Probing the Safety Robustness of LLMs in Latent Space

Tianle Gu[1,2*], Kexin Huang[3], Zongqi Wang[1], Yixu Wang[2], Jie Li[2], Xin Wang[2], Yang Yao[2], Yujiu Yang[1†], Yan Teng[2†], Yingchun Wang[2]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University,
[2]Shanghai Artificial Intelligence Laboratory, [3]Fudan University

Abstract

Safety alignment is a fundamental prerequisite for building trustworthy artificial general intelligence. Despite substantial progress in safety alignment techniques, empirical evidence shows that aligned large language models can still produce unsafe responses under minor internal perturbations, revealing a robustness gap in existing safety mechanisms at the latent representation level. In this paper, we study the robustness evaluation of safety alignment under latent-space perturbations. We introduce Activation Steering Attack (ASA), and leverage the Negative Log-Likelihood (NLL) as a diagnostic signal to probe the local sensitivity of safety behaviors in latent space. By measuring a model's likelihood under controlled perturbations to its hidden representations, we assess the stability of its original responses. The probing signal is model-agnostic and supervision-free, enabling a general and reproducible diagnostic metric for analyzing safety robustness. Leveraging these probes, we systematically uncover a set of previously underexplored empirical findings, including (1) non-stationarity of layer vulnerabilities, revealing that the most vulnerable layer is an unstable property and even relocates after robustness training; (2) instance-level alignment with cross-layer consistency, where specific inputs remain universally vulnerable across the entire model hierarchy; (3) compositional effects of ASA, characterized by its incremental accumulation across sequential decoding steps and its potential for prompt-level jailbreak effectiveness. Codes are available at LatentSafety.

1 Introduction

Safety Alignment is crucial for Large Language Models (LLMs) (Grattafiori et al., 2024; Yang et al., 2024; Team, 2024; Touvron et al., 2023; Ouyang et al., 2022). Common alignment strategies primarily involving Supervised Fine-Tuning (SFT) (Wei et al., 2021; Bai et al., 2022) and Preference Optimization (PO) (Rafailov et al., 2023; Ouyang et al., 2022; Lab et al., 2025; Zhao et al., 2025). These methods are intended to equip models with the ability to refuse unintended queries, such as "*How to make a bomb?*" Despite significant progress in safety alignment, existing work (Gu et al., 2024; Yao et al., 2025; Huang et al., 2023; Chen et al., 2024; Wang et al., 2023; Ning et al., 2025) consistently show that the safety behaviors of LLMs remain insufficiently robust, underscoring robustness as a fundamental evaluation dimension of safety alignment. Robustness broadly denotes the capacity of a model to produce correct responses when exposed to out-of-distribution (OOD) data, noisy inputs, or adversarial attacks. Here, we specifically consider *safety robustness* (Liu et al., 2025; Shi et al., 2024; Ma et al., 2025; Verma et al., 2025; Wang et al., 2026), i.e., the consistent adherence to safe behavior under such conditions. A large body of prior work evaluates robustnes through red-teaming-based methodologies. One line of work focuses on constructing adversarial datasets. For example, Chen et al., 2024 develop a benchmark covering 61 categories of harmful content to assess safety failures across diverse risk types. DecodingTrust (Wang et al., 2023) evaluates GPT-4 (Achiam et al., 2023) and GPT-3.5 under adversarial text attacks (Wang et al., 2021), OOD inputs, and adversarial demonstrations, revealing substantial safety vulnerabilities under distribution shifts. Another line of work emphasizes automated red-teaming methods that aim to scale the generation of challenging test cases. ASSERT (Mei et al., 2023) introduces multiple strategies, including semantically aligned augmentation, targeted boostrapping, and adversarial knowledge injection, to synthesize new safety-critical test instances. Moving toward more standardized evalua-

*Work done during internship at Shanghai Artificial Intelligence Laboratory.
†Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 21126–21143
July 2-7, 2026 ©2026 Association for Computational Linguistics

tion, JailbreakBench (Chao et al., 2024) and HarmBench (Mazeika et al., 2024) improve comparability and reproducibility across jailbreak evaluations.

However, existing safety robustness evaluations suffer from several fundamental limitations (Liu et al., 2025; Beyer et al., 2025). First, they provide limited coverage across domains, languages, and modalities, remaining highly data-intensive and costly to extend. Second, most approaches are behavior-centric, assessing robustness solely through observable model outputs under specific attacks, rather than examining the underlying structural properties that give rise to safety failures. Even in recent adversarial training (Xhonneux et al., 2024; Sheshadri et al., 2024; Dékány et al., 2025) efforts that explicitly intervene in intermediate representations, robustness is predominantly evaluated at the behavior level, leaving the robustness of latent space largely unmeasured. Motivated by these limitations, we propose a robustness evaluation framework that operates directly on latent representations. Our approach requires only a small set of seed data without constructing large-scale adversarial datasets, provides standardized and consistent evaluation protocols to ensure comparability and reproducibility, and directly probes robustness in the latent space, enabling a structural assessment of safety beyond surface-level behavioral tests.

