Attention Sink Is Sinking Causality: Causal Interpretation of Self-Attention in Decoder Language Models and Mitigating Attention Sink for Improved Interpretability

Anonymous ACL submission

Abstract

Self-attention is widely regarded as a key mechanism enabling Transformers to dynamically focus on relevant input tokens. However, this focusing process can become distorted by **attention sinks** (Xiao et al., 2024)—tokens such as the beginning-of-sequence marker or other function words that receive disproportionately high attention weights despite offering minimal semantic contribution. In this paper, we study the causal significance of self-attention in decoder-based Large Language Models (LLMs) for classification tasks, with a particular emphasis on how these attention sinks impact interpretability. We first document the prevalence of attention sink across diverse sentiment and short-prompt classification datasets, revealing that seemingly crucial tokens often have little causal influence on final predictions making it hard to interpret the LLM's thereby making them a blackbox models. We then propose and evaluate mitigation strategies—such as reweighting the attention distribution to reduce the effect of attention sinks. Empirical results show that these techniques improve alignment between attention weights and truly influential tokens, thereby enhancing the causal interpretability of the self-attention mechanism. Our findings underscore the importance of identifying and alleviating attention sinks, particularly for applications where transparent and trustworthy model explanations are paramount.

1 Introduction

What are decoder language models actually looking at (or attending to) when predicting a class label? In principle, the self-attention mechanism in Transformer decoders (Vaswani et al., 2023) is designed to dynamically focus on the most relevant parts of an input sequence. However, in practice, we observe a phenomenon we call an **attention sink** (Xiao et al., 2024): certain tokens—such as the beginning-of-sequence marker, prompt words, or other non semantically important tokens—attract disproportionately high attention weights, even though they contribute little semantically. This issue has been noted in prior work (Xiao et al., 2024; Yu et al., 2024), raising the question of whether high attention weights truly reflect a token's causal role in model decisions.

This paper starts by asking three fundamental questions:

- **Which tokens does the model attend to in order to predict a class?**
- **Are the tokens with the highest attention scores semantically meaningful, or are they merely attention sinks?**
- **Does mitigating attention sinks improve the causal interpretability of decoder language models and, in turn, their explainability?**

To illustrate these points, we used several architectures of decoder LLM's and for a classification prompt we average the attention scores of all the heads of last layer of LLM and extract the scores for the last token. Figures 1 and 2 show the contrast between raw attention distributions and those after sink mitigation. In Figure 1, the model's attention is heavily skewed towards starting and some function tokens. In contrast, Figure 2 demonstrates that once the attention sink is mitigated, the true meaningful and semantically critical tokens that drive the classification decision are reflected from the attention distribution which helps a lot in interpreting LLMs and explaining there prediction.

By addressing these questions and leveraging the insights provided by the attention visualizations, we investigate the causal significance of self-attention in decoder-based LLMs for classification tasks. We further propose mitigation techniques—such as reweighting the attention distribution using entropy and/or Z-scores to mitigate the

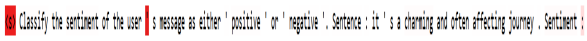

Figure 1: Raw attention distribution illustrating the *attention sink* phenomenon, where prompt tokens and function words receive disproportionately high attention.

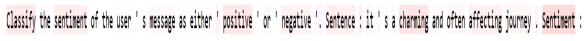

Figure 2: Attention distribution after mitigating attention sinks. The model now focuses on the key sentiment-bearing tokens that are causally relevant to the classification decision.

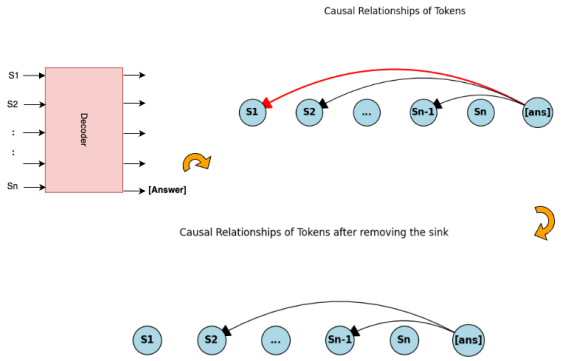

Figure 3: End-to-end pipeline.The red arrow marks an attention sink; black arrows indicate causal paths.

influence of attention sinks. Our empirical results indicate that these techniques realign attention with the truly influential tokens, thereby offering more transparent and trustworthy explanations for model predictions.

