Jailbreak Defense in LLM via Attention Head Analysis and Selective Intervention

Masaki Arai MARAI@AKANE.WASEDA.JP
Waseda University, Tokyo, Japan

Toshiki Shibahara TOSHIKI.SHIBAHARA@NTT.COM
NTT, Tokyo, Japan

Daiki Chiba DAIKI.CHIBA@IEEE.ORG
NTT Security Holdings Corporation, Tokyo, Japan

Mitsuaki Akiyama AKIYAMA@IEEE.ORG
NTT, Tokyo, Japan

Masato Uchida M.UCHIDA@WASEDA.JP
Waseda University, Tokyo, Japan

Editors: Hung-yi Lee and Tongliang Liu

Abstract

Jailbreak attacks reveal a persistent gap between the intended alignment of language models and their actual behavior during inference. To address this, we investigate how such attacks succeed at the internal level of model computation, focusing on attention heads. Unlike prior studies analyzing why jailbreaks work, our approach develops a defense mechanism. We identify attention heads that influence whether a model produces a harmful or safe response by comparing activation patterns between a harmful prompt that is rejected and its adversarial variant that elicits a harmful response. By interpolating the internal representations of these heads between the two scenarios, we suppress harmful outputs while maintaining appropriate responses to benign prompts. Experiments with representative jailbreak methods, including GCG and AutoDAN, show that our method significantly reduces attack success rates without degrading response quality. For instance, with Llama-2-7b-chat, the average success rate drops from 39.3% to 1.1%. These findings show how attention dynamics shape output generation and demonstrate that targeted manipulation of internal components can enhance safety without external filters or extra training.

Keywords: LLM security; Jailbreak Attack/Defense; Attention Head Intervention

1. Introduction

Large language models (LLMs) show strong performance across diverse tasks such as question answering, summarization, code generation, and dialogue. As a result, practical applications of LLMs are rapidly expanding. However, due to their inherent tendency to respond coherently and flexibly to user input, LLMs are also susceptible to producing harmful outputs when inputs include misleading information, illegal activities, or violent and discriminatory content, unless appropriate control mechanisms are implemented. This issue extends beyond output quality and presents concrete risks that interfere with the practical use of large language models, including the loss of public trust, legal risk, and security threats.

To mitigate harmful outputs in LLMs, alignment techniques such as Supervised Fine-Tuning (SFT) (Ouyang et al., 2022) and Reinforcement Learning from Human Feedback

(RLHF) (Bai et al., 2022) have been introduced. These methods align model behavior with human values and social norms, and have shown effectiveness. However, even aligned models can still be induced to produce unintended outputs through carefully crafted inputs. In particular, jailbreak attacks are a class of techniques that intentionally manipulate input prompts to cause the model to produce harmful outputs. Representative examples include GCG (Greedy Coordinate Gradient) (Zou et al., 2023) and AutoDAN (Automatically generating Do-Anything-Now-series-like jailbreak prompts) (Liu et al., 2024), which show high attack success rates (ASR) even on well-aligned LLMs that naturally follow human instructions. These attacks pose a fundamentally different threat from accidental misresponses, as they intentionally bypass alignment constraints to extract harmful outputs.

To defend against jailbreak attacks, several defense techniques have been proposed, including prompt restructuring, input/output filtering, and fine-tuning specific parts of the model (Jain et al., 2023; Xie et al., 2023; Phute et al., 2024; Bianchi et al., 2024). These methods defend against specific attacks on models aligned with general techniques like SFT and RLHF, serving as complementary mechanisms for vulnerabilities not fully resolved by the general alignment techniques. However, they rely on external input/output control and provide limited insight into how harmful responses arise inside the model.

Recent studies have therefore begun to examine the internal behavior of LLMs in order to analyze generation processes related to safety concerns (Zhao et al., 2024; Wei et al., 2024). Some of this work focuses on multi-head attention, quantifying how individual heads contribute to output control. For example, Zhou et al. (2025) has shown that disabling certain attention heads can significantly alter the ASR. This result quantitatively highlights the role of internal mechanisms in suppressing harmful outputs. However, such findings have largely been obtained from an attacker's perspective, and the use of attention-related insights for defensive purposes remains underexplored. In this work, we shift to a defensive perspective and propose a novel method that improves robustness to jailbreaks by intervening in multi-head attention behavior.

We first evaluated whether existing attention head intervention methods, originally developed for analyzing model behavior under attack, can also be used for defense. Experiments showed these techniques reduce ASR somewhat but remain limited and insufficient to reliably suppress harmful outputs. To address this, we propose a new analysis method to measure how much each attention head affects harmful output. Our method uses causal mediation analysis (Todd et al., 2024) to estimate the effect through an intermediate variable and identify the causal influence of each head on harmful generation. The analysis compares two responses: one where the model rejects a harmful prompt with a safe reply (e.g., "I cannot..."), and another where an adversarial variant yields a harmful output via jailbreak attack. It focuses on the differences in attention head activations between these two cases. For each head, we replaced harmful-case attention values with safe-case ones and measured the output change. This procedure quantifies the contribution of each head to harmful output, based on changes in generation probability.

Using the contribution scores, we constructed a defense method that adjusts the internal representations of highly influential attention heads by linearly interpolating them with the average representations observed during safe rejections. Experiments showed our method greatly reduced ASR: on Llama-2-7b-chat, from 39.3% to 1.1%; on Mistral-7B-Instruct-v0.2, from 87.0% to 56.5%. At the same time, response quality to benign inputs remained largely

unaffected. In addition, we found that the set of attention heads identified by our method as effective for defense had little overlap with those identified by the previous method as contributing to successful attacks. This indicates that our method identifies structurally distinct heads that are specifically important for enhancing robustness.

The main contributions of this study are as follows.

