SINK BEFORE YOU SPEAK: CONNECTING LANGUAGE GENERATION, ATTENTION PATTERNS, AND REPRESENTATIONAL GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have demonstrated an impressive capability to generate coherent text that is relevant to the subject matter provided in a user's prompt. Despite their sophistication, LLMs have also been shown to generate completely irrelevant text or strings of repeating, relevant tokens both in naturalistic settings and through adversarially constructed prompts in natural language. While some research has been done into the mechanistic cause for these sorts of generations, these investigations provide incomplete explanations on account of considering these phenomena independently. In this work, we provide a unified account for how both relevant, repetitive and coherent, irrelevant generations arise from specific activation patterns in the attention mechanisms of these models. This explanation centers on the attention sink, the emergent behaviour where high softmax attention values are assigned to seemingly mundane tokens such as the start-of-sentence delimiter. We evidence this explanation by directly modifying the attention maps and performing a novel perturbation to the latent representations. These causal interventions not only reliably induce the types of language generation we are interested in, but also the attention patterns we hypothesize to be at fault for them. We finally use this per-prompt spectral approach to characterize the operation of the attention sink. The findings of this work give a more complete understanding of the operation of LLMs, as well as how models that are seemingly steeled against existing jailbreaks remain susceptible to the shortcomings of the transformer architecture.

1 INTRODUCTION

Large Language Models (LLMs) have had an enormous impact on several industries, academia, and even society at large due to their ability to create coherent, relevant text given a wide variety of prompts (Brown et al., 2020; Bin-Nashwan et al., 2023; Holmes & Tuomi, 2022; McElheran et al., 2024). Despite working well in many contexts, LLMs have been shown to produce unhelpful responses, such as repetitive "loops" of text with no apparent cause (Yao et al., 2025; Su et al., 2025). Adversarially crafted prompts or "jailbreaks" can cause other harmful behaviours (Yan et al., 2025), such as leaking training data (Nasr et al., 2023). While some protective measures for reducing the efficacy of jailbreaks have been proposed (Nasr et al., 2023; Fairoze et al., 2026), the underlying mechanisms behind these jailbreaks, i.e. how they influence a model's internal activations, are poorly understood. Even when explanations are proposed, they typically stop short of characterizing the kind of undesirable generation the jailbreak will produce or linking this explanation to other related jailbreaks. In this work, we characterize two jailbreaks that produce qualitatively different output text in terms of the patterns in their attention maps. We then provide an explanation in the form of sufficient causal interventions on the attention patterns and hidden representations of two distinct LLMs that induce these patterns and generate similar text. In some cases, we find that models that are robust to a certain jailbreak are still susceptible to our interventions, revealing a dormant vulnerability.

Several works in recent years have investigated the attention sink phenomenon (Xiao et al., 2023; Cancedda, 2024), where the attention mechanism in transformers seems to place a large attention value on uninformative tokens such as the start of sentence tag. Attention sinks have been shown to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

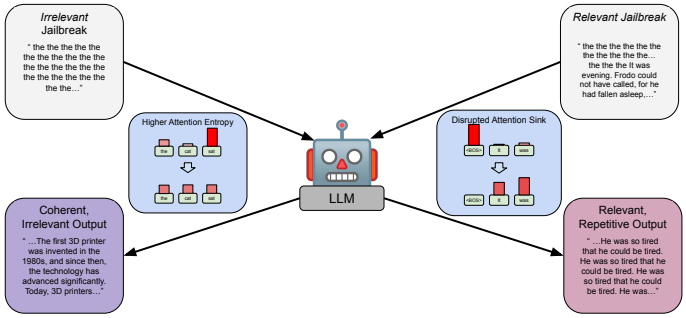


Figure 1: We investigate two kinds of jailbreaks: one that induces *relevant, repetitive* generation and one that induces *coherent, irrelevant* generation. We characterize both of these undesirable generations to a drop in the attention sink, while coherent, irrelevant generations also show a distinct increase in attention entropy.

be useful in the normal functioning of the model (Zhang et al., 2025; Cancedda, 2024; Yona et al., 2025). Furthermore, some geometric characterization of this phenomenon has been presented, with routine demonstrations that attention sink tokens make use of "outlier features", large magnitude directions in the residual stream that appear as top singular vectors in the singular value decomposition of representations (Queipo-de Llano et al., 2025; Zhang et al., 2025; Yona et al., 2025). However, the exact geometric mechanisms behind the appearance of attention sinks at various layers, how attention sinks and other attention map patterns relate to and potentially cause different types of generations, and how robust these models are to the loss of their attention sink are all poorly understood. In this work, we investigate these questions, and relate the attention sink to multiple jailbreak strategies.

Our work sheds light on how the complicated representational geometry of an LLM interplays with the generations and internal activations they produce, and relates that to the real-world concern around LLM jailbreaks. We provide the following robust contributions with respect to both a Llama model (Dubey et al., 2024) and Mistral model (Jiang et al., 2024):

1. We provide characterizations of two separate jailbreak strategies in terms of their attention patterns. One strategy produces *coherent but irrelevant* generations, while the other produces *relevant but repetitive* generations, and for each we provide distinct characterizations relating to the attention sink. This extends existing work relating attention sinks to individual jailbreaks (Yona et al., 2025).

2. We use this mechanistic characterization to produce an explanation for these different types of model behaviour in the form of sufficient causal interventions. These interventions operate either on the attention maps from the model directly, or implicitly through the representations. Furthermore, we use a novel analytical strategy involving the per-prompt singular value spectrum of the representations as part of our intervention on the latent space. This contrasts with previous works that have focussed on the spectrum of representations collected from several prompts (Kirsanov et al., 2025; Merullo et al., 2025).

3. We use our per-prompt spectral approach to better characterize the operation of the attention sink in terms of the representational geometry. We present novel findings relating the attention sink to different directions in the top singular vector subspace and use this to demonstrate how an attention sink can be disrupted through either all or a subset of the layers of an LLM.