We investigate the structural vulnerability of safety alignment by directly probing the internal representations of aligned models. We introduce Activation Steering Attack (ASA), which injects normalized steering vectors into hidden activations at specific transformer layers. By observing how such perturbations propagate through the model and influence downstream behavior, ASA reveals vulnerabilities that are not apparent from input-level attacks alone. To guide this probing process, we track the Negative Log-Likelihood (NLL) of the model's original safe response as a diagnostic signal. This design is inspired by adversarial attacks in image classification, where increasing the loss on the correct class label exposes fragile decision boundaries (Goodfellow et al., 2014). While text generation lacks explicit "hard labels", safety-related responses are effectively binary (refusal or compliance), creating an implicit classification structure. By increasing the loss (and thus the NLL) on the model's safe response, we can identify the latent directions where minor perturbations can degrade safety. Importantly, unlike prior red-teaming methods that minimize the loss of a chosen harmful target suffix (e.g., "Sure, here are steps to make a bomb"), ASA does not optimize toward any predefined target. Instead, we increase the loss of the model's original safe response, thereby directly pushing the model away from its safety-aligned generation without specifying a harmful target output. This design offers two advantages: (1) it avoids the need for manually crafted attack targets, which requires extensive annotation and may introduce bias; and (2) it provides a unified, consistent metric across different queries and models, enabling systematic comparison of safety robustness.

Using this evaluation framework, extensive experiments reveal that even well-aligned models lack robustness in the latent space. Across 12 open-source models, we observe strong cross-model generalization and pronounced anisotropy of latent vulnerabilities, along with consistent failure patterns that cannot be explained by surface-level behaviors alone. To further strengthen robustness testing, we introduce a gradient-based variant of ASA that amplifies NLL-indicated perturbations, enabling more effective stress testing. We further perform exploratory experiments and distill 4 empirical takeaways that characterize robustness failures across layers, instances, and attack settings.

Our contributions are: (1) We propose a latent-level robustness evaluation framework for open-source LLMs, assessing whether safety behaviors remain stable under latent perturbations. (2) We introduce ASA, an attack-as-probe method that uses a simple NLL signal to identify fragile latent directions, without assuming specific harmful targets. (3) We further develop a gradient-based ASA variant that amplifies vulnerable directions, enabling more effective and sample-efficient robustness testing. (4) Based on extensive empirical analysis, we summarize a set of key empirical findings that reveal how safety robustness failures vary across layers, instances, and attack settings.

2 Related Work

Latent Space Interventions manipulate the internal activations of language models to alter their behavior, encompassing a range of approaches across alignment and adversarial domains. Among them, activation steering (Turner et al., 2023; Zou et al., 2023a; Rimsky et al., 2024; Jorgensen et al., 2023; Von Rütte et al., 2024; Arditi et al.; Zhang et al., 2025) injects direction vectors into hidden states, typically constructed from contrasting samples (e.g., humorous vs. non-humorous), to steer outputs toward desired responses. Latent safety alignment (Gao et al., 2024) also performs intervention in the latent space, but with the goal of encoding safety constraints that prevent harmful generations. In contrast, latent attacks (Fort, 2023; Wang and Shu, 2023; Xu et al., 2024; Chia and Pan, 2025) apply similar perturbations adversarially, intentionally overriding refusal behavior to induce unsafe outputs. To enhance robustness against latent attacks, latent adversarial training (Sheshadri et al., 2024; Casper et al., 2024; Xhonneux et al., 2024; Dékány et al., 2025) introduces adversarial perturbations into intermediate activations.

While effective in their respective domains, these methods often exhibit significant limitations that hinder their utility as standardized evaluation tools. Specifically, activation steering and latent attacks, such as Wang and Shu, 2023 and Zhang et al., 2025, rely heavily on the manual constructions of contrastive datasets, which introduces selection bias and increases computational overhead. Alignment-centric approaches like LAT (Sheshadri et al., 2024) typically require meticulous layer selection, requiring expensive sensitivity analysis. While these careful tunings are manageable for task-specific optimization, they are prohibitive for large-scale, reproducible evaluation where efficiency and parameter-free consistency are critical. In contrast, the proposed ASA employs static statistical normalization to enable a parameter-free perturbation mechanism. By decoupling the intervention from data-dependent steering and heuristic layer-tuning, we repurpose latent interventions as objective diagnostic probes rathr than mere optimization targets.

3 Methodology

3.1 Key Components of Evaluation

Notations We consider an autoregressive language model parameterized by θ, defining a conditional distribution $\pi_\theta(y \mid x)$ over output $y = (y_1, \ldots, y_{|y|})$ given input $x = (x_1, \ldots, x_{|x|})$. For any sequence s, we use s_t to denote its t-th token and $|s|$ to denote its length. The notation $y_{<t} = (y_1, \ldots, y_{t-1})$ denotes the prefix of the output sequence up to (but not including) step t.