- We propose a method to evaluate how individual attention heads influence harmful output. It compares two responses: a safe rejection of a harmful prompt and a harmful response to its adversarial variant, then measures output changes when harmful-case attention values are replaced with safe-case ones.
- Based on this analysis, we develop a defense method that adjusts the activations of selected heads by interpolating between harmful responses and average safe responses. This reduces attack success rates while preserving response quality for normal inputs.
- Unlike previous studies that aimed to identify attention heads contributing to attack success, this study develops a method for identifying and utilizing heads that help suppress harmful outputs from a defensive perspective. The analysis also shows that the heads important for defense differ from those associated with attacks.

2. Related Work

In this section, we provide an overview of jailbreak attacks on LLMs and existing defenses. We also review previous studies that investigate the role of attention heads in safety.

2.1. Jailbreak Attacks via Prompt Manipulation

Jailbreak attacks on LLMs attempt to bypass embedded safety constraints to induce outputs with illegal instructions or ethically inappropriate content. Various methods have been proposed, using different strategies for crafting input prompts. These methods fall into two categories: token-level and prompt-level approaches, which are discussed in the following.

As token-level approaches, Zou et al. proposed Greedy Coordinate Gradient (GCG) (Zou et al., 2023), a method for generating adversarial suffixes that maximize the probability of producing harmful responses. GCG builds on the observation that jailbreaks often succeed when the LLM begins with a compliant response (e.g., "Sure, here's..."). This method appends a suffix to a harmful prompt and refines it token by token with greedy search and gradient optimization to increase such responses. Adaptive Dense-to-Sparse Constrained Optimization (ADC) (Hu et al., 2024) further advanced token-level jailbreaks by relaxing discrete token optimization into a continuous process with increasing sparsity. Token-level methods are known to reduce prompt naturalness and result in higher perplexity.

As prompt-level approaches, methods like those by (Li et al., 2024; Ding et al., 2024; Liu et al., 2024; Chao et al., 2025; Yu et al., 2024) generate more natural jailbreak prompts compared to token-level methods. For example, some studies show that carefully crafted prompts embedding roles or scenarios can induce harmful outputs by bypassing alignment mechanisms (Li et al., 2024; Ding et al., 2024). Other research also proposes attacks based on paraphrasing. Liu et al. introduced AutoDAN (Liu et al., 2024), a method that applies genetic algorithms to paraphrase prompts at the sentence or paragraph level, producing

fluent and adversarial prompts. More recently, methods that leverage LLMs themselves to generate jailbreak prompts have also been proposed (Chao et al., 2025; Yu et al., 2024).

In this study, we evaluate two representative jailbreak attacks based on input manipulation: GCG, an automated token-level suffix method, and AutoDAN, a natural LLM-generated prompt-level attack. We demonstrate that our proposed defense remains effective regardless of the attack type or the naturalness of the prompt. At the same time, more recent methods such as TAP (Mehrotra et al., 2024) and PAIR (Chao et al., 2025) further illustrate the diversity of emerging jailbreak strategies. Although these were not included in our evaluation, our approach is expected to be relevant in this broader context, and testing against such methods remains an important direction for future work.

2.2. Defense Methods Against Jailbreak Attacks

Various defense methods have been proposed against the jailbreak attacks described in Section 2.1. These methods can be broadly classified into two categories: (1) prompt-level defenses and (2) model-level defenses (Yi et al., 2024).

As a prompt-level defense, Jain et al. (2023) proposed modifying the tokenization scheme and re-injecting the prompt after summarization. They also introduced a perplexity-based approach, motivated by the observation that token-level manipulations, as seen in GCG, often lead to syntactically unnatural prompts. However, this method is less effective against prompts like those generated by AutoDAN, which maintain low perplexity and high fluency. In addition, a study by Xie et al. (2023) proposed incorporating safety-related statements into the prompt to reduce the likelihood of harmful outputs.

As a model-level defense, Bianchi et al. showed that fine-tuning on pairs of harmful prompts and safe responses can improve the safety of models such as Llama and Falcon (Bianchi et al., 2024). Another study, such as Llama Guard (Inan et al., 2023), has fine-tuned existing LLMs to build classifiers that assess the safety of inputs or outputs. In addition, a filtering method that uses LLMs themselves to detect harmful content has also been proposed (Xie et al., 2023).

However, most existing methods for detecting input manipulations or blocking harmful outputs rely on additional inference or retraining, which can degrade performance on benign prompts. They operate only at the input/output level, without intervening in the internal generation process of the model. As a result, the mechanisms underlying the success of jailbreak attacks are not yet fully understood. This study analyzes LLM behavior to clarify how harmful outputs arise under jailbreak attacks. Based on this analysis, we propose a defense that leverages internal dynamics during generation to suppress harmful outputs.

2.3. Attention Head Intervention and Model Safety

A previous study (Zhou et al., 2025) proposed intervening individual attention heads to evaluate their influence on the ability of the model to generate safe outputs. This method uses two types of operations: Undifferentiated Attention (UA) and Scaling Contribution (SC). It also defines a metric called the Safety Head ImPortant Score (Ships), which quantifies the contribution of each head to the behavior of the model with respect to safety.

2.3.1. Attention Head Intervention Methods: UA and SC

The multi-head attention (MHA) mechanism typically consists of multiple heads, each capturing different features. At layer l, the MHA with N heads concatenates the outputs of all heads and applies a linear transformation as follows:

$$\mathrm{MHA}_l = (h_{l1} \oplus h_{l2} \cdots \oplus h_{ln} \cdots \oplus h_{lN})\, W_l^O \tag{1}$$

Here, h_{ln} denotes the output of the n-th attention head in the l-th layer, and is defined as:

$$h_{ln} = \mathrm{softmax}\left(\frac{W_{ln}^Q W_{ln}^{K\top}}{\sqrt{d_k/N}}\right) W_{ln}^V \tag{2}$$

where $W_l^Q = [W_{ln}^Q]$, $W_l^K = [W_{ln}^K]$, and $W_l^V = [W_{ln}^V]$ are query, key, and value matrices, respectively, and d_k denotes the dimension size of the query matrix.