Each of these findings provides a deeper understanding of how LLMs function and how undesirable behaviours can be induced by manipulating the internal mechanisms of the model through the prompt or otherwise. These critical insights are valuable to those wishing to create safer, more controllable LLMs.

2 Related Work

Jailbreaking LLMs with Adversarial Prompts. The widespread adoption of AI systems in different aspects of society has motivated a great deal of research into how these systems can produce undesirable or even harmful outputs with normal operation over an adversarially chosen text prompt (Nasr et al., 2023; Fairoze et al., 2026; Lazaridis et al., 2026; Li et al., 2025; Breitenbach et al., 2023; Yan et al., 2025). Li et al. (2025) provide an overview of prefilling attacks, which prompt the model to produce some manner of dangerous output, and then prefill the prompt with text that resembles the model beginning to produce such output. Nasr et al. (2023) propose a data extraction attack where the model is asked to repeat a word indefinitely, causing the model to produce irrelevant text assumed to be from the training distribution. This repeated token attack has inspired several other jailbreaks, such as those discussed by Breitenbach et al. (2023) that repeat a token and then append a request to the model, which is then unsuccessfully answered in the output. In this work, we give a more exhaustive explanation for why the repeated token attack and other jailbreaks like it can yield both the divergent output initially reported by Nasr et al. (2023) but also the looping generations described in several other works (Yao et al., 2025; Su et al., 2025; Breitenbach et al., 2023).

Interpreting LLM Generation and Attention Sinks. Several researchers have worked to understand the patterns in model activations that give rise to undesirable and nonsensical kinds of text generated from an LLM. Yao et al. (2025) provide a thorough investigation of how LLMs can get caught in repetitive, looping generations. They propose that specific "repetition features" are responsible for this, and propose directly inhibiting these directions in representation space as a means for mitigating repetition. Following the output divergence from repeated token exploit discovered by Nasr et al. (2023), Yona et al. (2025) provide an explanation for this phenomenon: the repeated tokens cause the attention sink value to be weakened and repeated tokens to be over emphasized, bringing the model out of distribution. While this work, as well as the work of Su et al. (2025), establish how undesirable generations co-emerge with the loss of an attention sink due to an outside cause (repeating tokens in the prompt, compressing experts in a mixture-of-experts model, etc.), it is not established whether the loss of the attention sink on its own can induce an undesirable generation. Furthermore, it also remains unclear which circumstances would lead to a weakened attention sink yielding coherent, irrelevant text and when it would yield relevant, repetitive text.

A common technique used for understanding and manipulating the kinds of text an LLM can generate is the use of steering vectors (Panickssery et al., 2023; Tan et al., 2024; Siddique et al., 2026). Once identified, steering vectors can be added to the internal representations of an LLM to increase its likelihood of performing a certain behaviour, such as producing negatively biased responses (Siddique et al., 2026). In this work, we look into alternative means for manipulating model behaviour, allowing us to steer the model with respect to more abstract qualities such as repetitiveness, coherence, and relevance to the prompt.

Representational Geometry and Perturbations. Several previous works have taken steps towards understanding the role of representational geometry in the operation of LLMs (Park et al., 2023; 2024; Sarfati et al., 2024; Kirsanov et al., 2025; Ruscio et al., 2025; Li et al., 2026). Perhaps most similar to representation-based work in this paper are the work of Cancedda (2024) and Merullo et al. (2025). In both of these papers, the authors propose a means of relating the behaviour of LLMs to directions in representation space produced by singular value decomposition (SVD). In contrast to these works, we use the SVD of the matrix of representations associated with each prompt separately, as this geometric object best describes the relative importance of each direction for a specific generation rather than in aggregate. We also apply these analyses to rigorously study free form text produced by LLMs.

Some geometric properties of the attention sink have been presented in the literature. Attention sink token representations are shown to have massive norms (Zhang et al., 2025; Yona et al., 2025; Cancedda, 2024). Queipo-de Llano et al. (2025) demonstrate how the presence of these "outlier features" necessitates the emergence of a dominant singular value in the spectrum of representations in the residual stream. In this work, we further the geometric understanding of the attention sink, relating its function to different bands of the spectrum and demonstrating how it may rely on different subspaces at different layers.

3 JAILBREAKS FOR INDUCING UNDESIRABLE OUTPUTS

In this section, we investigate cases where jailbreak prompts cause LLMs to produce text that is in some way undesirable. We study both Llama-3.1-8B-Instruct (Dubey et al., 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2024) and find that despite the fact that the two jailbreaks we study have very similar forms, one will consistently output coherent, irrelevant text in the Mistral model, while the other jailbreak reliably causes relevant, repetitive outputs in both models. By analyzing the attention patterns within the models, we find a common signature: a reduction in attention sink value. We focus on undesirable outputs that have structural flaws in their syntax or semantics with respect to the prompt, as these apply to several different kinds of prompts. The first kind of outputs we are interested in are **coherent, but irrelevant**: the response is composed of readable text following some kind of grammar conventions, rather than randomly selected words, or characters. However, this response is not related to the content of the prompt. We are also interested in responses that are **relevant, but repetitive**: the response is related to the subject of the prompt, but it is composed of repeated sentences, sentence fragments, words or individual characters.

To understand the kinds of text being produced from these jailbreaks, we evaluate the outputs of these models with respect to three criteria: coherence, relevance, and repetitiveness. Doing this allows us to not only to find the rate at which coherent, but irrelevant and relevant, but repetitive text occurs, but also look into the generation rate for other kinds of text. Specifically, we also evaluate whether text is *good* (coherent, relevant, and not repetitive) or *bad* (neither relevant nor coherent, or relevant but not coherent or repetitive). Across the thousands of generations we produce, we use an automatic evaluation set up that leverages GPT5-Mini (Singh et al., 2025) as well as Llama-3.1-70B-Instruct (Dubey et al., 2024) to evaluate whether the response meets these criteria. For jailbreaks that produce slightly more complex outputs, such as the outputs that repeat one token and then become *coherent, but irrelevant*, we employ GPT-5.6-Terra. This approach is common but can produce errors (Gu et al., 2024; Szymanski et al., 2025). However, our qualitative judgement of these evaluations found them to be reliable. Details regarding this procedure are in Appendix A.1.

