2023). However, the massive parameter spaces of these models contain significant redundancy (Michel et al., 2019; Fan et al., 2025; He et al., 2024; Kaushik et al., 2025), serving as a primary bottleneck for efficient deployment. Consequently, while much effort has been devoted to model compression and pruning (Hoefler et al., 2021; Zhu et al., 2024; Song et al., 2024), the underlying nature and origins of this redundancy remain relatively unexplored.

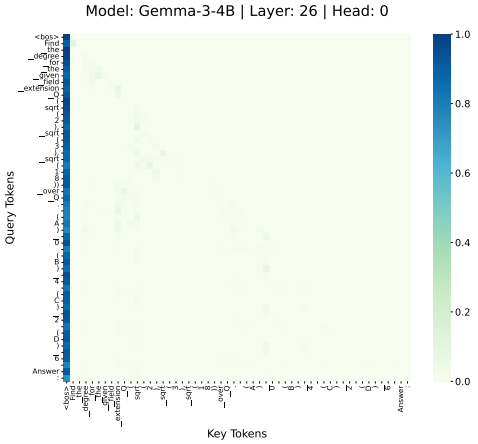

Figure 1: Attention weights of a representative <BOS> sink head (L26, H0) in Gemma-3-4B. The heatmap reveals a stark attention sink pattern where query tokens disproportionately attend to the <BOS> token during an MMLU task.

Previous research on the attention mechanism, the cornerstone of the Transformer architecture, has demonstrated that individual attention heads perform distinct functional roles (Clark et al., 2019b; Voita et al., 2019). Notably, a phenomenon known as the *attention sink* has been identified, where attention weights are disproportionately concentrated on specific tokens, particularly the Beginning-of-Sentence (<BOS>) token (Xiao et al., 2024). Further analysis reveals that this behavior is driven by specialized attention heads that exhibit a remarkably high propensity for such effects (Queipo-de Llano et al., 2025). The importance of attention sinks is evidenced by the substantial performance degradation observed when omitting the <BOS> token at inference time in models trained with it (Barbero et al., 2025).

In parallel, structural pruning studies have increasingly focused on the layer-wise sensitivity of LLMs to identify redundant components. A

recurring observation in this line of work is that LLMs exhibit a non-uniform distribution of importance across their depth, with higher layers often being more amenable to pruning or skipping without catastrophic performance loss (Gromov et al., 2025; Zhang et al., 2024; Men et al., 2025). Recent studies further demonstrate that attention heads in higher layers can be effectively pruned, particularly when combined with rescaling to preserve representation stability (Liu and Liu, 2025).

Despite their importance during training, the functional role of attention sink heads at inference remains largely underexplored. Moreover, while prior pruning literature consistently reports redundancy in higher layers, clear criteria for identifying such layers are still lacking, and proposed explanations are often post hoc.

In this work, we make the following contributions:

- We identify <BOS> sink heads (Figure 1) as structurally specialized attention heads that act as *dumping grounds* for superfluous attention weights, providing a mechanistic explanation for redundancy in LLMs.
- We show that the emergence and concentration of these heads in higher layers explain the layer-wise sensitivity consistently observed in prior pruning literature.
- We demonstrate that <BOS> sink scores constitute a simple yet robust structural metric for identifying redundant heads and layers, enabling effective structured pruning while preserving predictive performance.

2 Related Work

2.1 Attention Sink

Attention sinks describe the tendency of LLMs to allocate disproportionate mass to early tokens, especially <BOS> , as a stable outlet for surplus attention probability (Xiao et al., 2024; Gu et al., 2025). Prior work further suggests that this pattern is not merely incidental noise, but a learned allocation strategy that emerges during training to satisfy normalization constraints while preserving useful token-to-token interactions. Recent analyses show that the phenomenon is concentrated in specialized sink heads and aligned with highly redundant regions of the network (Queipo-de Llano et al., 2025), supporting the view that many such heads operate with limited task-specific utility. At the same time, removing <BOS> at inference can substantially degrade models trained to rely on it, indicating that structural anchoring and functional redundancy can coexist (Barbero et al., 2025). This duality motivates a more granular perspective: the anchor token may remain globally important even when a subset of sink-specialized heads is locally expendable. Building on this tension, we ask whether sink heads, rather than the <BOS> token itself, can be pruned safely and systematically.

2.2 Pruning Methods

Prior work reduces LLM cost through two broad paradigms: unstructured weight sparsification and structured removal of architectural units (Han et al., 2015; Ma et al., 2023). Unstructured methods target individual parameters and can achieve high sparsity, as exemplified by magnitude pruning, Wanda, SparseGPT, and Wanda++ (Sun et al., 2024; Frantar and Alistarh, 2023; Yang et al., 2025c). In contrast, structured methods remove entire components (e.g., heads) or full layers, often yielding clearer hardware compatibility and more direct latency gains (An et al., 2024; Ashkboos et al., 2024; Men et al., 2025). Within structured pruning, this distinction is commonly framed as component-level versus layer-level pruning (Song et al., 2024; Yang et al., 2024). Across these settings, multiple studies report that deeper blocks are often more prunable than earlier ones; however, selection criteria are typically derived from magnitude, saliency, similarity, or representation heuristics rather than from an explicit mechanism of functional redundancy (Men et al., 2025; Yang et al., 2025b). Our work complements this line by introducing the <BOS> sink score as a simple structural signal that links head-level redundancy to layer-wise pruning sensitivity.

3 Preliminary

3.1 Attention Sink Metrics

To analyze how attention is allocated across different tokens, we adopt the metrics from Gu et al. (2025). Let $\alpha_{t,k}^{(\ell,h)}$ denote the attention weight from the query token at position t to the key token at position k in the h-th head of the ℓ-th layer.

Attention Sink Score First, we use the attention sink score to quantify the degree to which a specific token position k attracts attention weights within a head. For a sequence of length T, the score for

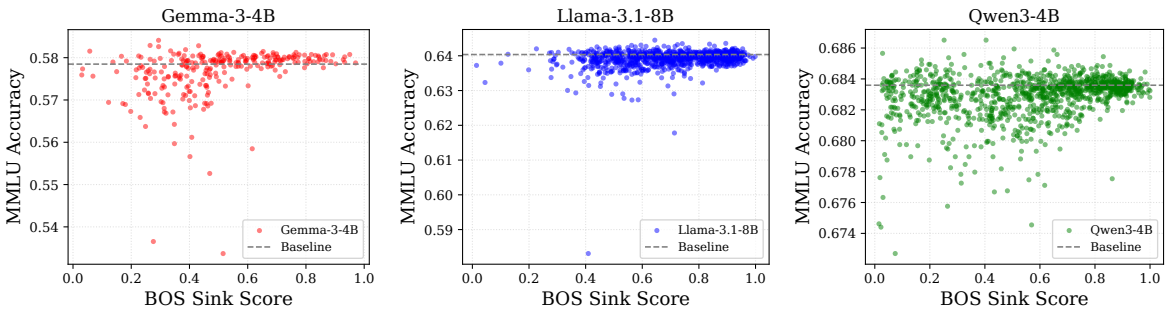

Figure 2: Single-head ablation on MMLU versus <BOS> sink score for Gemma-3-4B, Llama-3.1-8B, and Qwen3-4B. Dots denote heads and dashed lines denote model baselines; high-score heads consistently have minimal impact when ablated.

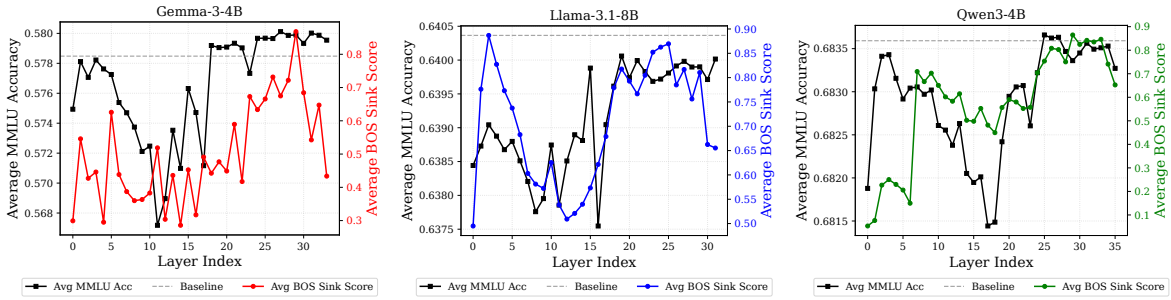

Figure 3: Relationship between <BOS> sink score and single-head ablation effect on MMLU. Colored and black lines denote layer-mean sink score (right axis) and ablated MMLU accuracy (left axis), respectively. In deeper layers, higher sink scores coincide with lower functional necessity, indicating redundant heads.

token k in head h of layer ℓ is defined as the average attention weight it receives from all tokens in the sequence:

$$\text{sink-score}_k^{(\ell,h)} = \frac{1}{T}\sum_{t=0}^{T-1} \alpha_{t,k}^{(\ell,h)} \tag{1}$$

A high score indicates that the token at position k serves as a primary destination for attention weights within that specific head.