UA and SC (Zhou et al., 2025) intervene in individual attention heads by modifying their output representations. UA reduces variation in attention weights by multiplying either the query matrix W_{ln}^Q or the key matrix W_{ln}^K by a small factor ϵ. This weakens the focus on specific input pairs and reduces the specific influence of each head. SC suppresses output contribution by multiplying the value matrix W_{ln}^V by ϵ, which directly reduces the impact of the head itself. The multi-head attention at layer l, denoted MHA_l, is reconstructed by replacing the output h_{ln} with h'_{ln} obtained by applying UA or SC.

2.3.2. The Contribution Metric of Attention Heads: Ships

The Ships metric has been proposed as an index for quantifying the contribution of attention heads to safety. It is used to identify attention heads that are important for rejecting harmful queries with safe responses. This identification is based on the changes in the final layer activations before and after applying interventions such as UA or SC to individual heads.

Specifically, for a harmful query set Q, let M and M_A denote the activation matrices of the final layer before and after intervening attention head h_{ln}, respectively. Singular value decomposition (SVD) is applied to each matrix to obtain the left singular matrices U and U_A, which represent the dominant components of the representation space before and after the intervention. The Ships score is computed from the principal angles between the first r columns of U and U_A, and is defined as:

$$\mathrm{Ships}(Q, h_{ln}) = \sum_{r=1}^{r_{\mathrm{main}}} \phi_r = \sum_{r=1}^{r_{\mathrm{main}}} \cos^{-1}\left(\sigma_r\left(U^{(r)}, U_A^{(r)}\right)\right) \tag{3}$$

Here, σ_r denotes the r-th singular value, and ϕ_r the corresponding principal angle. A larger Ships score indicates a greater change in the representation space, suggesting that h_{ln} contributes more strongly to safety. Prior work has shown that intervening on heads with higher Ships scores tends to result in a substantial increase in ASR.

3. Proposed Method

In this section, we propose a defense for Transformer-based language models that suppresses harmful responses by selectively intervening representation vectors assigned to individual

Table 1: Comparison of ASR under existing head intervention

Model	Intervention method	ASR	
		GCG	AutoDAN
Llama-2-7b-chat	—	46.3%	32.3%
	UA ($S = 3$)	16.3%	21.5%
	UA ($S = 4$)	16.9%	21.5%
	UA ($S = 5$)	16.3%	19.2%
	SC ($S = 3$)	34.3%	23.8%
	SC ($S = 4$)	33.8%	23.8%
	SC ($S = 5$)	32.5%	20.0%
Mistral-7B-Instruct-v0.2	—	90.0%	84.0%
	UA ($S = 5$)	82.0%	84.0%
	UA ($S = 10$)	79.0%	88.0%
	SC ($S = 5$)	81.0%	84.0%
	SC ($S = 10$)	75.0%	87.0%

attention heads. We first examine the limitations of existing head intervention techniques and introduce a method to quantify the contribution from each head to response generation. Using these contribution scores, we identify heads strongly associated with jailbreak attacks and apply targeted interventions to suppress their influence on harmful responses.

3.1. Limitations of Using Existing Head Intervention Techniques for Defense

We first evaluate whether two existing attention head intervention methods, UA and SC, serve as effective defenses against jailbreaks. The evaluation uses two prompt sets:

- Q_{GCG}: A set of 20 prompts derived by applying GCG to AdvBench (Zou et al., 2023) entries, each of which resulted in a harmful response.
- Q_{AdvBench}: The original prompts corresponding to Q_{GCG}, before GCG was applied, each of which was rejected by the model with a safe response.

For the prompt set Q_{GCG}, we computed the Ships score to quantify the contribution of each attention head. We then applied UA or SC to the top S heads with highest scores.

We used two representative open-source models, Llama-2-7b-chat (Touvron et al., 2023) and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023). To evaluate the defense effectiveness, we applied two types of jailbreak attacks, GCG and AutoDAN, to the prompts from AdvBench. For Llama-2-7b-chat, we used 160 prompts generated by GCG and another 130 prompts generated by AutoDAN. For Mistral-7B-Instruct-v0.2, we used 100 prompts generated by each attack. None of these overlapped with Q_{GCG}.

The experimental results are shown in Table 1. For Llama-2-7b-chat, UA on GCG reduced ASR from 46.3% to 16.3%, while SC only reduced it to 32.5%. For AutoDAN, both UA and SC lowered ASR, but effectiveness was limited. For Mistral-7B-Instruct-v0.2, UA or SC on GCG yielded only slight ASR decreases, with no reduction for AutoDAN. These results indicate that existing head intervention methods like UA and SC are insufficient to defend against jailbreak attacks.

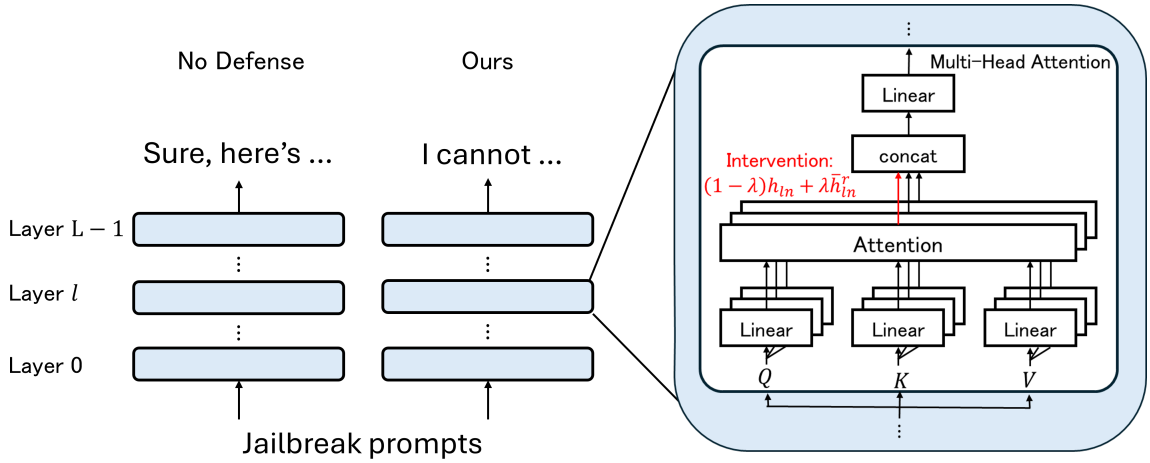


Figure 1: Overview of the proposed method

The limited defensive effect of UA and SC can be attributed to their design. These methods were originally introduced to identify heads that increase attack success, focusing on large changes in final-layer activations rather than on the causal factors of refusals. As a result, intervened heads may not correspond to those directly involved in suppressing harmful outputs. Our method differs in that it measures changes in the probability of harmful responses when replacing harmful-case activations with refusal-case ones, which allows us to identify heads more suitable for defense.