3.1 COHERENT, IRRELEVANT OUTPUTS FROM JAILBREAKS

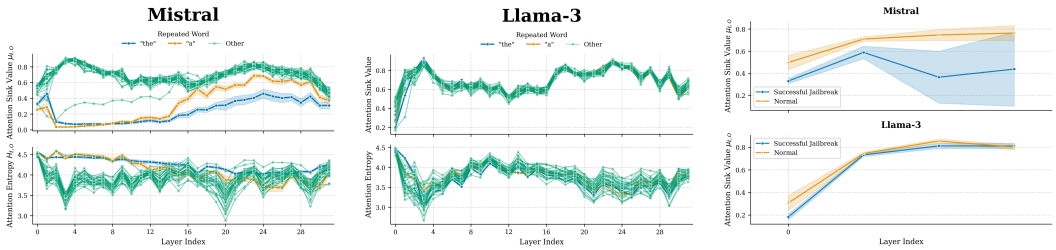
Nasr et al. (2023) discovered that large portions of an LLM's training data could be recovered from the model using an adversarial prompting strategy: simply repeating a single token several times. The divergent tokens in the output are clearly an example of *coherent, irrelevant text*, regardless of whether or not they are present in the model's training data. We investigate the rate at which this attack is successful against a series of different LLMs using different repeated words, producing 10 generations for each prompt. Due to its resultant output, we call this the *irrelevant* jailbreak.

We find that this jailbreak is very ineffective against Llama-3, with only one successful instance across 320 different generations. The majority of prompts are also ineffective against Mistral, with the exception of two. Both repeating "the" and repeating "a" cause the model to produce coherent, irrelevant text. Repeating "the" is the most successful, with a 60% success rate according to our automatic evaluation system.

We now look into the patterns of activation within these LLMs that give rise to these sorts of generations. We specifically look at the attention patterns for these models on different portions of the text: the attention sink, the prompt, and the output itself. Specifically, we will refer to the $(S+O)\times(S+O)$ attention map for attention head h at layer ℓ as $A^{h,\ell}$, and the index of the attention sink as γ. We calculate the average attention sink value across both the outputs and the prompt, and take the average of these across attention heads (where K is the total number of heads): $\mu_{l,S}$ and $\mu_{l,O}$. We would additionally like to understand the dispersion of attention values across different token regions. We define a new statistic $H_{\ell,O}$ which is the per-attention head average entropy of the attention values for the output tokens directed to the *non-attention sink* prompt tokens. This uses $\tilde{A}^{h,\ell}_{i,j}$, the normalized attention value for that row, where the normalization constant is the total attention value over non-attention sink prompt tokens. The definitions of each of these are:

$$\mu_{l,S} := \frac{1}{KS}\sum_{h=1}^{K}\sum_{i=1}^{S} A^{h,\ell}_{i,\gamma}, \quad \mu_{l,O} := \frac{1}{KO}\sum_{h=1}^{K}\sum_{i=S}^{S+O} A^{h,\ell}_{i,\gamma}, \quad H_{\ell,O} := \frac{1}{KO}\sum_{h=1}^{K}\sum_{i=1}^{O}(-\sum_{j=\gamma+1}^{S} \tilde{A}^{h,\ell}_{i,j}\log\tilde{A}^{h,\ell}_{i,j}). \tag{1}$$

Figure 2a demonstrates that instances where this jailbreak produces a coherent, irrelevant generation are characterized by two activation patterns: a sustained lower attention sink value and a notably higher entropy distribution of attention over the tokens in the prompt. Notably, Llama-3 never presents this pattern of activation and also is incredibly robust to this jailbreak (Figure 2b).



(a) (b) (c)

Figure 2: (a, b) Attention patterns for the irrelevant jailbreak examples. “Other” refers to a collection of 30 other irrelevant jailbreak prompts. Repeating ”the” or ”a” results in diverging text in Mistral. This is accompanied by a lower attention sink value and higher entropy of the attention distribution over the prompt. Llama-3 is robust to all words tried for this attack, and also never demonstrates this pattern. (c) Attention sink values for successful relevant jailbreak instances as well as 200 unperturbed prompts. We see that Mistral has a significant decrease in the attention sink value when processing a relevant jailbreak prompt.

3.2 Relevant, Repetitive Generations From Jailbreaks

We next propose a variant on the previous jailbreak similar to the one proposed by Breitenbach et al. (2023), where a regular text prompt is appended to a repeated token. While this jailbreak was described to produce text that did not properly respond to the appended prompt, we find that it reliably produces *relevant, repetitive text*. Consequently, we refer to it as the *relevant* jailbreak.

We randomly sample a prompt from each of four datasets (Appendix A.2) and append these to the two most successful jailbreak prompts from Section 3.1, repeating “the” and “a”. Performing 10 generations per prompt, we find that this approach is successful at producing relevant, repetitive outputs. Specifically, we find small strings of a few tokens related to the prompt appended to the jailbreak repeated in the output several times. This approach yields 78.8% and 56.3% success rates for Mistral and Llama-3, respectively.

To analyze the attention patterns, we compare with 200 randomly sampled prompts from 4 different datasets (Appendix A.2). We see in Figure 2 that the relevant jailbreak, when successful, results in significantly lower attention sink values for the output tokens in Mistral in the earlier layers. Meanwhile, though there is a decreased attention sink value in Llama-3.1 at the first layer, the difference in attention sink value is less pronounced at later layers (Appendix A.4). In both models, the entropy of the attention distribution over the prompt shows no significant difference between successful jailbreaks and typical prompts (Appendix A.4). However, these results indicate that disabling the attention sink can lead to degenerate outputs whose characteristics are dependent on the other tokens in the prompt.