<BOS> Sink Score Given that the <BOS> token (at index $k = 0$) often acts as a focal point for attention, we define the <BOS> sink score to measure its influence at both the head and layer levels. The head-wise <BOS> sink score $S_{\text{<BOS>}}^{(\ell,h)}$ and the layer-wise <BOS> sink score $S_{\text{<BOS>}}^{(\ell)}$ are defined as follows:

$$\begin{aligned} S_{\text{<BOS>}}^{(\ell,h)} &= \text{sink-score}_0^{(\ell,h)} \\ S_{\text{<BOS>}}^{(\ell)} &= \frac{1}{H}\sum_{h=1}^{H} S_{\text{<BOS>}}^{(\ell,h)} \end{aligned} \tag{2}$$

where H is the total number of attention heads in layer ℓ. The layer-wise <BOS> sink score $S_{\text{<BOS>}}^{(\ell)}$ serves as an aggregate metric to evaluate the dependence of layer ℓ on the <BOS> token.

3.2 Block Influence Score

To identify redundant layers within LLMs, Men et al. (2025) leverages the Block Influence (BI) score within the ShortGPT method. This metric quantifies the degree of representation transformation performed by each layer by measuring the distance between its input and output hidden states.

Definition Let $X_{i,t} \in \mathbb{R}^d$ denote the t-th row of the hidden state matrix (corresponding to the t-th token) at the i-th layer. The BI score for the i-th layer is defined as:

$$\text{BI}_i = 1 - \mathbb{E}_{X,t}\frac{X_{i,t}^T X_{i+1,t}}{\|X_{i,t}\|_2 \|X_{i+1,t}\|_2} \tag{3}$$

where $\|\cdot\|_2$ denotes the L_2 norm, and the expectation $\mathbb{E}_{X,t}$ is taken over a dataset X and token positions t.

4 <BOS> Sink as a Marker of Functional Redundancy

<BOS> Sink Score Extraction To analyze the model's internal behavior, we calculate the <BOS> sink score defined in Equation 2. Specifically, we perform 0-shot inference on the MMLU dataset and

extract attention weights directly from the attention maps during the initial decoding step.

Head-wise Ablation Figure 2 plots single-head ablation impact on MMLU versus <BOS> sink score for Llama-3.1-8B, Qwen3-4B, and Gemma-3-4B. Across all models, sensitivity separates into two regimes: high-<BOS> heads are consistently near-baseline when removed, whereas low-<BOS> heads induce larger variance and occasional accuracy drops. This low-score sensitivity is most pronounced in Gemma-3-4B, which we attribute to lower head density (272 heads) relative to Llama-3.1-8B (1,024) and Qwen3-4B (1,152), leaving less functional redundancy per head. Accordingly, high-<BOS> redundancy appears architecture-invariant, while vulnerability to low-<BOS> ablation increases as head density decreases.

Layer-wise Concentration in Deeper Blocks To characterize layer-wise redundancy, we measure the mean ablation impact of heads by layer. Figure 3 shows that <BOS> sink scores are concentrated in deeper layers. In these final blocks, higher mean <BOS> scores coincide with near-zero average accuracy loss under single-head ablation, indicating strong functional redundancy.

Early layers exhibit model-specific behavior. In Qwen3-4B and Gemma-3-4B, ablation impact remains small despite low <BOS> scores, whereas Llama-3.1-8B shows elevated early-layer <BOS> scores with occasional accuracy decline. These results suggest that high <BOS> scores are a reliable redundancy signal in deeper layers, while the role of <BOS> in initial layers is architecture-dependent.

5 Mechanistic Evidence: Sink Heads as Attention Dumping Grounds

Operational Characteristics A <BOS> sink head is defined by two primary traits: (i) a disproportionately high <BOS> sink score $S^{(\ell,h)}_{\text{<BOS>}}$, and (ii) near-zero functional sensitivity under ablation. As illustrated in Figure 2, heads in the high-<BOS> regime across Gemma-3, Llama-3.1, and Qwen3 consistently show negligible impact on MMLU accuracy when removed. This behavior suggests that these heads do not transform or convey critical semantic information but instead serve as a stable outlet within the attention mechanism.

The Structural Uniqueness of <BOS> and Redundancy A key enabler of this phenomenon is the structural uniqueness of <BOS> . Unlike other tokens, it is fixed at position 0 and intentionally non-semantic, encoding neither lexical meaning nor contextual information. It thus provides a stable, content-free target for attention mass under softmax normalization, which forces weights to sum to one even when no semantically relevant target exists (Xiao et al., 2024). We hypothesize that as representations stabilize in deeper layers, some heads increasingly exploit this property and converge on <BOS> as a non-semantic *dumping ground*, whose invariant, context-independent status offers a consistent outlet for surplus attention probability. These heads then contribute to the residual stream in a near-invariant way, shifting from semantic processing to structural maintenance—a transition reflected in Appendix A.

Layer-wise Localization of Redundancy The concentration of these dumping grounds in deeper blocks is driven by the stabilization of representations in higher layers. As the model reaches a consensus on the input's meaning, additional attention heads become functionally redundant, repurposed as numerical outlets to satisfy the softmax-induced requirement. This localized redundancy explains the minimal performance impact of pruning deeper layers. Consequently, high <BOS> sink scores offer a robust proxy for identifying this layer-specific redundancy, providing a clear empirical basis for efficient structural pruning in the model's final stages.

6 <BOS> Sink-based Structured Pruning

Based on the evidence of <BOS> sink heads as redundancy repositories, we introduce a structured pruning strategy that employs the <BOS> sink score to identify and eliminate redundant components.

Pruning Criterion We rank architectural components by their <BOS> sink scores and prune those with the highest values. Attention heads are selected by $S^{(\ell,h)}_{\text{<BOS>}}$, and transformer blocks are selected by the layer-wise score $S^{(\ell)}_{\text{<BOS>}}$ (Equation 2). To maintain stable representations, we exclude the first and last transformer blocks from pruning. The first block is preserved because it builds core lexical and positional features for later layers. The last block is preserved following ShortGPT (Men et al., 2025), which reports that removing its FFN can cause catastrophic performance degradation.

Implementation (Head Ablation) For head-wise pruning, we compute <BOS> sink scores on

Model	Strategy	Pruning Ratio	Wikitext-2 ↓	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	PIQA	WG	Avg.
Gemma-3-4B	Dense	0	10.21	0.533	0.816	0.791	0.580	0.597	0.352	0.788	0.725	0.648
	Mag-sp	0.125	17.98	0.421	0.754	0.624	0.451	0.442	0.324	0.767	0.559	0.543
	Wanda-sp		13.53	0.470	0.786	0.724	0.508	0.540	0.318	0.771	0.635	0.594
	Ours (<BOS>)		**11.27**	**0.522**	**0.813**	**0.770**	**0.571**	**0.594**	**0.348**	**0.788**	**0.718**	**0.641**
Gemma-3-12B	Dense	0	10.184	0.650	0.872	0.854	0.645	0.741	0.354	0.807	0.803	0.716
	Mag-sp	0.250	25.582	0.389	0.815	0.699	0.420	0.316	0.314	0.788	0.628	0.546
	Wanda-sp		24.619	0.411	0.824	0.661	0.447	0.595	0.310	0.768	0.564	0.573
	Ours (<BOS>)		**13.607**	**0.622**	**0.846**	**0.820**	**0.619**	**0.711**	**0.342**	**0.803**	**0.792**	**0.694**
Llama-3.1-8B	Dense	0	7.71	0.550	0.822	0.829	0.618	0.653	0.336	0.791	0.781	0.673
	Mag-sp	0.250	315.21	0.225	0.612	0.414	0.321	0.229	0.220	0.695	0.507	0.403
	Wanda-sp		**10.75**	0.410	0.796	0.760	0.487	0.344	**0.330**	**0.779**	0.668	0.572
	Ours (<BOS>)		20.55	**0.521**	**0.810**	**0.787**	**0.589**	**0.524**	0.316	0.778	**0.722**	**0.631**
Qwen3-4B	Dense	0	17.31	0.584	0.804	0.851	0.532	0.701	0.294	0.753	0.672	0.649
	Mag-sp	0.125	30.91	0.386	0.662	0.569	0.401	0.459	0.248	0.708	0.585	0.502
	Wanda-sp		**18.48**	0.542	0.760	0.804	0.513	0.653	**0.306**	0.734	0.637	0.618
	Ours (<BOS>)		19.87	**0.568**	**0.794**	**0.817**	**0.534**	**0.690**	0.302	**0.755**	**0.685**	**0.643**

Table 1: Performance comparison of component-level pruning strategies on Gemma-3-4B/12B, Llama-3.1-8B, and Qwen3-4B. We report WikiText-2 perplexity and MCQA task accuracy, where our <BOS> -based strategy consistently outperforms Mag-SP and Wanda-SP while remaining close to dense baselines.