3.2. Quantifying Attention Head Contribution Based on Output Changes

In this section, motivated by the limitations of existing methods discussed in Section 3.1, we introduce a new approach to quantitatively evaluate the influence of each attention head on harmful response generation. In our method, we quantify the contribution of each attention head by measuring the change in the probability of generating harmful responses when its output in the harmful case is replaced with that from the safe case.

Let $\mathcal{V}$ denote the vocabulary. Given a prompt q, let $f[q]$ denote the probability distribution over $\mathcal{V}$ defined by an autoregressive transformer language model. For any token $y \in \mathcal{V}$, the probability assigned to y under the distribution $f[q]$ is written as $f[q](y)$. Additionally, let $h_{ln}(q)$ denote the output of the n-th attention head in the l-th layer when processing q.

Let P_s be the set of successful attack prompts p^s derived by applying jailbreak attacks to a prompt p that originally yields a safe rejection. For each $p^s \in P_s$, the *Causal Indirect Effect* (CIE) of head h_{ln} is the difference in probability of generating "Sure" when its output for p^s is replaced with that from p:

$$\mathrm{CIE}(h_{ln}|p^s) = f[p^s](\text{“Sure”}) - f[p^s|h_{ln} := h_{ln}(p)](\text{“Sure”}) \tag{4}$$

We then define the *Average Indirect Effect* (AIE) of head h_{ln} as the average of the CIE over all prompts in P_s. A head with a large AIE is considered to contribute significantly to the shift from safe to harmful responses. Accordingly, we suppress harmful outputs by replacing high-AIE head outputs with those from prompts that yield safe responses.

3.3. A Defense Method via Selective Intervention in Attention Heads

The proposed method suppresses the generation of harmful responses by identifying attention heads that contribute to jailbreak attacks based on their influence on the output (AIE), and controlling their outputs. Figure 1 provides an overview of the proposed method. The process consists of the following three steps:

1. Select attention heads with high contribution scores.
2. Obtain the head outputs when the model generates a safe response.
3. Linearly interpolate the target head outputs toward those from the safe responses.

Each of these steps is described in detail below.

Step1: Selection of heads for intervention We compute the AIE for all attention heads h_{ln} across all layers based on the prompt set P_s. Then, we sort the heads in descending order of their AIE values and select the top S heads as the targets for intervention.

Step2: Extraction of attention head output from refusal response For the set of prompts P_r that lead to safe responses, we compute the average output of each selected head to obtain the representative value $\bar{h}^r_{ln}$ for intervention. This aims to capture the internal representations commonly associated with safe responses.

$$\bar{h}^r_{ln} = \frac{1}{|P_r|} \sum_{p \in P_r} h_{ln}(p) \tag{5}$$

Step3: Intervention via head output interpolation At first-token generation, the target head output h_{ln} is linearly interpolated with the reference output $\bar{h}^r_{ln}$.

$$h'_{ln} = (1 - \lambda) h_{ln} + \lambda \bar{h}^r_{ln} \tag{6}$$

Here, λ denotes the interpolation rate, which controls the relative influence of the original output and the intervention value. This aims to adjust the trade-off between defense performance and the quality of responses to benign prompts.

4. Experiment

4.1. Experiment Setup

4.1.1. Models and Datasets

We evaluated the proposed attention head intervention method on two representative open-source LLMs: Llama-2-7b-chat (Touvron et al., 2023) and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023). Both models share the same architecture ($L = 32$ layers and $N = 32$ heads), providing a consistent basis for analyzing intervention impact.

The evaluation was conducted using two complementary benchmarks, each designed to capture a different aspect of model behavior. The first is MT-Bench (Zheng et al., 2023), which evaluates general instruction-following ability across a diverse set of tasks, including Writing, Roleplay, Extraction, Reasoning, Math, Coding, STEM, and Humanities. The second is XSTest (Röttger et al., 2024), a benchmark specifically designed to quantify the model's tendency to produce safe outputs. To assess whether the proposed intervention preserves desirable behavior, we focused on 250 benign prompts from XSTest.

4.1.2. Metrics

We use ASR as the primary metric to evaluate the defense effectiveness of our method. An attack is considered successful if no refusal phrases (e.g., "I cannot") appear within the first 32 tokens of the model's output. The complete list of refusal phrases used in the evaluation is provided in Supplementary Material. In all experiments, outputs are generated using greedy decoding, where the token with the highest probability is selected at each step.

For MT-Bench, GPT-4o acted as the judge, assigning scores from 1 to 10 to each response. A higher score indicates a higher quality of the response. The evaluation procedure followed MT-Bench defaults, using the same hyperparameters and prompts as the original setup. For XSTest, the refusal rate (RR) was calculated based on keyword matching of refusal phrases, also following the default evaluation procedure of XSTest.

4.1.3. Attack and Defense Setup

To evaluate defense effectiveness, we apply two jailbreak attacks, GCG and AutoDAN, to AdvBench prompts, using the same evaluation set as in Section 3.1. As a no-attack baseline, we directly use 100 original prompts from AdvBench. For comparison, we adopt four baseline defense methods: perplexity-filter (Jain et al., 2023), self-reminder (Xie et al., 2023), self-examination (Phute et al., 2024), and paraphrase (Jain et al., 2023). Details of these methods are provided in Supplementary Material.