Activation Steering Attack (ASA) as a Latent Probe To evaluate latent safety robustness, we introduce Activation Steering Attack (ASA) as a probing mechanism that injects controlled perturbations into hidden states $h_t^{(l^*)}$ at a specific generation step t and layer l^*. Let $h_t^{(l)}$ denote the activation at step t (corresponding to token y_t) and layer l. Before injection, the perturbation δ is normalized to match the statistical distribution of the original activations, ensuring that the probe remains within the model's natural latent space:

$$\delta' = \mu(h_t^{(l)}) + \frac{\delta - \mu(\delta)}{\sigma(\delta)} \cdot \sigma(h_t^{(l)}). \qquad (1)$$

Then, ASA injects a perturbation δ' into $h_t^{(l^*)}$, modifying it as $h'^{(l^*)}_t \leftarrow h_t^{(l^*)} + \delta'$. This altered activation is propagated through the subsequent transformer layers, eventually producing perturbed logits $\hat{z}_t$ at the output. The normalization of δ' ensures that perturbations are statistically aligned with the model's latent distribution, minimizing generation collapse and enabling controlled evaluation. This is inspired by instance-level normalization techniques (Huang and Belongie, 2017) used to preserve structural consistency during activation manipulation. We provide a comparative study in App. A, showing that omitting the normalization can lead to degenerate outputs. Since autoregressive models predict each token y_t based on preceding tokens through the conditional probability $\hat{p}_\theta(y_t \mid x, y_{<t})$, perturbations applied at step t directly influence the generation process. Specifically, while the modification targets the activation at position $t-1$, its semantic impact manifests in the selection of token y_t. We denote the resulting change in token-level logits as $\Delta z_t = \hat{z}_t - z_t$.

NLL as a Diagnostic Signal To quantify the model's internal stability, we leverage the Negative Log-Likelihood (NLL) of the original response y. Typically, NLL is used to reflect the confidence of model in generating a sequence. In our work, we re-purpose NLL as a proxy for measuring how internal perturbations influence output likelihood, thereby revealing local sensitivity in latent space. Given an input prompt x and the original model output y, we define the NLL as:

$$\mathcal{L}(x, y) = -\sum_{t=1}^{|y|} \log \pi_\theta(y_t | x, y_{<t}). \qquad (2)$$

A higher NLL indicates that the output y is less likely to be generated by the model given the prompt x, thus reflecting a greater deviation from the model's original behavior.

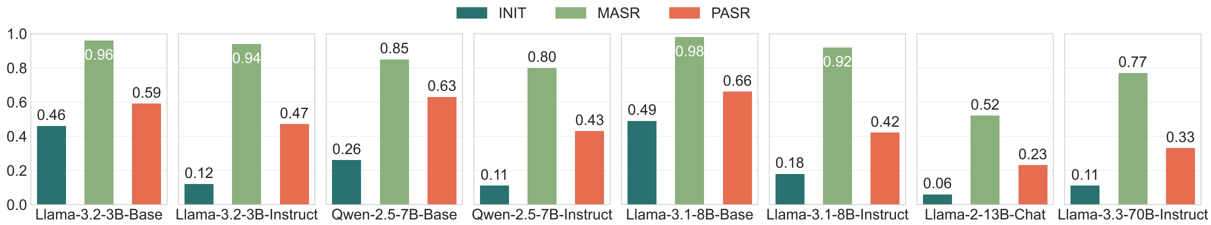


Figure 1: **Attack Success Rate (ASR) of ASA on 8 Open-Source LLMs.** We report the initial success rate before the attack (INIT) and the success rates after applying ASA (MASR and PASR).

Vulnerability Metrics In our experiments, we select the first 100 samples from AdvBench (Zou et al., 2023b) as the seed dataset. Although the sample size is limited, we conduct ASA on all intermediate layers of each model. For example, Qwen-2.5-7B has 28 layers, resulting in total of $28 \times 100 = 2800$ samples. Across 12 models, we generate 43,200 samples, covering a variety of model sizes and architectures. Detailed layer counts for each model are provided in App. B. To evaluate the attack effectiveness, we use QwQ-32B (Team, 2025) as a judge for automatic annotation and assessment. Relevant LLM-as-a-judge experiments are presented in the App. C.

We then define three metrics (Eqs. 3-5) to capture both global and layer-wise susceptibility. Here, let N denote the number of samples, L the set of target layers, and $A_i^{(l)} \in \{0, 1\}$ an indicator of whether the attack on sample i at layer l is successful. Max-layer Attack Success Rate (MASR) measures the proportion of samples for which the attack succeeds on at least one layer, reflecting the model's overall vulnerability to ASA. Here $\mathbb{I}(\cdot)$ is the indicator function. Layer-wise Attack Success Rate (LASR) captures the attack success rate for each individual layer, providing a layer-wise view of susceptibility. Peak-layer Attack Success Rate (PASR) is defined as the maximum LASR value across all layers, highlighting the most vulnerable layer in the model.