2 Experimental Setup

In Figure 3, we illustrate the pipeline used to investigate the causal influence of tokens in a decoder-based LLM during classification. We begin by feeding a prompt with n tokens, $\{S_1, S_2, \ldots, S_n\}$, into the decoder. The model then produces an output tokens, denoted as `[ans]`. As depicted by the red arrow (the *attention sink*), certain tokens—often the first token or special symbols—can disproportionately attract the highest attention weights despite being semantically meaningful. Meanwhile, the black arrows represent the *causal paths* or influential tokens that genuinely drive the classification decision. By measuring how attention is distributed across the input tokens and contrasting it with the causal impact of masking or removing some sink tokens, we can diagnose and mitigate attention sink effects. This process allows us to better interpret which tokens truly shape the model's final output, thereby offering a clearer view into the causal underpinnings of the self-attention mechanism for classification tasks.

2.1 Experimental Setup Evaluation

After applying our attention sink removal method, we evaluate the refined attention distribution by extracting the top K tokens (with $K = 10$) for each prompt which have the highest attention scores, this list is generated by prompting GPT-4o to suggest the most semantically informative tokens for each query in a dataset , we pre-define a collection of important tokens that capture key semantic features required for classification. For example, in the SST-2 dataset, this set includes tokens such as `classify`, `sentiment`, `positive`, and `negative`. We then match the extracted top-K tokens against these pre-specified sets separately for two cases: (i) the **positive** case, where the model's prediction is correct or matching the ground truth, and (ii) the **negative** case, where the prediction is incorrect or not matching the ground truth. We wanted to check if in the positive case does the model's attention is well-aligned with semantically meaningful tokens after removing the attention sink, thereby confirming the efficacy of our sink removal approach. Conversely, we wanted to check if in the negative case does misaligned attention is higher which implies that model is not attending to semantically meaningful tokens which lead to the poor downstream performance. This evaluation framework thus provides quantitative insight into how effectively the top attention tokens correspond to the tokens known to be causally influential in driving the model's decisions.

3 Tasks and Datasets

We focus on **short-passage classification** to reduce the computation costs easy experimentation and tasks that span various domains to do a robust analysis of our method. For choosing datasets and their label sets we followed the same setting as (Yu et al., 2024) paper's classification datasets. Which are: SST-2, SST-5, MR, SUBJ, DBPedia, AG News, trace1,trace2, CB, and BoolQ (Socher et al., 2013; HuggingFace contributors, Accessed 3 March 2025c,A,A; Lehmann et al., 2015; Hugging-

@]X@

Dataset	Label set x
SST-2	{positive, negative}
SST-5	{terrible, negative, neutral, positive, great}
MR	{positive, negative}
SUBJ	{subjective, objective}
DBPedia	14 classes (company, school, ..., book)
AGNews	{World, Sports, Business, Technology}
TREC	{Desc., Entity, Expr., Person, Number, Loc.}
CB	{Yes, No, Maybe}
BoolQ	{True, False}

Table 1: Label sets for the text-classification datasets used in our experiments.

Face contributors, Accessed 3 March 2025a; Li and Roth, 2002; Hovy et al., 2001; Wang et al., 2019; Clark et al., 2019). See Table 1 for an overview of labels and sources. We randomly sampled 900 samples from each dataset for our experements.

3.1 Prompt Templates and Label Choices

For each dataset, we provide an **instruction** (e.g., "Classify the sentiment into 'positive' or 'negitive') and a **template** (e.g., "Sentence: {`text`}\nSentiment: "), which frames the user input and the expected classification label. We specify the set of **label choices** (e.g., "positive", "negative"), which the model uses to predict the next token. These prompt templates and label sets are customized to each dataset, ensuring consistency in how we query our language model for classification.