4 Causal Intervention for Inducing Undesirable Outputs

In this section, we test our hypotheses on the role of the attention sink and other facets of the attention maps in producing undesirable generations through direct manipulation of the model’s activations. We first do this through direct manipulation of the attention maps at specific layers. We then circumvent the shortcomings of this approach using a representational projection strategy based on the singular value spectrum of each prompt. Throughout this section, we evaluate the characteristics of the generated text through the same autoeval setup as described in Section 3.

4.1 Directly Modifying Attention Patterns

Based on the results of Section 3, we hypothesize that disabling the attention sink can lead to degenerate outputs that are structured either by their cohesiveness or by their relevance to the prompt. We use the same set of 200 randomly sampled prompts as mentioned in Section 3.2, and manipulate the attention patterns in two ways:

- **Disabling Attention Sink (DAS)**: we set the pre-softmax attention logit for the attention sink to a high magnitude negative number for each attention head. This is applied at layers 2,3, and 4. Based on our results of Section 3.2, we hypothesize this will produce **relevant, repetitive text**.
- **Disabling Attention Sink + High Prompt Entropy (DAS + E)**: in addition to the operation described above, we uniformly set the logits for the prompt tokens to the minimum amongst the logits. This is applied at the first four layers. The high entropy demonstrated in Figure 2 suggests this will produce **coherent, irrelevant text**.

We present the efficacy of these approaches in Figure 4b. We see that while both DAS and DAS + E increase the rate at which the desired kind of text is produced in comparison to the unperturbed baseline, they also produce a large amount of either *good* responses in the case of DAS, or *bad* responses in the case of DAS + E. Notably, the success rates for these modifications differ between the models. DAS produces relevant, repetitive text 26.1% and 61.5% of the time for Mistral and Llama, respectively. Meanwhile, DAS + E produces coherent, irrelevant text 42.8% and 16.7% of the time for Mistral and Llama, respectively. While the stochasticity of decoding means that these perturbations always have some likelihood of producing *good* or *bad* text, the varying performance across models likely indicates that direct attention sink manipulation only explains some of this phenomenon.

4.2 Manipulating Representation Geometry with Per-Prompt Representation Spectra

The direct attention manipulation in Section 4.1 shows some promise in spurring the model to produce undesirable outputs in the same way as the jailbreaks presented in Section 3. However, the varying degrees of success of these interventions between models indicate that the attention patterns in isolation do not cause this behaviour. However, we theorize that the high entropy attention distribution over the input tokens themselves (Appendix A.5) will influence the resulting updates to the residual stream. Specifically, repeated updates from uniform attention to previous tokens will quickly result in tokens later in the sequence collapsing to a uniform representation (Barbero et al., 2024). If a decoded output token is not similar to this uniform, repeated representation, it will likely attend only to other output tokens and lead to an irrelevant generation. This motivates us to experiment with intervening on the model's representations in such a way that allows us to smoothly modulate the potential similarity between the output tokens and the input tokens. Towards this goal, we propose projecting out directions in representation space according to the singular value decomposition of the prompt tokens.

Denote the representations for a tokenized prompt $p := \{x_i\}_{i=1}^{S}$ of length S at layer ℓ as $p_{f_\ell} := \{f_\ell(x_i)\}_{i=1}^{S}$. A common way of characterizing the geometry of a pool of vectors is using the SVD of the matrix containing those vectors. Specifically, we can stack the vectors in p_{f_ℓ} into a matrix $P_{f_\ell} \in \mathbb{R}^{S \times D}$. SVD then decomposes the matrix into a product of three matrices: $P_{f_\ell} = U_{f_\ell} S_{f_\ell} V_{f_\ell}^{\top}$. Most important for understanding our analysis are the matrices $V_{f_\ell}^{\top}$, which has rows representing directions in representation space, and S_{f_ℓ}, which is a diagonal matrix with values representing how much of the representations' magnitude is along the corresponding direction in $V_{f_\ell}^{\top}$.

Seeing as SVD gives us an understanding of the directions in the representation space encompassing most of the magnitude for representations from a prompt, we use this as a means of examining an individual prompt's geometry. Specifically, we study the impact of ablating selected singular vectors from the representation space on the text generated by the model. Consider some subset of the rows of $V_{f_\ell}^{\top}$, $\Omega \subseteq Rows(V_{f_\ell}^{\top})$: we can use this set to create a new version of p_{f_ℓ} where the vectors in Ω

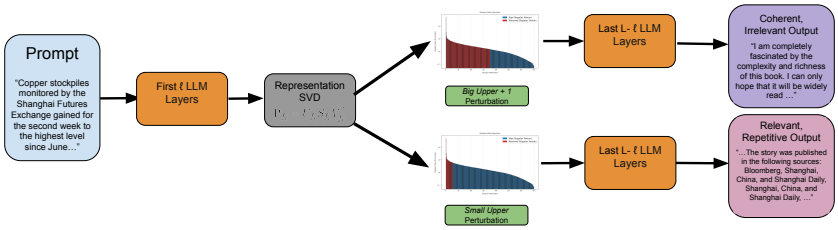


Figure 3: Representation perturbation pipeline and examples of output. Perturbing internal representations with specific SVD based projections yields generations with consistent characteristics.

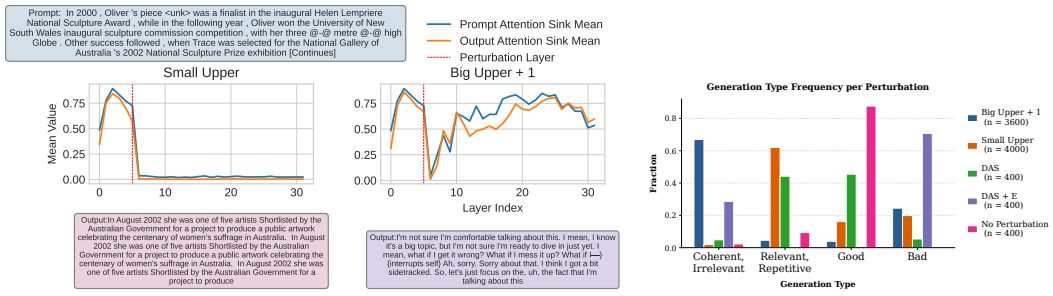
are projected out:

$$p_{f_\ell,\Omega} := \{f_\ell(f_{\ell-1}(x_i)) - \sum_{e\in\Omega} < e, f_\ell(f_{\ell-1}(x_i)) > \}_{i=1}^{S}. \tag{2}$$