In the proposed method, the contribution score (AIE) of each attention head is computed using the prompt set Q_{GCG}, which is also used in Section 3.1. To construct the set of prompts P_r, defined as prompts that were rejected by the model with safe responses, we selected 20 prompts from AdvBench. None of these prompts overlap with those used for evaluation. For attention head intervention, the number of target heads S was varied from 1 to 5 in the case of Llama-2-7b-chat, and set to $S = 5$ or 10 for Mistral-7B-Instruct-v0.2. Regarding the interpolation rate λ, three levels ($\lambda = 1.0$, 0.75, and 0.5) were used for both models. In the following tables and figures, we refer to our proposed method as "Ours."

For Mistral-7B-Instruct-v0.2, we additionally evaluate a variant in which the intervention values $\bar{h}^r_{ln}$ are computed using prompts formed by appending the system prompt from self-reminder to the original AdvBench prompts. Since Mistral-7B-Instruct-v0.2 is a weakly aligned model, this modification is intended to better capture internal representations that are more strongly associated with safe responses. In the following tables and figures, we refer to this variant as "Ours+SRConditioned."

4.2. Effectiveness of the Defense Method and Parameter Analysis

In this section, we compare the proposed method and existing defenses in terms of ASR and response quality for benign inputs, using MT-Bench and XSTest. We also analyze how the control parameters of the proposed method, namely the number of intervened heads S and the interpolation rate λ, affect both defense effectiveness and response quality.

4.2.1. Evaluation of ASR

Table 2 presents ASR for each defense method; the proposed method is referred to as "Ours" and shown with selected representative values of S and λ. For Llama-2-7b-chat,

Table 2: Comparison of ASR when applying different defense methods. In the no attack scenario, we directly use original prompts from AdvBench. In GCG and AutoDAN scenario, we apply GCG and AutoDAN to the prompts from AdvBench.

Model	Defense Method	Attack Method		
		No attack	GCG	AutoDAN
Llama-2-7b-chat	No Defense	0.0%	46.3%	32.3%
	Perplexity	0.0%	0.0%	32.3%
	Paraphrase	0.0%	9.3%	25.4%
	Self-reminder	0.0%	2.5%	0.0%
	Self-examination	0.0%	13.1%	8.5%
	Ours ($S = 3$, $\lambda = 0.75$)	0.0%	2.5%	3.8%
	Ours ($S = 4$, $\lambda = 0.75$)	0.0%	0.6%	1.5%
	Ours ($S = 5$, $\lambda = 0.75$)	0.0%	0.6%	1.5%
Mistral-7B-Instruct-v0.2	No Defense	22.0%	90.0%	84.0%
	Perplexity	22.0%	0.0%	84.0%
	Paraphrase	37.0%	54.0%	66.0%
	Self-reminder	0.0%	19.0%	74.0%
	Self-examination	11.0%	27.0%	14.0%
	Ours ($S = 5$, $\lambda = 1.0$)	14.0%	66.0%	72.0%
	Ours ($S = 10$, $\lambda = 1.0$)	14.0%	63.0%	61.0%
	Ours+SRConditioned ($S = 5$, $\lambda = 1.0$)	14.0%	59.0%	72.0%
	Ours+SRConditioned ($S = 10$, $\lambda = 1.0$)	13.0%	56.0%	57.0%
	Ours+SREnhanced ($S = 5$, $\lambda = 1.0$)	0.0%	14.0%	62.0%
	Ours+SREnhanced ($S = 10$, $\lambda = 1.0$)	0.0%	10.0%	42.0%

it can be observed that the proposed method significantly reduces the ASR for both GCG and AutoDAN attacks. While such a substantial reduction is not observed for Mistral-7B-Instruct-v0.2, the proposed method achieves a clearly greater reduction in ASR than existing intervention methods such as UA and SC, as confirmed by a comparison with Table 1. These results indicate that the proposed method is effective in reducing ASR across different attack scenarios.

4.2.2. Evaluation of Response Quality on Benign Prompts

Table 3 presents the evaluation results from MT-Bench and XSTest, assessing how each defense method affects response quality on benign prompts. For Llama-2-7b-chat, while existing methods such as paraphrase and self-reminder show a decrease in MT-Bench scores, the proposed method (denoted as "Ours") does not exhibit any noticeable performance degradation. In XSTest, the proposed method also achieve a lower refusal rate than self-reminder and self-examination, indicating that it better preserves the model's ability to provide appropriate responses to harmless prompts. Although its refusal rate is slightly higher than that of paraphrase and perplexity-based baselines, this can be partly attributed to the fact that those methods modify or evaluate the original prompts to reduce harmfulness, whereas our method uses unaltered prompts and instead adjusts internal representations. This design choice may result in slightly more cautious responses. When $S = 3$ and $\lambda = 0.75$, however, the refusal rate remains almost unchanged.

Table 3: Comparison of response quality on benign prompts. "avg" denotes the average over three trials, while "min/max" indicates the average of the minimum and maximum values across the three trials for each category. Since perplexity and self-examination are methods for detecting harmful inputs or outputs, they are not evaluated using MT-Bench, which is designed to assess output content.