MMLU (0-shot), rank heads by sink intensity, and select the highest-scoring heads as pruning candidates. We ablate each selected head by zeroing its slice in the output projection matrix (W_O). For a target head h in layer ℓ, we set $W_O^{(\ell,h)} \in \mathbb{R}^{d_{\text{head}} \times d_{\text{model}}}$ to zero:

$$W_O^{(\ell,h)} \leftarrow \mathbf{0} \tag{4}$$

This operation nullifies the contribution of head h before residual aggregation, effectively removing its functional effect while preserving model structure. We do not apply post-pruning rescaling or layer-norm recalibration, so performance changes directly reflect the quality of <BOS> sink score-based head selection.

7 Experiments

7.1 Experimental Setup

Models and Benchmarks We perform evaluations using three LLMs that employ Grouped-Query Attention (GQA): Gemma-3-4B/12B (Team et al., 2025), Llama-3.1-8B (Grattafiori et al., 2024), and Qwen3-4B (Yang et al., 2025a). All experiments are conducted with a maximum sequence length of 4,096 tokens.

To assess performance, we use the lm-evaluation-harness[1] (Sutawika et al., 2023) across multiple benchmark groups. For core language understanding, we evaluate WikiText-2 (Merity et al., 2017), ARC-Easy (Clark et al., 2018), BoolQ (Clark et al., 2019a), OpenbookQA (Mihaylov et al., 2018), and PIQA (Bisk et al., 2020). For reasoning-focused evaluation, we use Winogrande (Sakaguchi et al., 2019), MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), and ARC-Challenge (Clark et al., 2018). For generation, we evaluate HumanEval (Chen et al., 2021), TriviaQA (Joshi et al., 2017), and GSM8K (Cobbe et al., 2021). For long-context and retrieval, we use LongBench (Bai et al., 2023) and report HotpotQA (Yang et al., 2018), QASPER (Dasigi et al., 2021), and GovReport (Huang et al., 2021), following the LongBench configuration. Detailed dataset settings are provided in Appendix C.

Baselines and Implementation We compare our method with Wanda-SP (An et al., 2024), Mag-SP (An et al., 2024), and ShortGPT (Men et al., 2025). For layer-level pruning, we also include two positional baselines: Bottom-Up (pruning from early to late layers) and Top-Down (from late to early layers), to test whether redundancy reflects functional importance or layer position. In all settings, we keep the first and last layers to preserve stable input feature extraction and output representation.

Because the FLAP implementations of Wanda-SP and Mag-SP[2] do not natively support Grouped-Query Attention (GQA), we adapted their core logic for GQA and reimplemented all baselines under the same framework.

[1] https://github.com/EleutherAI/lm-evaluation-harness

[2] https://github.com/CASIA-LMC-Lab/FLAP

Model	Strategy	Pruning Ratio	Wikitext-2 ↓	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	PIQA	WG	Avg.
Gemma-3-4B	Dense	0	10.21	0.533	0.816	0.791	0.580	0.597	0.352	0.788	0.725	0.648
	Bottom Up	0.125	50643.64	0.206	0.314	0.466	0.260	0.241	0.146	0.549	0.488	0.334
	Top Down		45.68	0.379	0.629	0.254	0.427	0.178	0.274	0.717	0.657	0.439
	ShortGPT		20.82	0.307	0.673	0.412	0.431	0.283	0.254	0.743	0.549	0.456
	Ours (<BOS>)		**20.02**	**0.456**	**0.715**	**0.701**	**0.507**	**0.584**	**0.300**	**0.749**	**0.701**	**0.589**
Gemma-3-12B	Dense	0	10.18	0.650	0.872	0.854	0.645	0.741	0.354	0.807	0.803	0.716
	Bottom Up	0.250	3968.23	0.189	0.312	0.497	0.262	0.250	0.156	0.542	0.509	0.340
	Top Down		191.38	0.407	0.628	0.256	0.365	0.452	0.272	0.670	0.681	0.466
	ShortGPT		305692485.89	0.228	0.342	0.433	0.268	0.234	0.172	0.563	0.489	0.341
	Ours (<BOS>)		**63.98**	**0.430**	**0.665**	**0.655**	**0.450**	**0.501**	**0.274**	**0.708**	**0.690**	**0.547**
Llama-3.1-8B	Dense	0	7.71	0.550	0.822	0.829	0.618	0.653	0.336	0.791	0.781	0.673
	Bottom Up	0.250	129826.91	0.206	0.270	0.456	0.260	0.228	0.132	0.536	0.493	0.323
	Top Down		**1202.52**	**0.270**	0.362	**0.622**	0.249	**0.523**	**0.220**	0.578	**0.582**	**0.426**
	ShortGPT		7211.05	0.269	0.436	0.375	0.271	0.342	0.194	0.614	0.542	0.380
	Ours (<BOS>)		2017.63	0.212	**0.518**	0.478	**0.284**	0.266	0.210	**0.635**	0.521	0.390
Qwen3-4B	Dense	0	17.31	0.584	0.804	0.851	0.532	0.701	0.294	0.753	0.672	0.649
	Bottom Up	0.125	120.72	0.260	0.513	0.572	0.301	0.293	0.178	0.586	0.499	0.400
	Top Down		72.55	0.408	0.614	0.717	0.392	**0.682**	0.236	0.648	0.633	0.541
	ShortGPT		**41.62**	**0.460**	0.662	0.564	**0.450**	0.673	0.240	**0.680**	0.623	0.544
	Ours (<BOS>)		49.89	0.451	**0.666**	**0.826**	0.441	0.611	**0.252**	0.674	**0.642**	**0.570**

Table 2: Performance comparison of layer-level pruning strategies on Gemma-3-4B/12B, Llama-3.1-8B, and Qwen3-4B. We report WikiText-2 perplexity and MCQA task accuracy, where our <BOS> -based strategy generally outperforms Bottom-Up, Top-Down, and ShortGPT.

Model	Strategy	Pruning Ratio	GSM8K	HumanEval	TriviaQA
Gemma-3-4B	Dense	0	0.392	0.348	0.605
	Mag-sp	0.125	0.010	0.049	0.489
	Wanda-sp		0.024	0.171	0.571
	Ours (<BOS>)		**0.263**	**0.341**	**0.588**
Gemma-3-12B	Dense	0	0.715	0.427	0.747
	Mag-sp	0.125	0.040	0.189	0.674
	Wanda-sp		0.287	0.250	0.682
	Ours (<BOS>)		**0.667**	**0.402**	**0.714**
Llama-3.1-8B	Dense	0	0.491	0.366	0.704
	Mag-sp	0.125	0	0.073	0.517
	Wanda-sp		0.150	0.274	**0.687**
	Ours (<BOS>)		**0.445**	**0.329**	0.634
Qwen3-4B	Dense	0	0.861	0.750	0.442
	Mag-sp	0.125	0.017	0.134	0.232
	Wanda-sp		0.188	0.494	**0.408**
	Ours (<BOS>)		**0.317**	**0.543**	0.376

Table 3: Performance comparison of generation tasks under component-level pruning on Gemma-3-4B/12B, Llama-3.1-8B, and Qwen3-4B. We report GSM8K and TriviaQA accuracy, and HumanEval Pass@1.

7.2 Perplexity and MCQA Results

Component-level Pruning Table 1 summarizes component-level pruning results. Our <BOS>-based method achieves the highest MCQA average among the pruning baselines while remaining close to the dense baselines.

At 12.5% pruning, <BOS>-based pruning achieves average accuracies of 0.641 and 0.643 on Gemma-3-4B and Qwen3-4B, respectively, compared with dense baselines of 0.648 and 0.649. At 25.0%, it also performs best on Gemma-3-12B (0.694) and Llama-3.1-8B (0.631), consistently outperforming Mag-SP and Wanda-SP. Although Wanda-SP occasionally yields lower WikiText-2 perplexity, our method better preserves MCQA accuracy, suggesting that <BOS> sink scores are a stronger criterion for maintaining reasoning performance under structured pruning. Additional results are provided in Appendix B.

Layer-level Pruning Table 2 reports layer-level pruning, where each step removes full transformer blocks (attention + FFN). Overall, the <BOS>-based strategy shows the strongest or most competitive MCQA retention across model families.

At 12.5%, <BOS> achieves the best averages on Gemma-3-4B (0.589) and Qwen3-4B (0.570), outperforming ShortGPT and positional baselines. At 25.0%, it also leads on Gemma-3-12B (0.547), while on Llama-3.1-8B it surpasses Bottom-Up and ShortGPT (0.390 vs. 0.323/0.380) but trails Top-Down (0.426). These results support <BOS> sink scores as a practical criterion for identifying less critical layers under structured pruning.

7.3 Generation Results

Table 3 summarizes generation performance under component-level pruning. We report GSM8K and TriviaQA accuracy, and HumanEval Pass@1. Overall, <BOS>-based pruning is substantially more robust than Mag-SP and remains competitive with Wanda across model families.