These perturbed representations can be passed through the rest of the layers of the model to produce a new generation, potentially different from the unperturbed version. To select vectors for projection, we use the quantiles of the associated singular values. Specifically, for layer index ℓ and fractions $\alpha, \beta \in [0, 1]$, $\Omega_{\ell,\alpha,\beta}$ defines the set of singular vectors from layer ℓ that have singular values greater than α of the singular values and less than β of the singular values. We use these spectral bands to understand what directions in representation space of differing magnitudes account for in the operation of the model. This procedure is visualized in Figure 3.

Finally, we note that this is a departure from the usual application of SVD to a pool of representations from different prompts (Kirsanov et al., 2025; Merullo et al., 2025). SVD is meant to characterize directions along which representations have high magnitude, and while pooling representations may give a representative spectrum for the *hypothetical average prompt*, the spectrum of representations for a given prompt better characterizes these high magnitude directions for the current input.

4.2.1 Efficacy of Specific Perturbations



(a) (b)

Figure 4: **(a)** Mean attention sink values across attention heads ($\mu_{l,S}, \mu_{l,O}$) (Section 3.1) across layers for two perturbation regimes and their associated outputs to an individual prompt. While the kinds of generation produced vary, each regime results in a drop in the attention sink values. **(b)** Fraction of responses of each type described in Section 3 for both Llama and Mistral across three perturbation regimes, the two attention pattern interventions (Section 4.1), as well as the unperturbed baseline. Both of the perturbation regimes reliably generate a specific kind of text.

We now examine how using different perturbations can affect the outputs of an LLM. We specifically focus on perturbation strategies that disrupt the attention sink. Other works have suggested that attention sinks depend on a high-variance direction in latent space Zhang et al. (2025); Li et al. (2026); Yona et al. (2025), and so we target these high-singular value directions. After performing a selection process for the spectral perturbations on a separate set of data (Appendix A.2), we find the following perturbations to be especially effective:

- **Small Upper**: These perturbations target a very small band of the highest magnitude directions of the spectrum ($\ell \in [5, 9], \alpha \in [0.925, 0.99], \beta = 1.0$). This is intended to disrupt the attention sink while allowing the output tokens to attend to the input tokens, like the jailbreak from Section 3.2. These perturbations produce **relevant, repetitive text**.
- **Big Upper + 1**: We remove at least the top half of the highest magnitude directions, but keep the top singular vector intact ($\ell \in [5, 9], \alpha \in [0.3, 0.5], \beta = 1.0$). This perturbation is intended to not only weaken the attention sink, but also prevent the output from attending to the prompt. We do this by only maintaining small variance directions within the subspace defined by the prompt tokens. This is similar to what was observed in Section 3.1. These perturbations are the only ones which regularly produce **coherent, irrelevant responses**.

Figure 4b shows that *Big Upper + 1* produces more coherent, irrelevant responses than any other kind of response type, and it produces these responses more than any of our other generation regimes. Similarly, *Small Upper* produces relevant, repetitive responses in a similar manner. As a control, Figure 4b also shows that unperturbed models produce relevant, coherent, and non-repetitive responses (*Good*) most frequently. Finally, these perturbations have a controlled impact: each perturbation regime produces a specific type of generation more than any other kind of generation, particular more than corrupt generations that are neither relevant nor coherent, or relevant but not coherent or repetitive (*Bad*).

These interventions establish specific disruptions to the representation can induce generations similar to those seen in Section 3. These results notably reveal a lingering vulnerability within these models: while Llama was not susceptible to the irrelevant jailbreak seen in Section 3.1, our per-prompt spectral perturbation caused the model to produce these kinds of outputs with significant frequency. In the next section, we will investigate the representational geometry induced by these perturbations' effects the attention sink, establishing a link with the patterns we saw in Section 3.

5 Attention Patterns and the Per-Prompt Spectrum

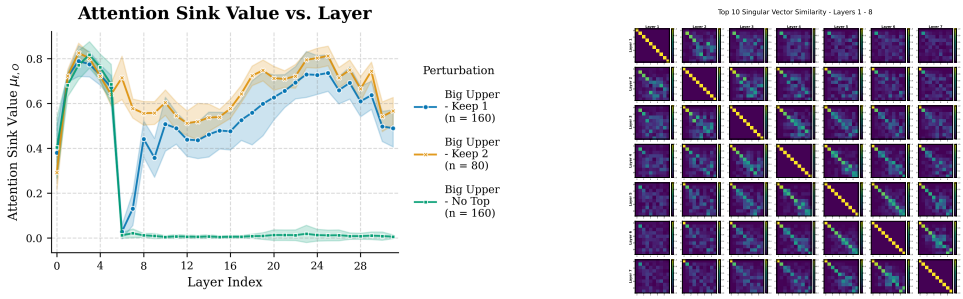
In this section we further analyze the attention patterns produced by perturbations using the per-prompt spectrum. We find that these effective perturbation strategies reduce the attention sink value, establishing a link to the attention patterns seen in successful jailbreaks in Section 3. We observe that these perturbations differ in the strength and duration of the attention sink disruption, and use the per-prompt spectrum to understand this variation in terms of the representational geometry.

5.1 Attention Sink Removal and Revival

We visualize the mean attention values ($\mu_{l,S}, \mu_{l,O}$) for a specific generation from each of the Big Upper + 1 and Small Upper perturbations in Figure 4a, and visualize the mean of these values across all generations for these perturbations in Appendix A.6. Both regimes result in a precipitous drop in the attention sink value. In Section 3, we saw that the jailbreaks also resulted in a drop in the attention sink value. Furthermore, we observe that Big Upper + 1 displays greater prompt attention entropy values than Small Upper (Appendix A.6). This is similar to the jailbreaks seen in Section 3.1, which also produced coherent, irrelevant outputs. The fact that these perturbations yield similar attention patterns and produce similar outputs as the jailbreaks provides further evidence linking these two phenomena together.