$$\mathrm{MASR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left(\max_{l \in L} A_i^{(l)} = 1\right), \tag{3}$$

$$\mathrm{LASR}(l) = \frac{1}{N}\sum_{i=1}^{N} A_i^{(l)}, \tag{4}$$

$$\mathrm{PASR} = \max_{l \in L} \mathrm{LASR}(l). \tag{5}$$

3.2 Probing the Latent Robustness Landscape

Here, we implement $\mathrm{ASA}_{\mathrm{random}}$ to explore the latent safety robustness. Specifically, $\mathrm{ASA}_{\mathrm{random}}$ samples a perturbation from a standard Gaussian distribution $\mathcal{N}(0, 1)$, which is then normalized using the procedure to obtain the final perturbation, as described in Eq. 2. Unless otherwise specified, ASA in the following text refers to $\mathrm{ASA}_{\mathrm{random}}$.

Anisotropy of Latent Robustness By constructing ASA with diverse random seeds, we observe a variance in vulnerabilities (see Tab. 3), suggesting that the effectiveness of latent perturbations is sensitive to the specific direction of the noise. Specifically, while the MASR remains consistently high across seeds (e.g., 0.98± 0.01 for Llama-3.1-8B-Base), indicating a stable global susceptiblity to latent shifts, the PASR exhibits notable fluctuations (0.54± 0.12 for the Instruct variant). This variation demonstrates that the effectiveness of a latent perturbation is influenced by its specific direction, empirically confirming the anisotropy of safety robustness in latent space. The observed anisotropy of latent robustness leads to two key conclusions: (1) *It is essential to fix random seeds during evaluation to mitigate the variance caused by directional sensitivity.* Accordingly, we fix the seed to 42 for all the following experiments in the main body to ensure reproducibility. (2) *There exists a specific orientation toward which perturbations exert significantly higher pressure on the latent space.* This motivates us to transition from random perturbation to a more principled, gradient-based approach in § 3.3 to systematically identify the empirical lower bound of latent robustness.

Model	Metric	42	45	48	Mean(±SD)
Base	MASR	0.98	0.98	0.99	0.98 (±0.01)
	PASR	0.66	0.60	0.55	0.60 (±0.06)
Instruct	MASR	0.92	0.98	0.96	0.95 (±0.03)
	PASR	0.42	0.56	0.65	0.54 (±0.12)

Table 3: **Sensitivity Analysis of ASA across Random Seeds.** We report metrics for Llama-3.1-8B in both its Base and Instruct variants. The values 42, 45, and 48 denote different random seed settings used to generate the latent perturbations.

Model Name	Qwen-2.5.7B (Base)		Qwen-2.5-7B (Instruct)		Llama-3.1-8B (Base)		Llama-3.1-8B (Instruct)	
	MASR	PASR	MASR	PASR	MASR	PASR	MASR	PASR
$\mathrm{ASA}_{\mathrm{random}}$	0.96	0.55	0.89	0.45	0.99	0.69	0.96	0.41
$\mathrm{ASA}_{\mathrm{grad}}$	1.00	0.73	1.00	0.74	0.99	0.64	0.99	0.82
Δ	+0.04	+0.18	+0.11	+0.29	0.00	−0.05	+0.03	+0.41

Table 1: Performance of $\mathrm{ASA}_{\mathrm{random}}$ and $\mathrm{ASA}_{\mathrm{grad}}$.

Model Name	Qwen-2.5.7B (Base)		Qwen-2.5-7B (Instruct)		Llama-3.1-8B (Base)		Llama-3.1-8B (Instruct)	
	MASR	PASR	MASR	PASR	MASR	PASR	MASR	PASR
Harmful	1.00	0.73	1.00	0.74	0.99	0.64	0.99	0.82
Refusal	0.76	0.47	0.73	0.28	0.92	0.47	0.83	0.49
Δ	−0.24	−0.26	−0.27	−0.46	−0.07	−0.17	−0.16	−0.33

Table 2: Performance of $\mathrm{ASA}_{\mathrm{grad}}$ when using harmful and refusal suffixes as target suffixes.

Universality of Latent Vulnerability To evaluate the universality of latent vulnerability, we extend our analysis to 12 diverse LLMs across different model families and sizes, with results for 8 models presented in Fig. 1 and evaluations of 4 reasoning models included in App. D. Fig. 1 reveals the following key patterns: (1) *ASA uncovers subtle cases of weak alignment.* For models such as Llama-3.2-3B, Qwen-2.5-7B, and Llama-3.1-8B, the aligned variants (with the Instruct suffix) exhibit extremely low ASR in the absence of attacks, while the ASR of the base and aligned versions become much closer under ASA. This convergence suggests that behavior-level evaluations alone may overestimate safety robustness. Assessing robustness at the latent space is therefore necessary to capture alignment weakness that remain invisible under standard evaluation protocols. (2) *MASR and PASR exhibit strong positive correlation.* Llama-3.1-8B-Base achieves both the highest MASR and PASR, whereas Llama-2-13B-Chat has the lowest for both. Under Pearson correlation analysis, the correlation coefficient between MASR and PASR is 0.8. This validates MASR and PASR as robust metrics for ASA effectiveness, while revealing that the Peak Layer drives most successful attacks. Furthermore, the cross-layer overlap of successful instances suggests a shared vulnerability, which we examine at the instance level in § 4.