For Gemma-3-4B/12B and Llama-3.1-8B, <BOS> consistently yields the best pruned results on all three generation benchmarks, with especially

Model	Strategy	Pruning Ratio	HotpotQA	QASPER	GovReport
Gemma-3-4B	Dense	0	0.071	0.101	0.270
	Mag-sp	0.125	0.015	0.023	0.071
	Wanda-sp		0.047	0.043	**0.153**
	Ours (<BOS>)		**0.048**	**0.063**	0.140
Llama-3.1-8B	Dense	0	0.108	0.134	0.302
	Mag-sp	0.125	0.043	0.048	0.098
	Wanda-sp		0.087	0.090	0.191
	Ours (<BOS>)		**0.093**	**0.134**	**0.237**
Qwen3-4B	Dense	0	0.128	0.104	0.300
	Mag-sp	0.125	0.034	0.043	0.132
	Wanda-sp		0.069	**0.100**	**0.247**
	Ours (<BOS>)		**0.084**	0.087	0.235

Table 4: Performance comparison of long-context and retrieval tasks under component-level pruning on Gemma-3-4B, Llama-3.1-8B, and Qwen3-4B. We report HotpotQA and QASPER F1 scores, and GovReport ROUGE.

large gains on GSM8K and HumanEval relative to magnitude-based baselines. On Qwen3-4B, the trend is more mixed: <BOS> remains clearly stronger than Mag-SP and is competitive with Wanda-SP. These results are consistent with Section 7.2, indicating that <BOS> sink scores preserve not only MCQA performance but also generation quality under structured pruning.

7.4 Long-Context and Retrieval Results

Table 4 summarizes long-context and retrieval performance under component-level pruning. We report HotpotQA and QASPER F1 scores, and GovReport ROUGE. Overall, <BOS>-based pruning consistently outperforms Mag-SP and remains competitive with Wanda across LongBench tasks.

In particular, <BOS> shows strong gains on retrieval-oriented tasks for Gemma-3-4B and Llama-3.1-8B, including matching the dense QASPER score on Llama-3.1-8B at a pruning ratio of 0.125. On Qwen3-4B, our method remains robust and consistently outperforms Mag-SP, although Wanda performs better on QASPER and GovReport. This pattern indicates model-family sensitivity in the reliability of the <BOS> pruning signal.

8 Validation of the <BOS> Sink-based Pruning Criterion

8.1 Distinctiveness of <BOS> -based Head Selection

To directly examine whether our pruning criterion selects a head set that is truly distinct from existing methods, we compute the intersection-over-union (IoU) between the attention heads pruned by our method and those selected by Mag-SP and Wanda-SP under component-level pruning. As shown in Table 5, our method has consistently low overlap with Mag-SP (0.125, 0.113, and 0.104 for Gemma-3-4B, Llama-3.1-8B, and Qwen3-4B, respectively). Although overlap with Wanda-SP is higher (0.219, 0.453, and 0.431), a substantial portion of heads remains non-overlapping. These results suggest that the <BOS> sink signal captures structural redundancy that is not fully explained by weight magnitude or activation-based criteria alone. Additional experimental details are presented in Appendix E.

Model	Pruning Ratio	Mag-sp	Wanda-sp
Gemma-3-4B	0.125	0.125	0.219
Llama-3.1-8B	0.25	0.113	0.453
Qwen3-4B	0.125	0.104	0.431

Table 5: IoU of pruned attention heads between our method and Mag-SP/Wanda-SP under component-level pruning.

8.2 Consistency Across Context Lengths

To further validate the reliability of <BOS> sink scores as a pruning metric, we investigate their stability across varying input sequence lengths T.

Scaling and Structural Normalization By definition, the <BOS> sink score represents the average of attention weights assigned to the <BOS> token throughout a sequence. Due to the Softmax operation's normalization—where the sum of attention weights for each query must equal 1—the average attention weight per token naturally scales with $1/T$. Consequently, as T increases, absolute <BOS> sink scores exhibit an inevitable downward trend across all heads. This reduction should be viewed as a structural consequence of the attention mechanism's normalization rather than a loss of functional importance.

Differential Stability of Sink Heads Notably, we observe a clear correlation between a head's average <BOS> sink score μ and its robustness to increasing sequence lengths. As illustrated in Figure 4, the rate of decline in attention scores is inversely related to the magnitude of μ: while the average score for all heads drops sharply, heads with $\mu \geq 0.6$ show more tempered decreases, and those with $\mu \geq 0.8$ exhibit the most gradual downward trajectories. This differential stability indicates that higher-scoring <BOS> sink heads appear to partially mitigate the $1/T$ scaling effect, maintaining a disproportionately high focus on the <BOS> token even as the total attention budget is distributed over an

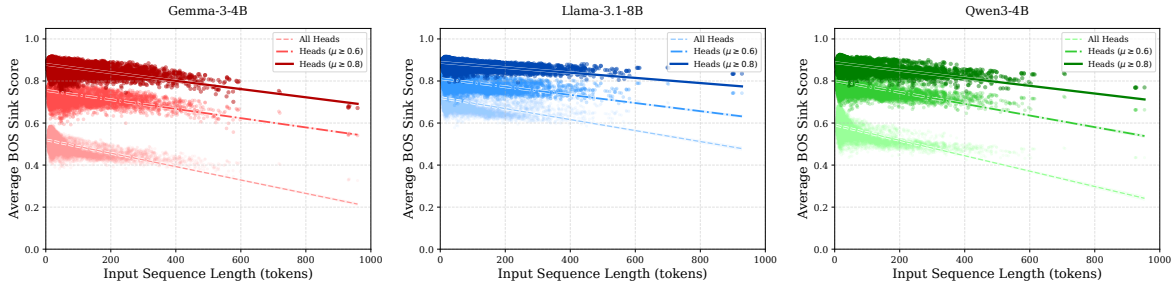

Figure 4: Mean <BOS> sink score versus sequence length T. Points show input-level means, and dashed, dash-dotted, and solid lines are linear fits for all heads, $\mu \geq 0.6$, and $\mu \geq 0.8$, respectively. Although softmax implies $1/T$ scaling, higher-μ sink heads show shallower slopes, indicating more stable <BOS> focus at longer contexts.

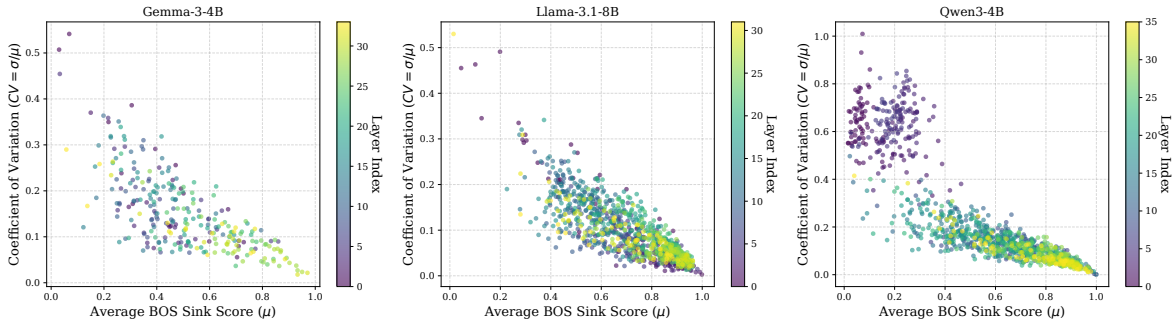

Figure 5: Relationship between mean <BOS> sink score (μ) and coefficient of variation (CV). Each dot denotes an attention head, colored by layer. Higher-μ heads exhibit lower CV and are concentrated in deeper layers, indicating stable sink behavior and layer-wise redundancy.

increasingly large number of tokens.

Positional Bias and Convergence Furthermore, our analysis reveals a strong positional bias in the emergence and stability of these heads. To quantify this, we use the coefficient of variation ($CV = \sigma/\mu$), where σ and μ represent the standard deviation and mean of the <BOS> sink scores across different sequence lengths. As shown in Figure 5, a clear inverse correlation exists between μ and CV. Crucially, heads with high μ and near-zero CV are predominantly concentrated in the deeper layers of the architecture (indicated by the yellow and green hues). This suggests that as the model's layers deepen, certain heads converge into stable <BOS> sinks that maintain consistent attention patterns regardless of the input sequence length.

9 Conclusion

In this study, we established the <BOS> sink phenomenon as a primary driver of functional redundancy in LLMs, revealing a strong correlation between high <BOS> sink scores and negligible predictive contribution. Our findings provide a systematic explanation for the high redundancy observed in deeper layers, where specialized heads act as stable repositories for unnecessary attention weights, effectively functioning as internal *dumping grounds*. By bridging the gap between internal attention patterns and layer-wise sensitivity, we demonstrated that structural attention properties can serve as a more efficient and direct alternative to traditional weight-based metrics for model compression. Experiments across Gemma-3, Llama-3.1, and Qwen3 confirm that our straightforward <BOS>-based pruning matches or exceeds the efficacy of established methods like Wanda-SP and Mag-SP.

Future work will focus on validating the <BOS> sink phenomenon across different attention architectures, including MoE and MHA, to assess whether sink behavior is architecture-dependent. Furthermore, we aim to investigate whether similar attention-anchoring mechanisms exist in other Transformer-based architectures beyond text, such as Vision-Language Models (VLMs). Additionally, investigating the developmental dynamics of these sinks during the pre-training phase may offer deeper insights into how models establish representational stability through structural redundancy.