We can observe in Figure 4a that both perturbation types disrupt the attention sink value. However, Small Upper permanently removes the attention sink, while Big Upper + 1 only temporarily removes it. Both of these regimes project out large magnitude components of the representations, but Big Upper + 1 keeps the largest singular vector. To investigate how projecting out top singular vectors from the per-prompt spectrum influences the attention sink, we calculate mean $\mu_{l,O}$ values for three variants on a perturbation from Big Upper + 1 at layer 5: one where the first singular vector is maintained (as per usual), one where it is removed, and one where the first *and second* singular vectors are both maintained.

Figure 5a shows the mean difference in attention values across these three variants. We can again observe that keeping the first vector but removing the rest of the top space briefly removes the attention sink, while removing the entire top space including the first vector permanently removes



(a) (b)

Figure 5: **(a)** Mean Attention sink values ($\mu_{l,O}$) across layers calculated from different variations of the Big Upper + 1 perturbation at layer 5. How many top singular vectors are kept during the projection determines whether the attention sink value will never drop (keep 1 and 2), drop and return (keep 1), or drop and never return (keep neither). **(b)** Alignment (dot product) of the top 10 singular vectors across layers for one of the prompts. Alignment in the top singular vector is strong across all layers. Alignment in singular vectors 2-10 quickly drops as layers become more separated.

the sink. Meanwhile, keeping both the top two singular vectors but removing the rest of the top space still does not result in a weakening of the attention sink.

5.2 Top Subspace Alignment

Previous works have discussed the reliance of attention sinks on "outlier features" which occupy the top end of the singular vector spectrum (Zhang et al., 2025; Queipo-de Llano et al., 2025). However, it is not clear why a projection of all but one of the top singular vectors could remove the attention sink at one layer while allowing it to return at later layers. We theorize that this may be due to the top subspaces across layers being only *partially aligned*: while the full top singular vector space is necessary for the operation of the attention sink, only the top vector is shared across layers, implying the existence of layer-specific subspaces necessary for attention sinking. This would allow for a representation to accommodate attention sinking at a subset of the layers.

To test this hypothesis, we examine the alignment of the top 10 singular vectors for representations from different layers of Llama 3.1 for one of the prompts. Qualitatively, we can observe in Figure 5b that the top singular vector has strong alignment across all layers with the exception of the first. Meanwhile, the strength of the alignment for the remaining vectors quickly degrades, with the degree of degradation increasing for layers further apart. This supports our hypothesis for how the attention sink is able to be "revived", and further demonstrates the interesting phenomena that can be discovered using the per-prompt spectrum as an analytical tool.

6 Conclusion

LLMs can demonstrate a wide variety of failure modes, from repeating text to potentially leaking unrelated training data (Nasr et al., 2023). The widespread adoption of LLMs as well as the existence of textual jailbreaks posits a need for understanding these behaviours. However, understanding and ultimately mitigating these behaviours requires a stronger understanding of the model's attention patterns and representations and how they interact with the model's various designed and learned mechanisms. We contribute to the growing literature on this important topic by characterizing two LLMs' activation patterns with respect to two jailbreaks that induce different undesirable outputs. We find sufficient causal interventions for both these behaviours, some of which are based on a novel per-prompt singular vector projection strategy. This spectral analysis yields novel insights into different mechanisms within these models, such as geometrically characterizing the operation of the attention sink. This work is not without its limitations: it is not clear why the repetition of certain words leads to a successful jailbreak and others do not, and ultimately preventing this behaviour is an open problem. However, the findings presented in this work can be used towards these goals to design safer, more controllable AI systems.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic data sets (autoeval) and for writing code for generating plots.

We have not used generative AI tools for developing theoretical models or conceptual frameworks, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results.

Formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work.

Additionally, we used generative AI tools for finding relevant literature. We have reviewed all AI-assisted work, such as by inspecting examples from the autoeval dataset. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, João G Araújo, Alex Vitvitskyi, Razvan Pascanu, and Petar Veličković. Transformers need glasses! information over-squashing in language tasks. *Advances in Neural Information Processing Systems*, 37:98111–98142, 2024.

Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa-Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. Semeval 2018 task 2: Multilingual emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 24–33, 2018.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*, 2020.

Dan Benayoun. Processed dataset of financial news articles from bloomberg (2006-2013), 2024.

Saeed Awadh Bin-Nashwan, Mouad Sadallah, and Mohamed Bouteraa. Use of chatgpt in academia: Academic integrity hangs in the balance. *Technology in Society*, 75:102370, 2023.

Mark Breitenbach, Adrian Wood, Win Suen, and Po-Ning Tseng. Don't you (forget nlp): Prompt injection with control characters in chatgpt, 2023.

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

Nicola Cancedda. Spectral filters, dark signals, and attention sinks. *arXiv preprint arXiv:2402.09221*, 2024.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1415–1425, 2014.

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 e-prints*, pp. arXiv–2407, 2024.

Jaiden Fairoze, Sanjam Garg, Keewoo Lee, and Mingyuan Wang. Bypassing prompt guards in production with {Controlled-Release} prompting. In *35th USENIX Security Symposium (USENIX Security 26)*, pp. 2067–2086, 2026.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *The Innovation*, 2024.

Wayne Holmes and Ilkka Tuomi. State of the art and practice in ai in education. *European journal of education*, 57(4):542–570, 2022.

AQ Jiang, A Sablayrolles, A Mensch, C Bamford, DS Chaplot, Ddl Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. Mistral 7b. arxiv 2023. *arXiv preprint arXiv:2310.06825*, 2024.

Artem Kirsanov, Chi-Ning Chou, Kyunghyun Cho, and SueYeon Chung. The geometry of prompting: Unveiling distinct mechanisms of task adaptation in language models. *arXiv preprint arXiv:2502.08009*, 2025.