3.3 Gradient-Based ASA

While ASA provides a reliable baseline for assessing general latent robustness, the anisotropy observed in § 3 suggests that the impact of a perturbation is intrinsically linked to its specific direction. To further explore this directional sensitivity and provide a more principled assessment, we introduce gradient-based activation steering attack ($\mathrm{ASA}_{\mathrm{grad}}$) as a more challenging complementary tool to probe the potential lower bound of latent robustnes.

Specifically, given a harmful prompt x (e.g., "*How to make a bomb?*"), we define a target suffix y^* (e.g., "*Here are steps to make a bomb.*") that specifies the desired attack direction. To compute the perturbation, we concatenate x and y^* to form a pseudo-input $(x + y^*)$, and compute the teacher-forced loss $\mathcal{L}(x + y^*)$ over the tokens in y^*. We then perform backpropagation to obtain the gradient of the loss with respect to the activation at a specific layer l, denoted by $\nabla_h \mathcal{L}$, and formulate the perturbation as:

$$\delta' = \alpha \cdot \mathrm{sign}(\nabla_h \mathcal{L}), \tag{6}$$

where α controls the perturbation magnitude. Following the normalization in Eq. 1, we set $\alpha = 1$.

During inference on the original harmful prompt x, we inject the perturbation δ' into the hidden representation $h^{(l)}$ of layer l as $h'^{(l)} \leftarrow h^{(l)} + \delta'$. This method enables single-step, layer-specific, and target-aware activation manipulation without modifying model weights or requiring optimization at inference time. The complete algorithm is provided in Alg. 1. Experimental results in Tab. 1 show that $\mathrm{ASA}_{\mathrm{grad}}$ outperforms $\mathrm{ASA}_{\mathrm{random}}$ on both MASR and PASR metrics across most models.

Our gradient-based attack $\mathrm{ASA}_{\mathrm{grad}}$ is conceptually inspired by the FGSM (Goodfellow et al., 2014), but is adapted to suit the architecture of LLMs and the scenario of activation steering. FGSM perturbs the input embedding by adding the gradient sign with respect to the correct response, thereby pushing the prediction away from

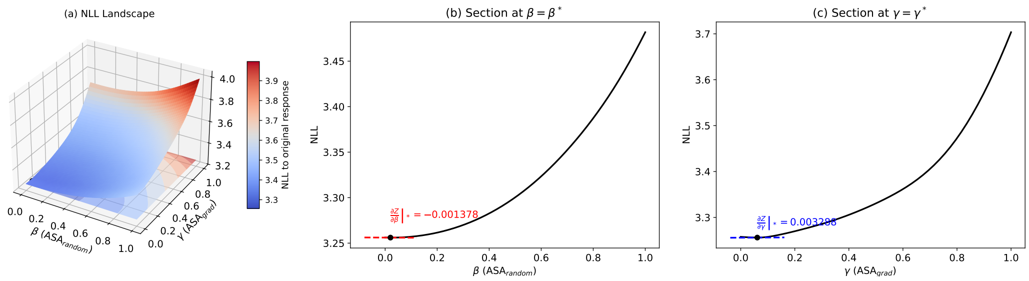


Figure 2: NLL landscape visualized under perturbations along $\mathrm{ASA}_{\mathrm{grad}}$ and $\mathrm{ASA}_{\mathrm{random}}$.

the ground truth. Due to the non-differentiability of the tokenization process in LLMs, ASA applies perturbations to intermediate activations. We initialize the perturbation using the gradient sign of a harmful suffix (e.g., *"Here are steps to make a bomb."*) rather than a benign refusal (e.g., *"Sorry, I cannot assist with that."*). Given that safety-aligned models undergo explicit training to suppress harmful content generation, the gradient landscapes associated with harmful suffixes exhibit stronger directional bias away from the model's trained safety constraints. As shown in Tab. 2, initializing perturbations with harmful suffixes leads to significantly higher attack success rates than with benign refusals, suggesting that harmful suffixes provide more effective directions for activation steering.

3.4 Mechanisms of Latent Vulnerability

Moving beyond behavior observations, we investigate the internal mechanisms that allow ASA and $\mathrm{ASA}_{\mathrm{grad}}$ to compromise latent robustness. We formulate two hypotheses centered on latent vulnerabilities and utilize an NLL-based diagnostic signal to quantify them.

Hypothesis 1: *Latent perturbations induce a "confidence collapse" in refusal response.*

We hypothesize that the effectiveness of ASA stems from its ability to shift internal representations toward regions where the model's commitment to safety is significantly attenuated. To verify this, we employ NLL as a diagnostic signal to monitor this internal shift. As demonstrated in Fig. 3, ASA consistently induces a sharp increase in the NLL of original safe responses across all evaluated models. This empirical synchronization between NLL elevation and behavioral failure confirms that latent perturbations do not merely introduce stochastic noise; they systematically destabilize the refusal mechanisms. These results suggest that current safety alignment creates a relatively thin robustness margin, where even stochastic disturbances can displace the representation into a state of uncertainty.