3.2 Model and Attention Analysis

We use Llama-2-7B-Chat-HF (Llama, 2023), Mistral-7B-v0.1 (Mistral, 2023), DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025) where the later two have a mixture of experts type of architecture, we wanted to check if our method is robust to such architecture changes. We also observed that for our chat setting non instruction tuned models are not performing well for our classification tasks so we chose the instruction tunded models. We extract self-attention scores from the last decoder layers and average over all the attention heads. We extract the attention scores of all the tokens for the last token, that is which tokens were more attended to? to produce the answer? The experimental pipeline:

1. **Query a sentiment-related sentence:** Example: *"The movie was surprisingly enjoyable."*
2. **Extract token-wise attention weights** from the final layers.
3. **Identify key tokens influencing sentiment decisions before and after sink mitigation.**

4 Sink Detection Overview

4.1 Preliminaries: Last-Token Attention

Let the input be a sequence of n tokens:

$$\mathcal{S} = (s_1, s_2, \ldots, s_n),$$

and suppose a Transformer decoder of L layers with H heads per layer. Denote by

$$A_{n,i}^{(L,h)}$$

the attention weight from the *last token* s_n (query) to token s_i (key) in the h-th head of the L-th (final) layer. As usual:

$$A_{n,i}^{(L,h)} = \mathrm{softmax}_i\Big(\frac{(\mathbf{W}_Q^{(L,h)}\,\mathbf{h}_n^{(L)})^\top(\mathbf{W}_K^{(L,h)}\,\mathbf{h}_i^{(L)})}{\sqrt{d_k}}\Big).$$

Average Over Heads: We aggregate across heads:

$$a_i = \frac{1}{H}\sum_{h=1}^{H} A_{n,i}^{(L,h)}, \quad i = 1, \ldots, n.$$

Thus $\mathbf{a} = (a_1, \ldots, a_n)$ forms a valid probability distribution over the n tokens as they sum to 1:

$$\sum_{i=1}^{n} a_i \;=\; 1.$$

4.2 Drawbacks of Simple Mean-Thresholding

A common approach is to define a threshold $\theta \cdot \mu$, as suggested by (Yu et al., 2024), where

$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i,$$

and if $a_i > \theta\mu$, we call s_i a "sink." While easy to implement and computationally less expensive in practice when we tested on numerous examples we had to change the threshold hyperparameter differently for each test case to get meaningful causal interpretation after removing the sink attention. So, below, we outline more flexible methods.

4.3 Entropy-Based Sink Detection

Overall Distribution Entropy

we use the *Shannon entropy* of the attention distribution $\mathbf{a}$:

$$H(\mathbf{a}) = -\sum_{i=1}^{n} a_i \log(a_i),$$

A very *low* overall entropy often indicates a highly peaked attention distribution (i.e., only a few tokens receive large weights, potentially indicating a sink).

Change in Entropy When Removing a Token

One way to detect if a specific token s_i is a "sink" is to see how *removing* it from the distribution changes the overall attention *entropy*.

1. Define a *masked* attention vector

$$\tilde{a}_j^{(-i)} = \begin{cases} 0, & \text{if } j = i, \\ a_j, & \text{otherwise.} \end{cases}$$

2. Normalize to create a valid probability distribution:

$$\hat{a}_j^{(-i)} = \frac{\tilde{a}_j^{(-i)}}{\sum_{k=1}^{n} \tilde{a}_k^{(-i)}}.$$

3. Compute the new entropy:

$$H(\mathbf{a}^{(-i)}) = -\sum_{j=1}^{n} \hat{a}_j^{(-i)} \log\big(\hat{a}_j^{(-i)}\big).$$

4. Let

$$\Delta H_i = H(\mathbf{a}) - H(\mathbf{a}^{(-i)}).$$

If $\Delta H_i \ll 0$ (i.e., removing token i *increases* entropy a lot), or $\Delta H_i \gg 0$, either scenario can reveal outlier behavior. A large positive ΔH_i means that removing i *destroys* the distribution's focus, suggesting i is crucial to the current peaked distribution (potentially a sink). On the other hand, a large *negative* ΔH_i (rare but can happen if renormalization yields an even spikier distribution) also flags abnormal distribution changes. One may define a suitable threshold on ΔH_i to identify sinks.

4.4 Z-score Based Sink Detection

This approach is to identify "sinks" as statistical outliers in the attention vector:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i, \quad \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(a_i - \mu)^2.$$

Define a *z-score* for each token i:

$$z_i = \frac{a_i - \mu}{\sigma}.$$

If z_i exceeds some threshold α, s_i is considered a sink:

$$\text{sink if } z_i > \alpha.$$

Unlike raw mean-thresholding, this approach accounts for *variance* in the attention distribution.