Aristotelis Lazaridis, Aman Sharma, Dylan Bates, Brian King, Vincent Lu, and Jack FitzGerald. Can editing 1 neuron fix repetition loops in llms? *arXiv preprint arXiv:2606.13705*, 2026.

Siquan Li, Kaiqi Jiang, Jiacheng Sun, and Tianyang Hu. The structural origin of attention sink: Variance discrepancy, super neurons, and dimension disparity. *arXiv preprint arXiv:2605.06611*, 2026.

Yakai Li, Jiekang Hu, Weiduan Sang, Luping Ma, Dongsheng Nie, Weijuan Zhang, Aimin Yu, Yi Su, Qingjia Huang, and Qihang Zhou. Prefill-level jailbreak: A black-box risk analysis of large language models. *arXiv preprint arXiv:2504.21038*, 2025.

Kristina McElheran, J Frank Li, Erik Brynjolfsson, Zachary Kroff, Emin Dinlersoz, Lucia Foster, and Nikolas Zolas. Ai adoption in america: Who, what, and where. *Journal of Economics & Management Strategy*, 33(2):375–415, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Jack Merullo, Srihita Vatsavaya, Lucius Bushnaq, and Owen Lewis. From memorization to reasoning in the spectrum of loss curvature. *arXiv preprint arXiv:2510.24256*, 2025.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. *arXiv preprint arXiv:2406.01506*, 2024.

Xiao Ding Philippe Remy. Financial news dataset from bloomberg and reuters. `https://github.com/philipperemy/financial-news-dataset`, 2015.

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

Valeria Ruscio, Umberto Nanni, and Fabrizio Silvestri. What are you sinking? a geometric approach on attention sink. *arXiv preprint arXiv:2508.02546*, 2025.

Raphaël Sarfati, Toni JB Liu, Nicolas Boullé, and Christopher J Earls. Lines of thought in large language models. *arXiv preprint arXiv:2410.01545*, 2024.

Zara Siddique, Irtaza Khalid, Liam Turner, and Luis Espinosa Anke. Shifting perspectives: Steering vectors for robust bias mitigation in llms. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 809–820, 2026.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Zunhai Su, Qingyuan Li, Hao Zhang, Weihao Ye, Qibo Xue, YuLei Qian, Yuchen Xie, Ngai Wong, and Kehong Yuan. Unveiling super experts in mixture-of-experts large language models. *arXiv preprint arXiv:2507.23279*, 2025.

Annalisa Szymanski, Noah Ziems, Heather A Eicher-Miller, Toby Jia-Jun Li, Meng Jiang, and Ronald A Metoyer. Limitations of the llm-as-a-judge approach for evaluating llm outputs in expert knowledge tasks. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pp. 952–966, 2025.

Daniel Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. Analysing the generalisation and reliability of steering vectors. *Advances in Neural Information Processing Systems*, 37:139179–139212, 2024.

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

Yu Yan, Sheng Sun, Zenghao Duan, Teli Liu, Min Liu, Zhiyi Yin, Jingyu Lei, and Qi Li. from benign import toxic: Jailbreaking the language model via adversarial metaphors. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4785–4817, 2025.

Junchi Yao, Shu Yang, Jianhua Xu, Lijie Hu, Mengdi Li, and Di Wang. Understanding the repeat curse in large language models from a feature perspective. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 7787–7815, 2025.

Itay Yona, Ilia Shumailov, Jamie Hayes, Federico Barbero, and Yossi Gandelsman. Interpreting the repeated token phenomenon in large language models. *arXiv preprint arXiv:2503.08908*, 2025.

Stephen Zhang, Mustafa Khan, and Vardan Papyan. Attention sinks and outlier features: A'catch, tag, and release'mechanism for embeddings. *arXiv preprint arXiv:2502.00919*, 2025.

A APPENDIX

A.1 AUTOEVAL PROCEDURE

When evaluating the outputs of the jailbreaks, we use GPT-5.6-Terra to evaluate either for coherence and irrelevance or repetitiveness and relevance simultaneously.

We treat coherence, relevance, and repetitiveness as binary attributes of a response given a prompt. For the first two attributes we use GPT5-Mini, and for the last we use Llama-3.1-70B-Instruct as the task is quite straightforward.

Evaluating for each of these attributes is accomplished using a custom in-context-learning prompt for each of the tasks that prompts the model to provide a binary score after the word "score". If a binary score is not matched this way, we give a score of "N/A" and this example is omitted from our analysis. Prompts are included in attached code.

A.2 DATASETS AND PROMPT SELECTION

We select four different datasets across four domains: the TweetEval dataset (Barbieri et al., 2018; 2020), the Financial News Dataset from Bloomberg and Reuters (Ding et al., 2014; Benayoun, 2024; Philippe Remy, 2015), the WikiText dataset (Merity et al., 2016), and a dataset of paragraphs from the Lord of the Rings fantasy series[1].

For our jailbreaking scheme for inducing relevant, repetitive generations 3.2, we randomly sample one prompt meeting some minimum length from each dataset.

For our causal interventions (Section 4), we randomly sample 50 prompts meeting some minimum length from each dataset.

A.3 LIST OF PERTURBATIONS

We sample two random prompts from each of four different datasets (see Appendix A.2) to create a set of 8 prompts. For each prompt, we generate 10 responses from each model using a variety of perturbations.

We perform a wide search across an array of perturbations to understand how the absence of different magnitude components influence generation. We test 245 different spectral perturbations (see Appendix A.3), varying both the band of the spectrum being targeted, as well as the layer where the spectrum is calculated and the perturbation takes place.