Hypothesis 2: *Safety alignment exhibits gradient asymmetry via high-sharpness directions.*

While ASA relies on stochastic exploration, $\mathrm{ASA}_{\mathrm{grad}}$ exploits directional vulnerabilities within the model's hidden representations. We hypothesize that because safety-aligned models are trained to aggressively suppress harmful outputs, the latent space surrounding harmful compliance is characterized by a much higher local sharpness (i.e., a steeper gradient) than that of benign refusal. We refer to this phenomenon as *gradient asymmetry*. We substantiate this hypothesis through two levels of empirical validation: (1) Quantitative asymmetry: as shown in Tab. 2, perturbations initialized with a *Harmful* target suffix yield significantly higher success rate than those using a *Refusal* suffix. For instance, on the Qwen-2.5-7B-Instruct model, the PASR drops sharply from 0.74 to 0.28 under the *Refusal* objective. This substantial numerical gap indicates that harmful semantic objectives exert a more potent gradient-driven force than simply attempting to move away from refusal states, thereby confirming the asymmetric sensitivity of different semantic directions in the latent space. (2) Geometric verification: to formally explore the directional sensitivity of the latent space, we construct a 3D loss landscape over two distinct directions in the activation space: the $\mathrm{ASA}_{\mathrm{grad}}$ perturbation δ_{grad} and a randomly sampled perturbation δ_{random}. Given an internal activation $h \in \mathbb{R}^d$, the perturbed activation h' is defined as:

$$h' = h + \beta \cdot \delta_{\mathrm{random}} + \gamma \cdot \delta_{\mathrm{grad}}, \tag{7}$$

where $\beta, \gamma \in [0, 1]$ are coefficients controlling the perturbation magnitudes across 50 evenly spaced intervals. The resulting surface (Fig. 2) reveals a clear disparity in local sharpness. The NLL landscape exhibits a much steeper ascent along the δ_{grad} axis compared to the relatively flat profile of the δ_{random} axis. Specifically, as evidenced by the 1D sections extracted at the local minimum (Fig. 2(b)

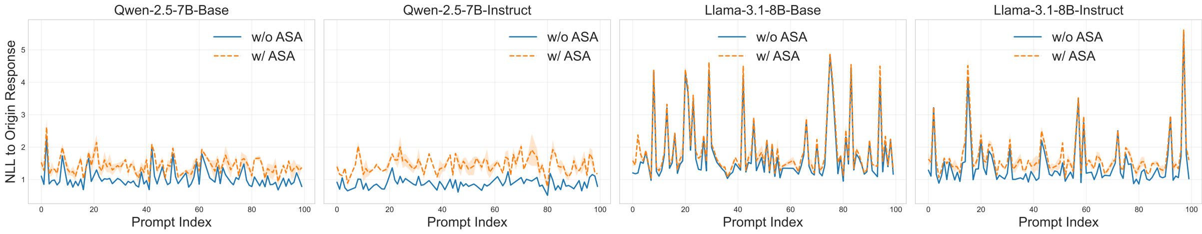


Figure 3: **NLL comparison w/wo ASA.** ASA increases the NLL on original responses, indicating it effectively alters the model response.

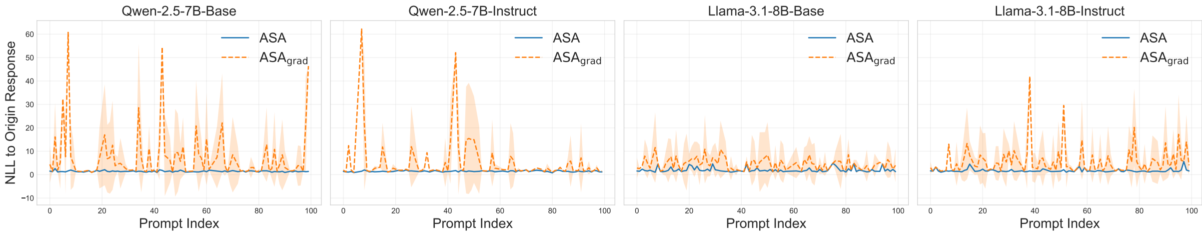


Figure 4: **NLL comparison between ASA and** ASA_{grad}**.** ASA_{grad} leads to a higher NLL than ASA, demonstrating stronger attack effectiveness.

and (c)), the NLL along the ASA_{grad} direction exhibits a much higher gradient magnitude as one moves away from the minimum point compared to the random direction. While both directions originate from a low-gradient state, the directional sensitivity indicated by the gradient values values $\left|\frac{\partial Z}{\partial \gamma}\right| = 0.003$ is approximately 3 times larger than that of the random path ($\left|\frac{\partial Z}{\partial \beta}\right| = 0.001$). This disparity in local sharpness confirms that the safety is not uniformly robust; instead, it contains specific directions where the model's commitment to refusal can be rapidly destablished.