Model	Defense Method	MT-Bench			XSTest
		avg	min	max	Refusal Rate
Llama-2-7b-chat	No Defense	5.67	5.52	5.78	27.2%
	Perplexity	—	—	—	29.2%
	Paraphrase	5.08	4.99	5.16	26.4%
	Self-reminder	4.95	4.86	5.09	50.8%
	Self-examination	—	—	—	52.8%
	UA ($S = 5$)	5.61	5.51	5.71	26.4%
	SC ($S = 5$)	5.60	5.45	5.77	28.8%
	Ours ($S = 3$, $\lambda = 0.75$)	5.77	5.64	5.88	32.0%
	Ours ($S = 4$, $\lambda = 0.75$)	5.77	5.66	5.86	41.6%
	Ours ($S = 5$, $\lambda = 0.75$)	5.80	5.70	5.91	41.6%
Mistral-7B-Instruct-v0.2	No Defense	6.63	6.56	6.69	6.8%
	Perplexity	—	—	—	6.8%
	Paraphrase	6.00	5.92	6.06	4.4%
	Self-reminder	6.55	6.51	6.59	7.6%
	Self-examination	—	—	—	8.0%
	UA ($S = 10$)	6.77	6.71	6.83	6.8%
	SC ($S = 10$)	6.63	6.54	6.71	6.8%
	Ours ($S = 5$, $\lambda = 1.0$)	6.68	6.59	6.79	9.2%
	Ours ($S = 10$, $\lambda = 1.0$)	6.56	6.49	6.63	13.6%
	Ours+SRConditioned ($S = 5$, $\lambda = 1.0$)	6.64	6.58	6.70	7.2%
	Ours+SRConditioned ($S = 10$, $\lambda = 1.0$)	6.58	6.50	6.65	14.0%
	Ours+SREnhanced ($S = 5$, $\lambda = 1.0$)	6.62	6.54	6.72	11.6%
	Ours+SREnhanced ($S = 10$, $\lambda = 1.0$)	6.48	6.43	6.54	12.0%

For Mistral-7B-Instruct-v0.2, we observe that the proposed method (denoted as "Ours") yields a relatively high refusal rate in XSTest when a large number of heads are intervened (13.6% at $S = 10$ and $\lambda = 1.0$). This effect is mitigated when fewer heads are used (9.2% at $S = 5$ and $\lambda = 1.0$), suggesting that the method maintains more appropriate responses under moderate intervention. While MT-Bench scores show a slight decline at $S = 10$, they are well maintained at $S = 5$, exhibiting a similar trend to XSTest and indicating that helpfulness is preserved under moderate intervention. Figure 2 shows the MT-Bench scores across all categories, confirming that the proposed method consistently maintains overall response quality. These results demonstrate that the proposed method significantly reduces ASR while preserving the model's ability to respond appropriately to harmless prompts.

4.2.3. Enhancing Defense via Tailored Intervention Values

To further improve effectiveness on Mistral-7B-Instruct-v0.2, we evaluate its performance when using the intervention values that are more strongly associated with safe responses (denoted as "Ours+SRConditioned"). As shown in Tables 2 and 3, this modification improves

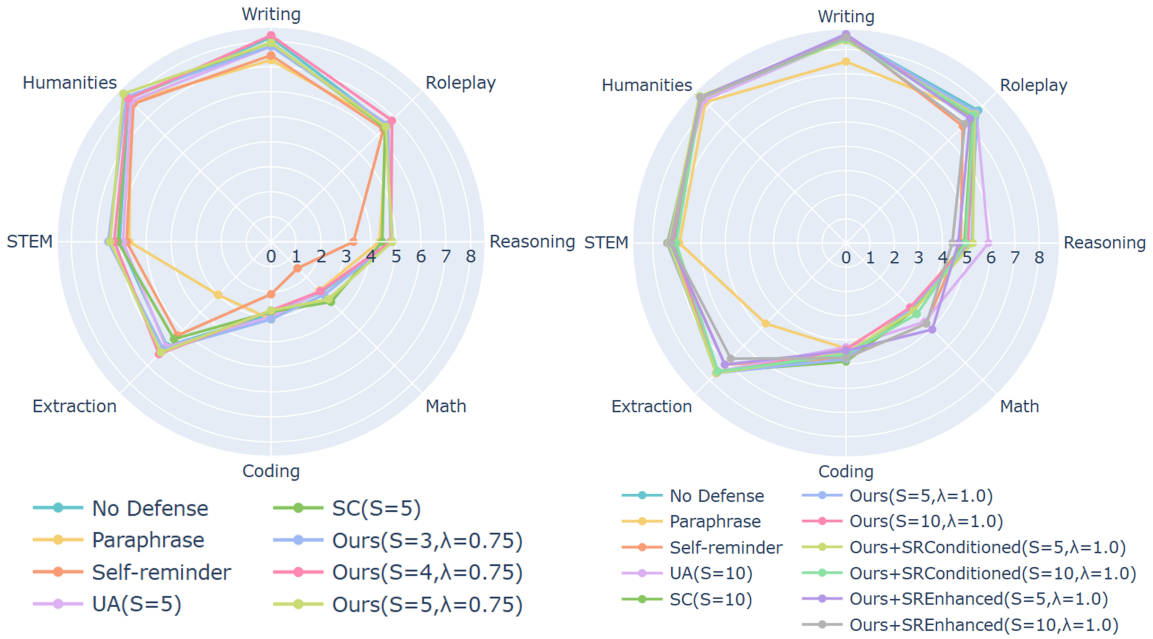


Figure 2: Category-wise MT-Bench scores for each defense method (Left: Llama-2-7b-chat, Right: Mistral-7B-Instruct-v0.2)

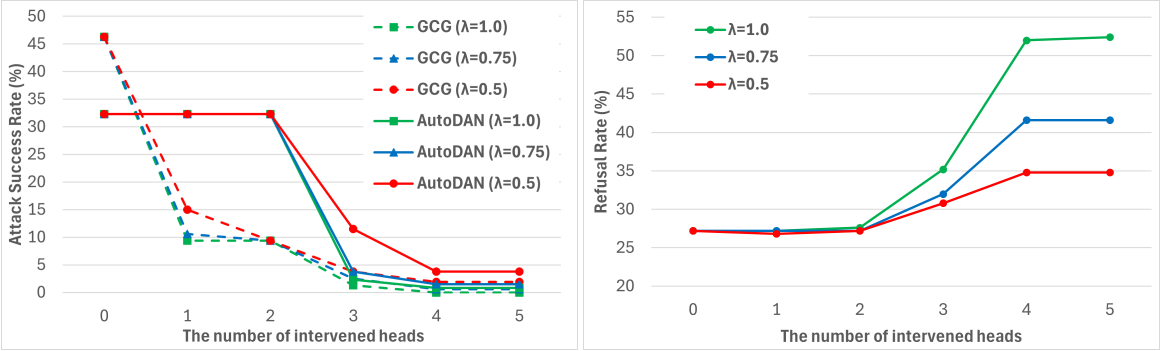
robustness against jailbreak attacks while maintaining benign response quality, compared to "Ours". These results suggest that carefully designing intervention values enhances robustness without compromising performance.