We produce generations for all 8 prompts across both models for the following perturbation regimes using the per-prompt spectrum:

- $\alpha \in \{0.3, 0.5, 0.65, 0.75\}, \beta = 1.0, \ell \in \{1, 3, 5, 7, 9, 11, 13, 15, 17\}$ (first vector not removed),
- $\alpha \in \{0.9, 0.925, 0.95, 0.975, 0.99\}, \beta = 1.0, \ell \in \{1, 3, 5, 7, 9, 11, 13, 15, 17\}$ (first vector not removed),
- $\alpha \in \{0.3, 0.5, 0.65, 0.75\}, \beta = 1.0, \ell \in \{1, 3, 5, 7, 9, 11, 13\}$,
- $\alpha \in \{0.9, 0.925, 0.95, 0.975, 0.99\}, \beta = 1.0, \ell \in \{1, 3, 5, 7, 9, 11, 13\}$,
- $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}, \beta = 0.9, \ell \in \{1, 3, 5, 7, 9, 11, 13, 15, 17\}$.

Just for Llama 3.1, we consider the following perturbations with the per-prompt spectrum:

- $\alpha \in \{0.5, 0.65, 0.75\}, \beta = 1.0, \ell \in \{1, 3, 5, 7, 9, 11, 13\}$ (first vectors not removed),
- $\alpha \in \{0.925, 0.95, 0.975, 0.99\}, \beta = 1.0, \ell \in \{1, 3, 5, 7, 9, 11, 13\}$ (first two vectors not removed).

[1] https://huggingface.co/datasets/AL49/lotr_paragraphs

A.4 Attention Patterns for Relevant Jailbreak

Here we include the attention patterns for the output with respect to the prompt for the relevant jailbreak (Figure 6). We see that differences in attention sink value are primarily relegated to the first few layers. Furthermore, the attention entropy does not differ much from the non-jailbreak text.

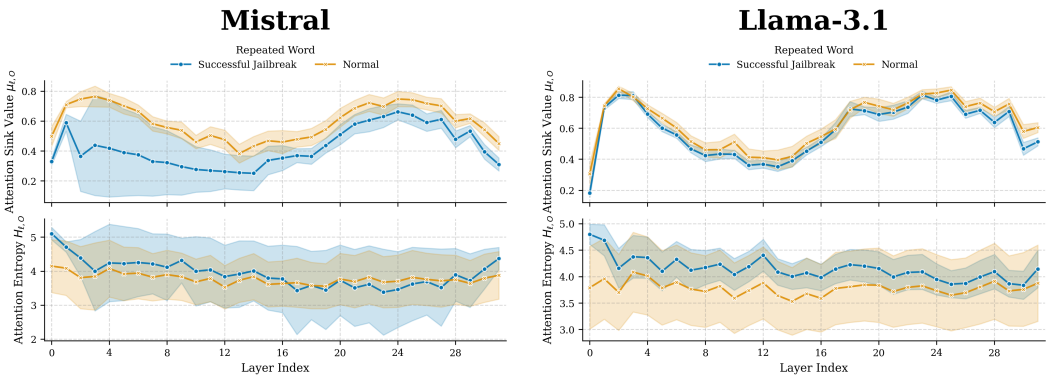


Figure 6: Attention patterns for the relevant jailbreak examples.

A.5 Prompt Attention Patterns for Irrelevant Jailbreak

Here we include the entropy over the prompt tokens from the prompt tokens themselves, as well as the mean attention sink values for the prompt (Figure 7). We see that the trends match those presented in Section 3.1.

A.6 Attention Patterns for Representation Perturbations

Here we include the mean attention sink values and attention entropies across layers for both the Big Upper + 1 and Small Upper regimes (Figure 8). Note that the uptick in entropy and drop in attention sink value for Big Upper + 1 is not noticeable in earlier layers because the perturbation is applied no earlier than layer 5. Furthermore, since the perturbation is applied either at layer 5, 7, or 9, averaging across these different perturbations will not show the sink value dropping to zero.

A.7 Compute Description

Generating all 200 generations associated with one combination of model and perturbation regime can be done on a single A100 GPU in less than 30 minutes. This includes the time to calculate and save different attention statistics.

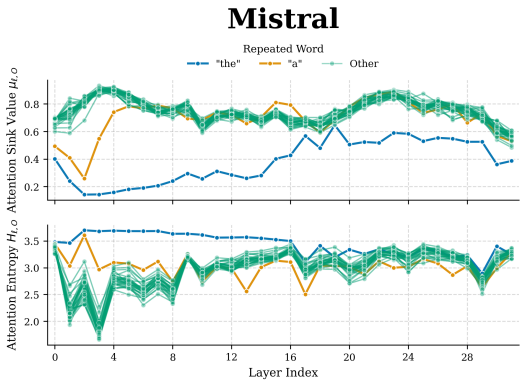


Figure 7: Attention patterns over the prompt for Mistral on the irrelevant jailbreak.

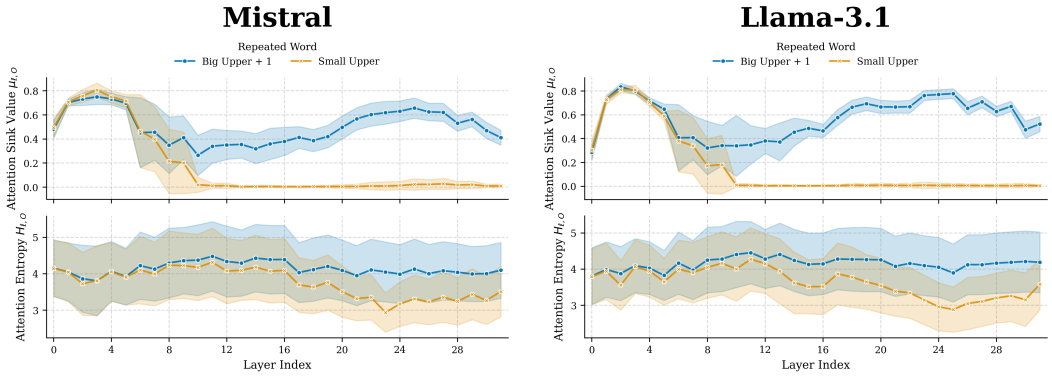


Figure 8: Attention patterns for Big Upper +1 and Small Upper.