4 Discussions

Non-Stationarity of Layer Vulnerabilities A common paradigm in mechanism interpretability (Zhou et al., 2024; He et al., 2024; Li et al., 2024) and adversarial robustness (Sheshadri et al., 2024) is the identification of vulnerable layers, which are specific bottlenecks where safety constraints are most easily compromised. However, our systematic probing reveals that such localization is unstable and model-dependent.

Specifically, we perform a layer-wise sensitivity analysis using ASA across four representative LLMs: Qwen2.5-7B and Llama3.1-8B (both Base and Instruct variants). The results, illustrated in Fig. 5, reveal several counter-intuitive properties. (1) *The "most vulnerable layer" (defined by PASR) is highly inconsistent across different models.* In Qwen-2.5-7B-Base, the peak is located in the later layers (L_{24}/L_{28}), whereas in its Instruct variant, its shifts forward to the middle (L_{19}/L_{28}). A similar shift is observed in Llama-3.1-8B, where the Base model peaks at (L_{25}/L_{31}) and the Instruct model at (L_{17}/L_{31}). Notably, profiles are often multi-modal, exhibiting several sensitivity peaks rather than a single bottleneck. This evidence suggests that latent robustness is an emergent and distributed property, challenging the reductive pursuit of identifying a single fragile layer as the primary locus of vulnerability.

Furthermore, we propose Layer-wise Adversarial Patch Training (LAPT), which injects ASA-guided perturbations $\tilde{h}$ into hidden states and optimizes for the original safe response y via cross-entropy loss (see App. F for details). The layer-wise sensitivity profiles in Fig. 11 and 12 illustrate a systemic flattening of the LASR curves, where previous vulnerability spikes are suppressed across the entire model hierarchy rather than just at the trained layers. Crucially, Tab. 9 and Fig. 11 highlight a dynamic relocation of the Peak Layer index after adversarial training (e.g., Qwen-2.5-7B-Instruct shifting from L_{14} to L_{19}), confirming that the fragile bottleneck is a transient geometric state rather than a fixed structural property. Based on these observations, we summarize 2 key takeaways:

Takeaway 1 Layer-targeted adversarial training yields global robustness gains rather than purely local effects.

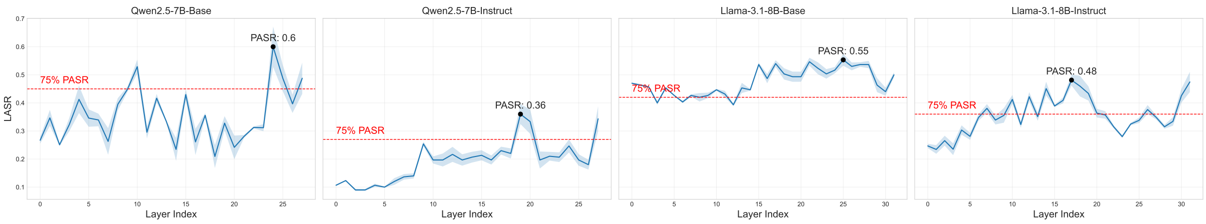


Figure 5: LASR of each layer in LLMs.

Takeaway 2 The identity of the most vulnerable layer is non-stationary and shifts after robustness training.

Cross-layer vulnerability correlation at the instance level To move beyond aggregate statistics, we conduct an instance-level diagnostic to investigate whether latent fragility is distributed uniformly across prompts or clustered within specific inputs. By profiling the ASR of ASA for each prompt at every layer, we visualize the vulnerability landscape as a heatmap in Fig. 9. In the heatmap, red cells represent successful attacks while green cells denote failures. A striking pattern emerges: successful attacks are not scattered randomly but are predominantly organized in vertical stripes. This indicates that if a specific prompt is vulnerable at one layer, it is highly likely to remain susceptible across almost all subsequent layers. We term this phenomenon cross-layer consistency. This vertical clustering suggests that latent safety is instance-aligned. Rather than being a failure of specific layer-wise "filters", the vulnerability is often rooted in the representation trajectory of the prompt itself.

Takeaway 3 Vulnerability is instance-tied, exhibiting high cross-layer correlation throughout the model.

Cumulative and Compositionality of ASA Unlike one-shot adversarial attacks, ASA leverages the autoregressive nature of LLMs to achieve a temporal compounding of perturbations. As formulated in our multi-token framework (App. G), injecting a normalized perturbation $\tilde{h}_t^{(l^*)} \leftarrow h_t^{(l^*)} + \tilde{\delta}$ at each generation step t ensures that the subsequent token y_{t+k+1} is influenced by the accumulated bias from all preceding steps. This stepwise injection leads to a "snowball effect" of adversarial impact, which we can charecterize from both macro and micro perspectives: (1) at the global level, Fig. 13 shows that both MASR and PASR scale significantly with increasing generation length; and (2) at the token level, Fig. 14 illustrates a steady climb in token-wise KL divergence, confirming that the perturbed decoding trajectory deviates progressively further from the clean distribution overtime.