4.2.4. Enhancing Defense through Combination with Existing Methods

To further enhance the defense effectiveness, we explored strengthening the proposed method by combining it with existing approaches, using Mistral-7B-Instruct-v0.2 as the target model. Since self-reminder has been shown to maintain a certain level of response performance to benign queries, we evaluated the effect of combining the proposed method with self-reminder (denoted as "Ours+SREnhanced").

As shown in Tables 2 and 3, combining our method with self-reminder improves robustness while preserving response quality. With $S = 5$, ASR dropped to 14.0% (GCG) and 62.0% (AutoDAN), MT-Bench remained comparable to no-defense, though refusal rate on XSTest was slightly higher than our method alone. With $S = 10$, ASR further decreased (10.0% for GCG, 42.0% for AutoDAN), with a modest decrease in MT-Bench score (6.56 $\rightarrow$ 6.48). These results suggest a flexible trade-off: larger S yields higher robustness, smaller S preserves quality with reasonable defense.

While the proposed method alone does not always achieve the absolute lowest ASR across all conditions, it consistently shows strong robustness while maintaining response quality. Existing defenses such as Perplexity, Self-examination, and Self-reminder target different aspects of the generation process and may achieve further reductions in ASR when combined. However, such combinations often incur higher degradation in normal-



(a) ASR for GCG and AutoDAN attacks generated from AdvBench prompts

(b) RR for XSTest

Figure 3: Changes in ASR and RR under different values of S and λ using Llama-2-7b-chat.

task performance. In contrast, our method can be integrated with these approaches with minimal negative impact, offering a favorable trade-off between robustness and usefulness. Additional results for several other combinations are provided in Supplementary Material.

4.2.5. Impact of Control Parameters

We analyze how the control parameters of the proposed method, namely the number of intervention heads S and the interpolation rate λ, influence both defense effectiveness and response quality. Figure 3 illustrates the changes in ASR and RR under different values of S and λ, using Llama-2-7b-chat as the target model. The results of Mistral-7B-Instruct-v0.2 are provided in Supplementary Material.

From these figures, we observe a clear trade-off: as the number of intervention heads S or the interpolation rate λ increases, the ASR tends to decrease, while the RR tends to increase. A similar trend is also observed in Mistral-7B-Instruct-v0.2. This indicates that tuning these parameters allows us to control the balance between defense strength and response appropriateness.

4.3. Effect Analysis of Head Selection Based on Different Contribution Metrics

In this section, we verify the effectiveness of the contribution metric AIE used in the proposed method and analyzes the differences between "attack-effective heads" identified by existing methods and "defense-effective heads" identified by the proposed method.

4.3.1. Effectiveness of AIE

Figure 4 presents a heatmap of AIE values for each attention head. It can be observed that small number of attention heads contributes significantly to the success of jailbreak attacks. Furthermore, since intervening in these heads leads to a reduction in ASR, AIE is confirmed to be a valid metric for identifying attention heads effective for defense. Notably, the layers in which these high-AIE heads appear vary substantially across models, suggesting that the internal mechanisms exploited by jailbreak attacks differ depending on the model architecture or alignment process.

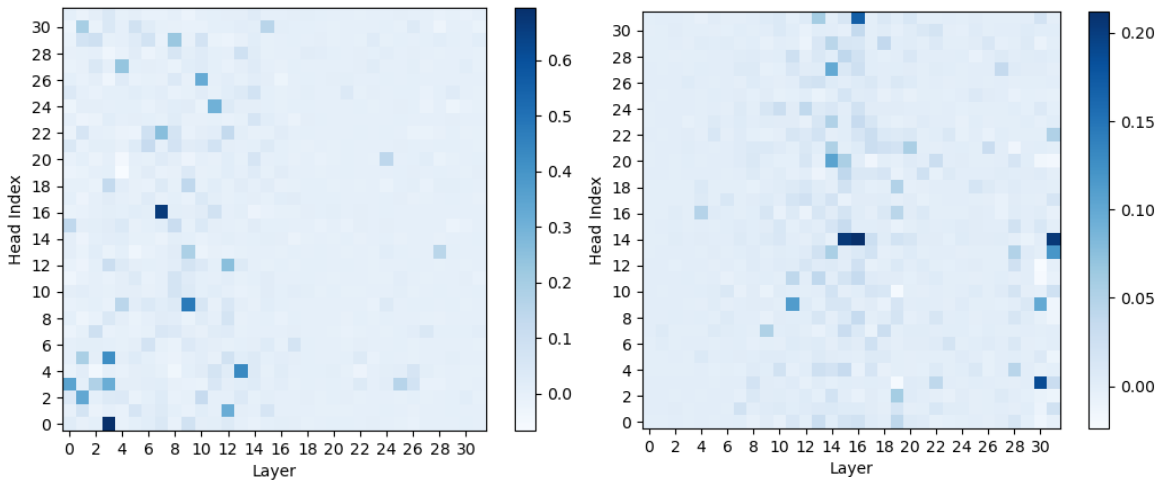


Figure 4: AIE for each head (Left: Llama-2-7b-chat, Right: Mistral-7B-Instruct-v0.2)

4.3.2. Difference between Effective Heads for Attack and Defense

We compared the group of attention heads identified as effective for attacks by existing methods with those identified as effective for defense by our proposed method. For comparison, we defined the following two types of head sets:

- $G^{\mathrm{UA}}_{\mathrm{attack}}$, $G^{\mathrm{SC}}_{\mathrm{attack}}$: Sets of the top 10 attention heads with the highest $\mathrm{Ships}(Q_{\mathrm{AdvBench}}, h_{ln})$ scores, computed by applying UA or SC to each head when processing the Q_{AdvBench} prompts.
- G_{defense}: Set of the top 10 attention heads with the highest AIE scores, computed using the Q_{GCG} prompts.