Limitations

Despite the robust findings presented in this study, several limitations remain.

First, while we established a strong correlation between high <BOS> sink scores and functional redundancy, our work is primarily based on empirical observations. We demonstrated that these specialized heads predominantly emerge and concentrate in the deeper layers of the architecture, yet we do not provide a formal theoretical derivation for the underlying cause of this positional concentration.

Second, although we validated the efficacy of our <BOS> -based pruning strategy across diverse GQA-based architectures—specifically Gemma-3-4B/12B, Llama-3.1-8B, and Qwen3-4B—further experiments are needed on ultra-large-scale models (e.g., 70B and above) and on a broader range of attention architectures beyond GQA, including MoE and MHA, to verify whether these attention patterns and redundancy characteristics generalize.

Ethical Considerations

Fairness and Potential Bias The pruning process, by removing model parameters, carries the risk of inadvertently amplifying biases toward specific demographic groups inherent in the training data. While our study demonstrates robust performance retention on general reasoning benchmarks such as MMLU and ARC, the specific impact on social bias and toxicity remains to be fully characterized. Consequently, we strongly advise conducting a comprehensive evaluation to detect any discriminatory behaviors against marginalized communities before deploying the pruned models in production environments.

Use of AI Assistants In accordance with the ACL Policy on AI Assistance, we acknowledge the use of Gemini 3 Pro[3] to assist with code debugging and improving the readability and grammar of this manuscript. The final content and responsibility for the work rest solely with the authors.

References

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. 2024. Fluctuation-based adaptive structured pruning for large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 10865–10873. AAAI Press.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari Do Nascimento, Torsten Hoefler, and James Hensman. 2024. Slicegpt: Compress large language models by deleting rows and columns. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongyuan Zhang, Yinan Li, Zhe Xu, Yutao Sun, Xin Wang, Jing Liu, Lei Hou, Yuxiao Dong, and Jie Tang. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Petar Veličković, Razvan Pascanu, and Michael M. Bronstein. 2025. Why do LLMs attend to the first token? In *Second Conference on Language Modeling*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019a. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019b. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP:*

[3] https://deepmind.google/models/gemini/

Analyzing and Interpreting Neural Networks for NLP, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. In *arXiv preprint arXiv:2110.14168*.

Pradeep Dasigi, Kyle Lo, Luca Soldaini, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610. Association for Computational Linguistics.

Siqi Fan, Xin Jiang, Xiang Li, Xuying Meng, Peng Han, Shuo Shang, Aixin Sun, and Yequan Wang. 2025. Not all layers of llms are necessary during inference. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 5083–5091. ijcai.org.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pages 10323–10337. PMLR.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. 2025. The unreasonable ineffectiveness of the deeper layers. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2025. When attention sink emerges in language models: An empirical view. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.

Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. 2024. What matters in transformers? not all attention is needed. *arXiv preprint arXiv:2406.15786*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. 2021. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241):1–124.

Luyang Huang, Shuyang Cao, Nicola Parulian, Heng Ji, and Lu Wei. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611. Association for Computational Linguistics.

Prakhar Kaushik, Shravan Chaudhari, Ankit Vaidya, Rama Chellappa, and Alan Yuille. 2025. The universal weight subspace hypothesis. *arXiv preprint arXiv:2512.05117*.

Songtao Liu and Peng Liu. 2025. High-layer attention pruning with rescaling. *arXiv preprint arXiv:2507.01900*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2025. Shortgpt: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on*

Empirical Methods in Natural Language Processing, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Enrique Queipo-de Llano, Álvaro Arroyo, Federico Barbero, Xiaowen Dong, Michael Bronstein, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Attention sinks and compression valleys in llms are two sides of the same coin. *arXiv preprint arXiv:2510.06477*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Jiwon Song, Kyungseok Oh, Taesu Kim, Hyungjun Kim, Yulhwa Kim, and Jae-Joon Kim. 2024. SLEB: streamlining llms through redundancy verification and elimination of transformer blocks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Lintang Sutawika, Leo Gao, Hailey Schoelkopf, Stella Biderman, Jonathan Tow, Baber Abbasi, ben fattori, Charles Lovering, farzanehnakhaee70, Jason Phang, Anish Thite, Fazz, Aflah, Niklas Muennighoff, Thomas Wang, sdtblck, nopperl, gakada, tttyuntian, and 11 others. 2023. Eleutherai/lm-evaluation-harness: Major refactor.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

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

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Liangwei Yang, Yuhui Xu, Juntao Tan, Doyen Sahoo, Silvio Savarese, Caiming Xiong, Huan Wang, and Shelby Heinecke. 2025b. Entropy-based block pruning for efficient large language models. *arXiv preprint arXiv:2504.03794*.

Yifan Yang, Kai Zhen, Bhavana Ganesh, Aram Galstyan, Goeric Huybrechts, Markus Müller, Jonas M. Kübler, Rupak Vignesh Swaminathan, Athanasios Mouchtaris, Sravan Babu Bodapati, Nathan Susanj, Zheng Zhang, Jack FitzGerald, and Abhishek Kumar. 2025c. Wanda++: Pruning large language models via regional gradients. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 4321–4333. Association for Computational Linguistics.

Yifei Yang, Zouying Cao, and Hai Zhao. 2024. Laco: Large language model pruning via layer collapse. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 6401–6417. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Yang Zhang, Yawei Li, Xinpeng Wang, Qianli Shen, Barbara Plank, Bernd Bischl, Mina Rezaei, and Kenji Kawaguchi. 2024. Finercut: Finer-grained interpretable layer pruning for large language models. *arXiv preprint arXiv:2405.18218*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.

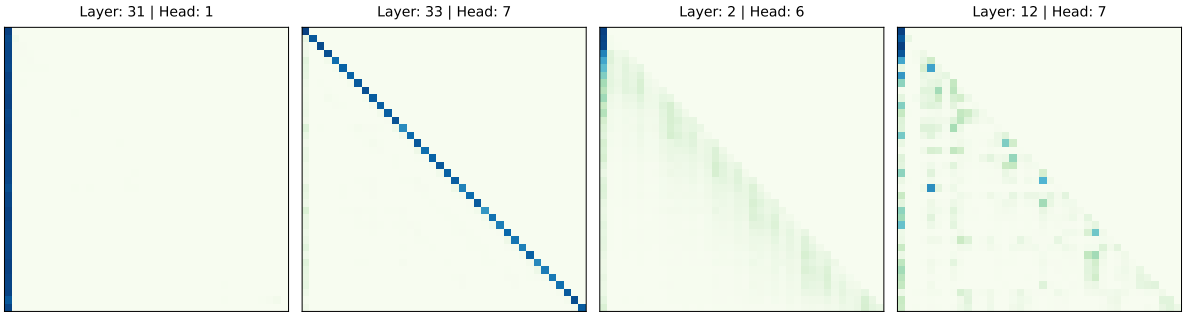

Figure 6: Visualization of four representative attention patterns in Gemma-3-4B during MMLU tasks. From left to right, the maps illustrate: (1) **<BOS> Sink** showing intense focus on the initial token (Layer 31, Head 1); (2) **Diagonal (Self-Attention)** with weights concentrated strictly on the query token (Layer 33, Head 7); (3) **Uniform** broad attention distribution primarily observed in lower layers (Layer 2, Head 6); and (4) **Random** sparse attention patterns (Layer 12, Head 7).

A Patterns of Attention

To qualitatively characterize the attention behaviors underlying model redundancy, we visualize the attention maps of Gemma-3-4B during inference. As illustrated in Figure 6, we identify four representative patterns that define the functional landscape of the attention mechanism: <BOS> Sink, Diagonal, Uniform, and Random. Our qualitative analysis of these signatures yields three primary findings regarding the structural properties of these attention heads.

Dominance of Non-Transformative Patterns A significant majority of attention heads exhibit either <BOS> sink or Diagonal patterns. While <BOS> sinks are known as repositories for unnecessary attention weights, we hypothesize that Diagonal patterns may serve a similar function. In a residual architecture, a head attending solely to its own position effectively acts as an identity map, contributing no new contextual information to the hidden state and thus remaining functionally redundant.

Layer-wise Specialization Uniform patterns, which facilitate broad information mixing, are predominantly concentrated in the lower layers (e.g., Layer 2). This suggests that global contextual integration occurs early in the forward pass, while deeper layers transition toward specialized *dumping grounds* behaviors like <BOS> or Diagonal sinks.

Functional Stability We observed that these attention signatures tend to remain input-invariant. Regardless of changes in the input sequence, individual heads appear to maintain their specific patterns. This stability suggests that attention behaviors could be interpreted as structural traits inherent to the pre-trained architecture, rather than purely dynamic responses to specific semantic content. While these observations provide meaningful insights, a more rigorous validation remains for future work to confirm the extent of this inherent stability.

B Robustness to Model Pruning

To evaluate our method, we perform component-level pruning on Gemma-3-4B, Qwen3-4B, and Llama-3.1-8B, and test on MMLU (5-shot) and WikiText-2. Across all three model families, the same trend appears: performance degrades more gracefully under our criterion than under baseline pruning rules. This margin is especially clear in moderate pruning regimes, where accuracy remains close to each dense baseline. Figure 7 shows that our approach is notably more robust on MMLU than strong baselines (Wanda-SP, Mag-SP).