4.5 Sink Removal & Normalization

Regardless of how sinks is detected (entropy, z-score, or/and causal masking), the removal and normalization step follows:

- **Mask identified sinks:**

$$\tilde{a}_i = \begin{cases} 0, & \text{if } s_i \text{ is a sink}, \\ a_i, & \text{otherwise.} \end{cases}$$

- **Renormalize:**

$$C = \sum_{i=1}^{n} \tilde{a}_i, \quad b_i = \frac{\tilde{a}_i}{C}.$$

- The vector $\mathbf{b} = (b_1, \ldots, b_n)$ is now your "sink-free" attention distribution for interpretation.

Algorithm 1 Detect–Mask–Renormalise Attention Sinks

Require: Attention vector $\mathbf{a} = (a_1, \ldots, a_n)$, method $\mathsf{m} \in \{\text{Entropy}, \text{Z-score}\}$, threshold τ

$\mathcal{I}_{\text{sink}} \leftarrow$
for $i \leftarrow 1$ **to** n **do**
  if $\mathsf{m} = \text{Entropy}$ **then**
    $\Delta H_i \leftarrow H(\mathbf{a}) - H\big(\text{Renorm}(\mathbf{a}_{\setminus i})\big)$
    if $|\Delta H_i| > \tau$ **then**
      $\mathcal{I}_{\text{sink}} \leftarrow \mathcal{I}_{\text{sink}} \cup \{i\}$
    end if
  else if $\mathsf{m} = \text{Z-score}$ **then**
    $z_i \leftarrow (a_i - \mu)/\sigma$ ▷ μ, σ pre-computed once
    if $z_i > \tau$ **then**
      $\mathcal{I}_{\text{sink}} \leftarrow \mathcal{I}_{\text{sink}} \cup \{i\}$
    end if
  end if
end for
for $i \leftarrow 1$ **to** n **do**
  if $i \in \mathcal{I}_{\text{sink}}$ **then**
    $\tilde{a}_i \leftarrow 0$
  else
    $\tilde{a}_i \leftarrow a_i$
  end if
end for
$\mathbf{b} \leftarrow \tilde{\mathbf{a}} \big/ \sum_j \tilde{a}_j$ ▷ renormalise
return $\mathbf{b}$ ▷ sink-free attention distribution

5 Mitigating Attention Sink for Causal Interpretability

5.1 Reweighted Attention Scaling

We introduce a **normalization factor** to redistribute attention weights:

$$A'_{ij} = \frac{A_{ij}}{\sum_k A_{ik}} \tag{1}$$

where A'_{ij} represents the normalized attention weights.

6 Results and Discussion

As shown in Table 2, our experiments reveal several insights:

- **Overall Performance.** Llama achieves the highest accuracy on sentiment tasks (e.g., 91.2% on SST-2), suggesting that it is better at identifying polarity cues than Mixtral and Deepseek. However, on TREC question-type classification, all models underperform, indicating that short-prompt classification remains challenging for fine-grained tasks.

- **Mismatch vs. Match.** The "Match" metric consistently exceeds the "Mismatch" metric. For example, on SST-2, Llama's match score is 0.91 compared to a mismatch score of 0.45. This gap implies that correctly predicted examples exhibit stronger alignment between attention distributions and semantically salient tokens, whereas misclassifications often correspond to diffuse or misaligned attention. This also leads to another question can we calibrate the model's attention distribution to focus on important tokens to improve the downstream accuracy of the model? This could be a good direction for researchers.

- **Attention Sink Removal.** Mitigating attention sinks significantly improves interpretability. After removing highly attended but semantically uninformative tokens (e.g., prompt markers, punctuation), the attention distribution re-focuses on content-rich words in the question (e.g., "Classify," "Sentiment"). This realignment is evident in the higher Match scores across all datasets.

- **Model-Specific Patterns.** Although Llama generally excels at sentiment classification, Mixtral outperforms it on AGNews (83.1% vs. 78.3%), and Deepseek attains a leading accuracy on BoolQ (86.1%). These findings suggest that attention sink removal aids interpretability but does not fully explain domain-specific variations in model performance, which likely stem from architectural and training differences.