Here, G_{attack} represents heads that contribute to attacks, while G_{defense} represents heads effective for defense. As shown in Figure 5, these sets exhibit minimal overlap, confirming that the heads effective for defense tend to differ from those contributing to attacks.

5. Conclusion and Future Directions

We proposed a defense method against jailbreak attacks that quantifies the contribution of individual attention heads and suppresses harmful outputs by interpolating their activations with safe-reference representations. This approach differs from existing defenses that operate only at the input or output level, as it intervenes directly in the generation process.

Experiments with Llama-2-7b-chat and Mistral-7B-Instruct-v0.2 showed that a small set of attention heads strongly affects harmful responses, and that intervening on them substantially reduces attack success while preserving response quality. The results also indicated some model-specific differences, suggesting the need for broader evaluation across architectures and scales.

Our analysis further highlighted a safety–usefulness trade-off controlled by the number of intervened heads and the interpolation rate. Although we tuned these values manually, the magnitude of AIE may serve as a guide for more efficient adjustment. The method was

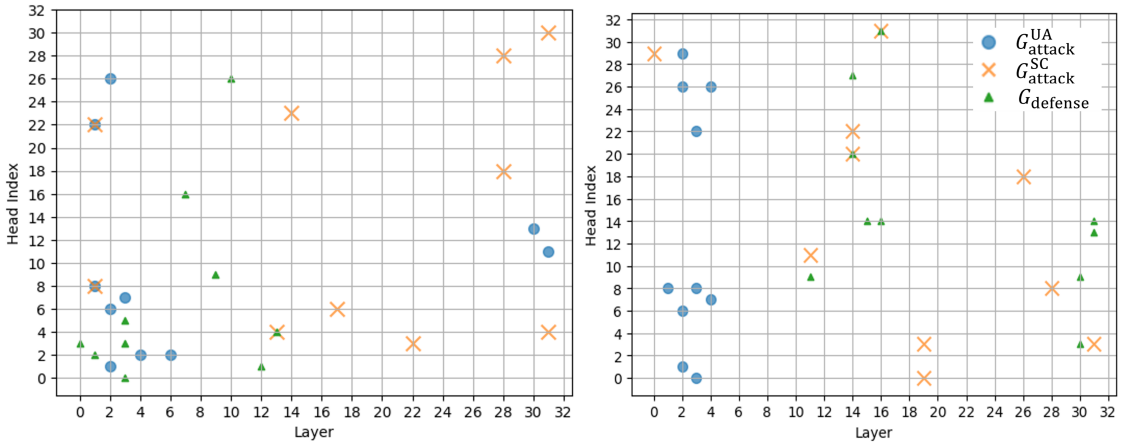


Figure 5: Comparison between attention heads effective for attacks and for defenses. (Left: Llama-2-7b-chat, Right: Mistral-7B-Instruct-v0.2)

tested on 7B-parameter models, but since the intervention is lightweight, it is expected to be applicable to larger open models as well.

Finally, while the token "Sure" was used as an indicator because it often appears in jailbreak outputs, other variants could be employed. Extending this idea and exploring interventions in other model components remain promising directions for future work.

Acknowledgments

This work was supported in part by the Japan Society for the Promotion of Science through Grants-in-Aid for Scientific Research (C) (23K11111).

References

Yuntao Bai, Andy Jones, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Federico Bianchi, Mirac Suzgun, et al. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In *ICLR 2024*, 2024.

Patrick Chao et al. Jailbreaking black box large language models in twenty queries. In *SaTML 2025*, 2025.

Peng Ding, Jun Kuang, et al. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. *arXiv preprint arXiv:2311.08268*, 2024.

Kai Hu, Weichen Yu, et al. Efficient LLM jailbreak via adaptive dense-to-sparse constrained optimization. In *NeurIPS 2024*, 2024.

Hakan Inan, Kartikeya Upasani, et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

Neel Jain, Avi Schwarzschild, et al. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Xuan Li, Zhanke Zhou, et al. Deepinception: Hypnotize large language model to be jailbreaker. In *Neurips Safe Generative AI Workshop 2024*, 2024.

Xiaogeng Liu, Nan Xu, et al. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *ICLR 2024*, 2024.

Anay Mehrotra et al. Tree of attacks: Jailbreaking black-box llms automatically. In *NeurIPS 2024*, 2024.

Long Ouyang, Jeff Wu, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Mansi Phute, Alec Helbling, et al. LLM self defense: By self examination, LLMs know they are being tricked. In *The Second Tiny Papers Track at ICLR 2024*, 2024.

Paul Röttger, Hannah Kirk, et al. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *NAACL 2024*, 2024.

Eric Todd, Millicent L. Li, et al. Function vectors in large language models. In *ICLR 2024*, 2024.

Hugo Touvron, Louis Martin, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Boyi Wei, Kaixuan Huang, et al. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *ICML 2024*, 2024.

Yueqi Xie, Jingwei Yi, et al. Defending chatgpt against jailbreak attack via self-reminders. *Nat. Mac. Intell.*, 5(12):1486–1496, 2023.

Sibo Yi, Yule Liu, et al. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*, 2024.

Jiahao Yu, Xingwei Lin, et al. LLM-fuzzer: Scaling assessment of large language model jailbreaks. In *USENIX Security 24*, 2024.

Wei Zhao, Zhe Li, et al. Defending large language models against jailbreak attacks via layer-specific editing. In *EMNLP 2024*, 2024.

Lianmin Zheng, Wei-Lin Chiang, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Zhenhong Zhou, Haiyang Yu, et al. On the role of attention heads in large language model safety. In *ICLR 2025*, 2025.

Andy Zou, Zifan Wang, et al. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.