Table 6 reports downstream results for component-level pruning. At pruning ratios of 0.125 and 0.25 across WikiText-2 and eight MCQA benchmarks, our <BOS>-based method achieves the best average retention in most settings.

Although <BOS> perplexity rises more sharply beyond a 0.5 pruning ratio, it remains competitive in the practically useful low-to-moderate range (≤ 0.5). Given that heavier pruning often yields non-functional models, this result supports <BOS> as an effective strategy for real-world compression.

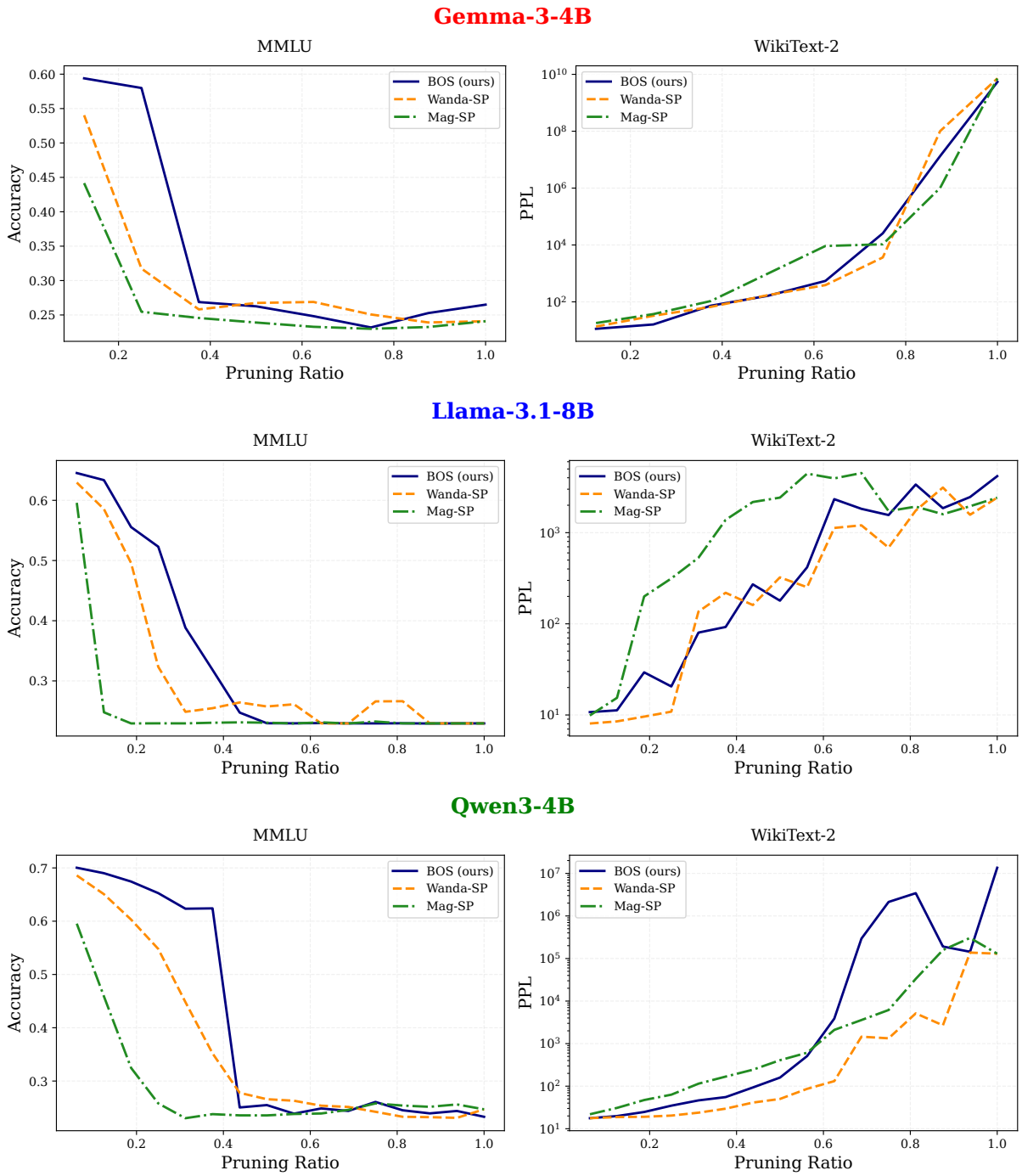

Figure 7: Performance comparison of component-level pruning methods on Gemma-3-4B, Qwen3-4B, and Llama-3.1-8B. We report MMLU 5-shot accuracy and WikiText-2 perplexity as a function of the pruning ratio. Our method consistently demonstrates stronger accuracy robustness across all models over the full range of pruning ratios compared to the baselines.

Model	Strategy	Pruning Ratio	Wikitext-2 ↓	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	PIQA	WG	Avg.
Gemma-3-4B	Dense	0	10.211	0.533	0.816	0.791	0.580	0.597	0.352	0.788	0.725	0.648
	Mag	0.125	17.983	0.421	0.754	0.624	0.451	0.442	0.324	0.767	0.559	0.543
	Wanda		13.532	0.470	0.786	0.724	0.508	0.540	0.318	0.771	0.635	0.594
	Ours (<BOS>)		**11.271**	**0.522**	**0.813**	**0.770**	**0.571**	**0.594**	**0.348**	**0.788**	**0.718**	**0.641**
	Mag	0.25	37.125	0.223	0.714	0.592	0.365	0.255	0.272	0.742	0.543	0.463
	Wanda		33.109	0.241	0.689	0.616	0.341	0.300	0.246	0.742	0.537	0.464
	Ours (<BOS>)		**16.028**	**0.494**	**0.777**	**0.765**	**0.548**	**0.580**	**0.328**	**0.781**	**0.720**	**0.624**
Llama-3.1-8B	Dense	0	7.713	0.550	0.822	0.829	0.618	0.653	0.336	0.791	0.781	0.673
	Mag	0.125	15.408	0.384	0.773	0.580	0.445	0.248	0.326	0.762	0.591	0.514
	Wanda		**8.501**	0.533	0.812	0.812	0.559	0.585	**0.346**	0.788	0.743	0.647
	Ours (<BOS>)		11.249	**0.559**	**0.825**	**0.823**	**0.611**	**0.634**	0.340	**0.789**	**0.770**	**0.669**
	Mag	0.25	315.209	0.225	0.612	0.414	0.321	0.229	0.220	0.695	0.507	0.403
	Wanda		**10.751**	0.410	0.796	0.760	0.487	0.344	**0.330**	**0.779**	0.668	0.572
	Ours (<BOS>)		20.550	**0.521**	**0.810**	**0.787**	**0.589**	**0.524**	0.316	0.778	**0.722**	**0.631**
Qwen3-4B	Dense	0	17.311	0.584	0.804	0.851	0.532	0.701	0.294	0.753	0.672	0.649
	Mag	0.125	30.915	0.386	0.662	0.569	0.401	0.459	0.248	0.708	0.585	0.502
	Wanda		**18.485**	0.542	0.760	0.804	0.513	0.653	**0.306**	0.734	0.637	0.618
	Ours (<BOS>)		19.867	**0.568**	**0.794**	**0.817**	**0.534**	**0.690**	0.302	**0.755**	**0.685**	**0.643**
	Mag	0.25	63.242	0.334	0.547	0.453	0.353	0.258	0.210	0.652	0.511	0.415
	Wanda		**20.515**	0.486	**0.760**	0.668	0.450	0.550	**0.290**	**0.729**	0.590	0.565
	Ours (<BOS>)		34.869	**0.534**	0.741	**0.747**	**0.494**	**0.653**	0.246	**0.729**	**0.672**	**0.602**

Table 6: Downstream performance under component-level pruning at ratios of 0.125 and 0.25. We compare Mag, Wanda, and our <BOS>-based criterion across Gemma-3-4B, Llama-3.1-8B, and Qwen3-4B, reporting Wikitext-2 perplexity (↓) and the average score over eight zero/few-shot benchmarks (↑).

Category	Dataset	Year	Description	Shots	License
Perplexity	WikiText-2	2017	Language modeling/perplexity evaluation on Wikipedia text passages.	0-shot	CC BY-SA 3.0
MCQA	ARC-Easy	2018	Grade-school science multiple-choice questions (easy split).	0-shot	CC BY-SA 4.0
	BoolQ	2019	Yes/no reading comprehension based on short passages.	0-shot	CC BY-SA 3.0
	OpenBookQA	2018	Science multiple-choice reasoning with open-book style facts.	0-shot	Apache 2.0
	PIQA	2020	Physical commonsense reasoning for everyday situations.	0-shot	Apache 2.0
	Winogrande	2019	Pronoun/coreference disambiguation under commonsense constraints.	5-shot	Apache 2.0
	MMLU	2021	Multi-domain knowledge and reasoning across academic/professional subjects.	5-shot	MIT
	HellaSwag	2019	Commonsense sentence completion and grounded continuation selection.	10-shot	MIT
	ARC-Challenge	2018	More difficult split of grade-school science QA.	25-shot	CC BY-SA 4.0
Generation	HumanEval	2021	Code generation from docstrings, evaluated by unit tests.	0-shot	MIT
	TriviaQA	2017	Open-domain question answering using evidence documents.	5-shot	Apache 2.0
	GSM8K	2021	Multi-step grade-school math word-problem solving.	8-shot	MIT
Long Context and Retrieval	HotpotQA	2018	Multi-hop question answering requiring evidence integration across documents.	0-shot	CC BY-SA 4.0
	QASPER	2021	Question answering over scientific papers (long-context comprehension).	0-shot	CC BY 4.0
	GovReport	2021	Long-document government report summarization.	0-shot	CC BY 4.0

Table 7: Summary of datasets used in our pruning experiments. For long-context evaluation, we follow the LongBench configuration and report results on HotpotQA, QASPER, and GovReport.