Overall, reducing attention sinks clarifies the model's decision pathway by emphasizing tokens with genuine causal influence, thus offering more transparent and trustworthy explanations.

7 Conclusion and Future Directions

Our empirical observations indicate that, for many classification inputs, the **final token often devotes large fractions of its attention** to **initial tokens** (such as the beginning-of-sequence token) or punctuation. This **attention sink** effect can overshadow genuinely semantic tokens e.g.,and thereby obscures which tokens truly contribute to classifi-

	Llama			Mixtral			Deepseek		
Dataset	Acc.	Mismatch	Match	Acc.	Mismatch	Match	Acc.	Mismatch	Match
SST-2	91.2%	0.45	0.91	54.1%	0.40	0.85	48.3%	0.35	0.90
SST-5	45.3%	0.28	0.85	44.7%	0.22	0.82	44.1%	0.26	0.85
MR	90.2%	0.48	0.83	86.0%	0.43	0.85	89.9%	0.17	0.81
AGNews	78.3%	0.27	0.65	83.1%	0.30	0.68	88.2%	0.25	0.60
TREC	12.2%	0.25	0.74	23.1%	0.28	0.75	20.0%	0.28	0.73
CB	67.8%	0.27	0.65	70.1%	0.28	0.75	65.9%	0.26	0.75
BoolQ	79.2%	0.47	0.73	84.0%	0.53	0.73	86.1%	0.48	0.72

Table 2: Evaluation of attention sink removal across three decoder architectures. For each dataset, we report (i) Accuracy, (ii) the average keyword match score on mispredicted examples (Mismatch), and (iii) the average keyword match score on correctly predicted examples (Match).

cation decisions. When we adjust or "reweight" attentions to reduce sink-token's dominance, we observe that **more relevant sentiment cues receive meaningful attention weights, helping us to interpret what the model is attending to produce answer**

These results underscore a potential misalignment between raw attention distributions and causally important tokens: the model may "look at" function tokens and special symbols, even though removing those tokens has little impact on the output label. Consequently, standard attention visualizations alone can be misleading for causal interpretability. To mitigate this, we propose:

By incorporating these steps, we find that **attention sink is reduced**, revealing more consistent correspondence between high-attention tokens and classification-critical words. Future work will investigate a broader range of transformer layers and heads separately and explore how calibrating attention sinks to focus on semantically meaningful tokens improves the downstream performance.

References

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. 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 (NAACL-HLT)*, pages 2924–2936.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

HuggingFace contributors. Accessed 3 March 2025a. AG News dataset on huggingface. https://huggingface.co/datasets/ag_news.

HuggingFace contributors. Accessed 3 March 2025b. Rotten Tomatoes dataset on huggingface. https://huggingface.co/datasets/rotten_tomatoes.

HuggingFace contributors. Accessed 3 March 2025c. SST5 dataset from the setfit library. https://huggingface.co/datasets/SetFit/sst5.

HuggingFace contributors. Accessed 3 March 2025d. SUBJ dataset from the setfit library. https://huggingface.co/datasets/SetFit/subj.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Chris Bizer. 2015. DBpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Llama. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Mistral. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier

benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3266–3280.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. *Preprint*, arXiv:2309.17453.

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

Limitations

Our study is confined to English, short-passage classification with three open-weight 7-B-parameter decoder-only models; we therefore do not know how attention-sink behaviour—or our fixed entropy/z-score thresholds—will transfer to other languages, long-context generation, encoder–decoder architectures or much larger frontier models. The causal-alignment metric relies on small, hand-curated keyword lists that introduce human bias, and itself is a post-hoc re-weighting that offers probabilistic cues rather than guaranteed causal explanations, leaving it vulnerable to adversarial prompts and unsuitable as the sole basis for high-stakes decisions. We inherit demographic and topical skews present in SST-2, AG News and related corpora, and although the extra computation is minimal (5 ms per sample), we did not measure full life-cycle carbon costs. Addressing these gaps—multilingual and long-form coverage, unbiased evaluation, training-time mitigation and broader ethical audits—remains essential future work before real-world deployment.

Acknowledgments

We employed an AI writing assistant (OpenAI ChatGPT) solely for copy-editing early drafts like tightening phrasing and fixing LaTeX syntax; all technical ideas, experiments, analyses and final wording were authored and verified by the paper's human authors.