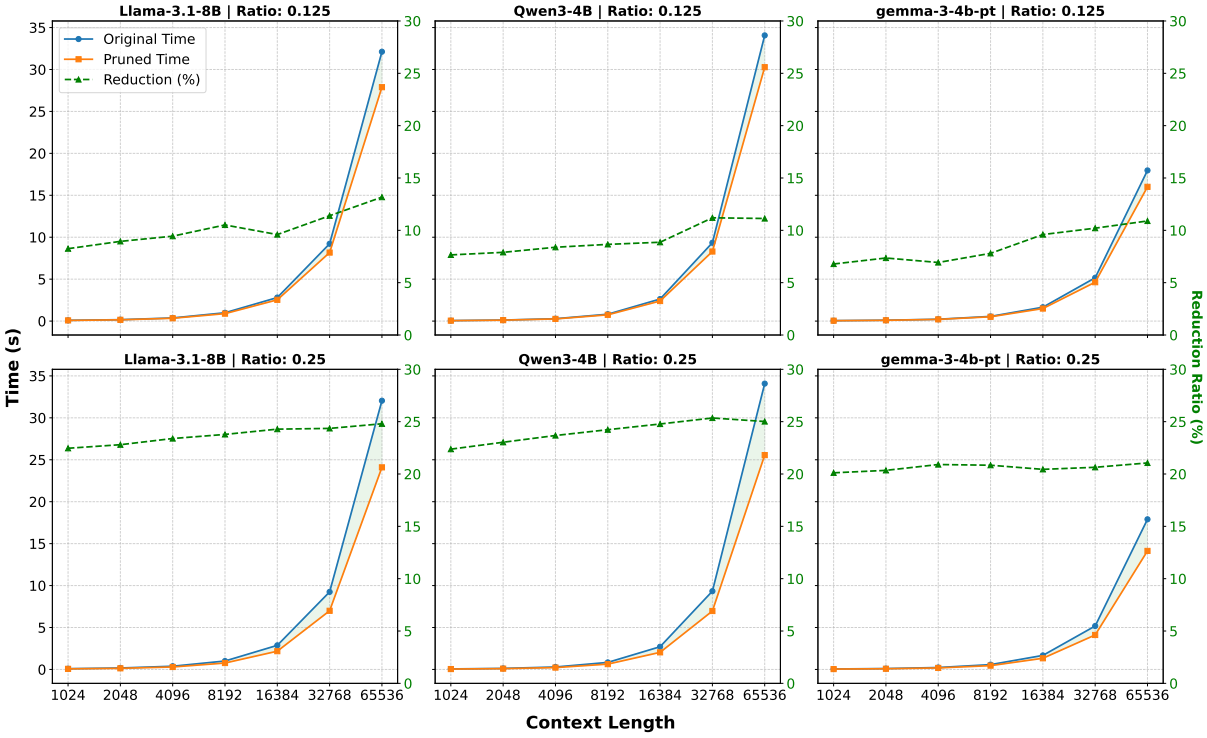

Figure 8: Wall-clock inference latency (seconds) of original and pruned models under layer-level pruning across sequence lengths from 1,024 to 65,536 tokens. Results are reported for Llama-3.1-8B, Qwen3-4B, and Gemma-3-4B at pruning ratios of 0.125 and 0.25. The proposed method consistently reduces latency across all settings, yielding about 7–13% speedup at 0.125 pruning and 20–25% speedup at 0.25 pruning.

C Dataset Summary for Models and Benchmarks

Table 7 summarizes the benchmark suite used in our experiments, including dataset name, release year, task description, and shot setting. The protocol covers diverse capability axes—language modeling, commonsense and scientific reasoning, coding, mathematical reasoning, multi-hop retrieval, and long-document understanding—to evaluate pruning robustness across domains. Most benchmarks are evaluated in 0-shot to measure retained capability after pruning, while selected reasoning-intensive tasks (e.g., MMLU, ARC-Challenge, GSM8K, and HellaSwag) use few-shot settings following standard LLM practice. This breadth supports the consistency of our pruning observations across evaluation formats. For long-context evaluation, we follow LongBench and report HotpotQA, QASPER, and GovReport, covering both short- and long-context scenarios.

D Efficiency Analysis under Layer-level Pruning

Inference Latency. Figure 8 compares wall-clock inference latency between original and pruned models under layer-level pruning across sequence lengths from 1,024 to 65,536 tokens. We evaluate Llama-3.1-8B, Qwen3-4B, and Gemma-3-4B at pruning ratios of 0.125 and 0.25. Across all models and sequence lengths, our approach consistently reduces inference time, delivering approximately 7–13% speedup at a pruning ratio of 0.125 and 20–25% speedup at 0.25. These results indicate that the proposed layer-level pruning strategy provides robust and scalable latency gains, especially in long-context inference settings where computational overhead is most substantial.

Parameter Efficiency. Table 8 summarizes parameter efficiency under layer-level pruning by reporting the total number of parameters (in billions) before and after pruning for Llama-3.1-8B, Qwen3-4B, and Gemma-3-4B. Our method consistently reduces parameter counts across all models, with reductions of approximately 9–11% at a pruning ratio of 0.125 and 18–22% at a pruning ratio of 0.25. These results confirm that layer-level pruning yields substantial model compression while preserving a consistent reduction trend across different model families.

Model	Pruning Ratio	Original (B)	Pruned (B)	Reduction (%)
Llama-3.1-8B	0.125	8.03	7.16	10.86
	0.25	8.03	6.29	21.73
Qwen3-4B	0.125	4.02	3.62	10.04
	0.25	4.02	3.11	22.58
Gemma-3-4B	0.125	4.30	3.92	8.78
	0.25	4.30	3.55	17.56

Table 8: Parameter efficiency of our method under layer-level pruning. We report the total number of parameters (in billions) before and after pruning for Llama-3.1-8B, Qwen3-4B, and Gemma-3-4B. Our method achieves parameter reductions of approximately 9–11% at a pruning ratio of 0.125 and 18–22% at a pruning ratio of 0.25.

Model	Pruning Ratio	BOS–Wanda	BOS–Mag	Wanda–Mag
Gemma-3-4B	0.125	0.2188	0.1250	0.3438
	0.25	0.3750	0.1354	0.3125
Gemma-3-12B	0.125	0.3043	0.1159	0.2246
	0.25	0.3992	0.1708	0.2741
Llama-3.1-8B	0.125	0.3803	0.0810	0.1384
	0.25	0.4679	0.1128	0.1913
Qwen3-4B	0.125	0.4445	0.1045	0.1297
	0.25	0.4851	0.1433	0.1755

Table 9: Pairwise IoU overlap of pruned attention-head sets across <BOS> -based pruning, Wanda-SP, and Mag-SP under component-level pruning.

E Extended IoU Analysis

To further characterize what our criterion removes, Table 9 reports pairwise IoU overlap between pruned head sets from <BOS>-based pruning, Wanda-SP, and Mag-SP. A consistent pattern emerges across all model families and sparsity settings: overlap between our method and Mag-SP remains low, while overlap with Wanda-SP is moderate. This indicates that our method is not a simple proxy for weight magnitude and only partially aligns with activation-based saliency. In other words, <BOS> sink scores identify a structurally different subset of redundant heads that conventional criteria do not fully capture. Even when sparsity increases from 0.125 to 0.25 and overlap values rise overall, this relative ordering is preserved, reinforcing that our method captures complementary redundancy signals rather than reproducing baseline selections.

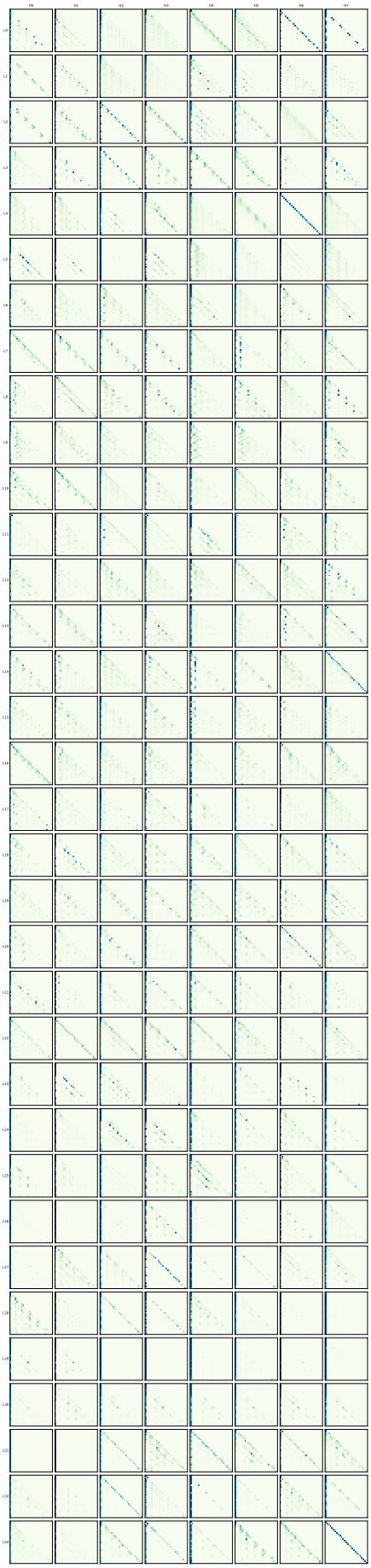

Figure 9: Full Attention Maps for Gemma-3-4B.

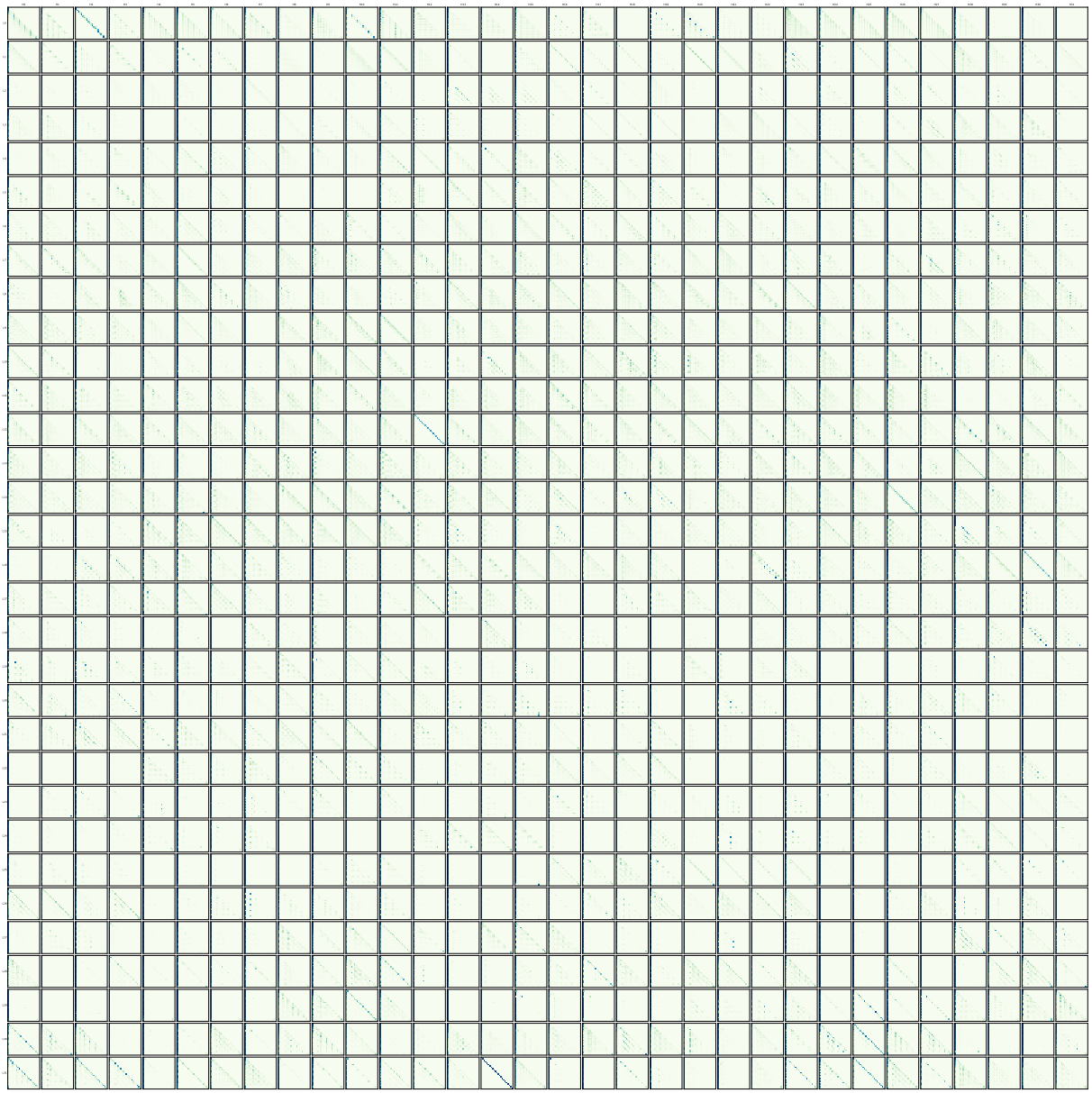

Figure 10: Full Attention Maps for Llama-3.1-8B.

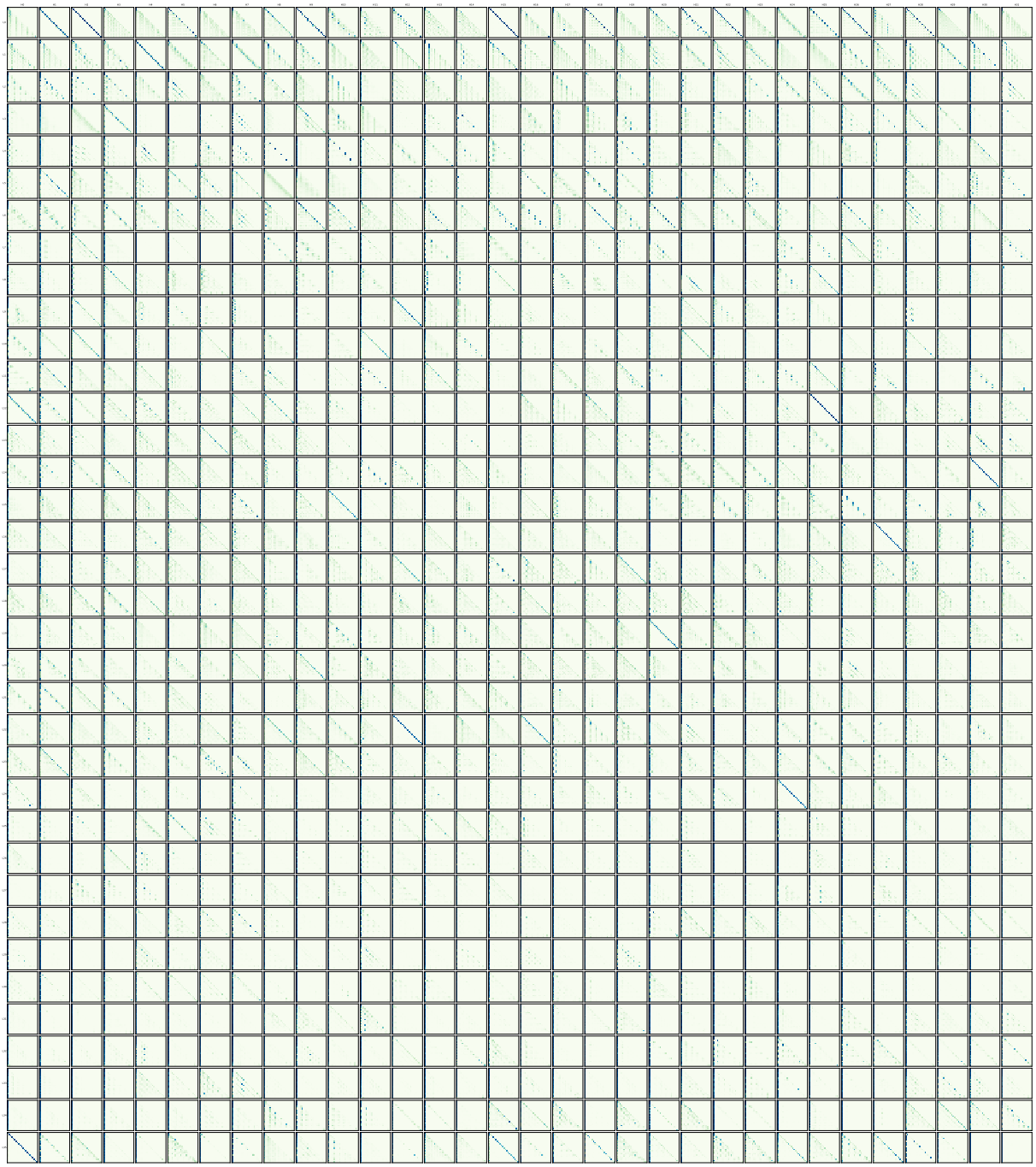

Figure 11: Full Attention Maps for Qwen3-4B.