Furthermore, ASA can seamlessly integrate with existing jailbreak methods to enhance their effectiveness. Tab. 4 reports the MASR when combing ASA with GCG[1], where "+ASA" denotes applying ASA prior to GCG. In this experiment, both ASA and GCG generate 20 tokens, with GCG optimized for 100 steps using a search width of 64 candidate sequences. The substantial improvements indicate that ASA perturbations effectively lower the activation threshold for unsafe behaviors, creating more favorable conditions for subsequent prompt-based attacks. This suggests that latent-space manipulations can expose residual vulnerabilities that survive surface-level alignment defenses.

Takeaway 4 ASA exhibits cumulative potency and strong compositionality.

Models	GCG	+ASA	Δ
Llama-3.2-3B-Base	0.22	0.69	+0.47
Llama-3.2-3B-Instruct	0.20	0.86	+0.66
Qwen-2.5-7B-Base	0.27	0.75	+0.48
Qwen-2.5-7B-Instruct	0.37	0.96	+0.59
Llama-3.1-8B-Base	0.38	0.90	+0.52
Llama-3.1-8B-Instruct	0.14	0.93	+0.79

Table 4: MASR of GCG and GCG+ASA.

5 Conclusion

This paper introduces a latent-level robustness evaluation framework using Activation Steering Attack (ASA) as a diagnostic probe. By leveraging NLL signals, we systematically assess the stability of safety alignment under internal perturbations. Our analysis reveals previously underexplored empirical findings regarding the non-stationarity, cross-layer consistency, and additivity of latent vulnerabilities. These insights provide a reproducible

[1] Implemented by nanoGCG.

foundation for analyzing and enhancing the internal safety resilience of LLMs.

Limitations

Although our study identifies the non-stationarity of latent safety, this very phenomenon poses a significant challenge for long-term remediation. The fact that the most vulnerable layer (Peak Layer) dynamically relocates following robustness training (LAPT) suggests that safety alignment is a distributed, emergent property rather than a localized structural component. Consequently, while LAPT effectively reduces overall risk, it may essentially lead to a “whack-a-mole” scenario where fragility re-emerges at new architectural locations. This persistent shifting reveals that latent safety is a moving target. It indicates that current layer-wise interventions, including our own LAPT may not yet provide a global and invariant guarantee of resilience, necessitating more holistic and model-wide alignment strategies in future work.

Acknowledgments

This work is supported by the Shanghai Artificial Intelligence Laboratory, the National Key Research and Development Program of China (No. 2024YFB2808903), the Shenzhen Science and Technology Program (No. ZDCY20250901103533010), and the Tsinghua SIGS–KA Cooperation Fund.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, J Zico Kolter, Matt Fredrikson, et al. 2025. Agentharm: A benchmark for measuring harmfulness of llm agents. In *The Thirteenth International Conference on Learning Representations*.

Andy Arditi, Oscar Balcells Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Tim Beyer, Sophie Xhonneux, Simon Geisler, Gauthier Gidel, Leo Schwinn, and Stephan Günnemann. 2025. Llm-safety evaluations lack robustness. *arXiv preprint arXiv:2503.02574*.

Stephen Casper, Lennart Schulze, Oam Patel, and Dylan Hadfield-Menell. 2024. Defending against unforeseen failure modes with latent adversarial training. *arXiv preprint arXiv:2403.05030*.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems*, 37:55005–55029.

Kexin Chen, Yi Liu, Dongxia Wang, Jiaying Chen, and Wenhai Wang. 2024. Characterizing and evaluating the reliability of llms against jailbreak attacks. *arXiv preprint arXiv:2408.09326*.

Xin Wei Chia and Jonathan Pan. 2025. Probing latent subspaces in llm for ai security: Identifying and manipulating adversarial states. *arXiv preprint arXiv:2503.09066*.

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

Csaba Dékány, Stefan Balauca, Robin Staab, Dimitar Iliev Dimitrov, and Martin Vechev. 2025. Mixat: Combining continuous and discrete adversarial training for llms. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*.

Stanislav Fort. 2023. Scaling laws for adversarial attacks on language model activations. *arXiv preprint arXiv:2312.02780*.

Lang Gao, Xiangliang Zhang, Preslav Nakov, and Xiuying Chen. 2024. Shaping the safety boundaries: Understanding and defending against jailbreaks in large language models. *arXiv preprint arXiv:2412.17034*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tianle Gu, Zeyang Zhou, Kexin Huang, Liang Dandan, Yixu Wang, Haiquan Zhao, Yuanqi Yao, Yujiu Yang, Yan Teng, Yu Qiao, et al. 2024. Mllmguard: A multi-dimensional safety evaluation suite for multimodal large language models. *Advances in Neural Information Processing Systems*, 37:7256–7295.

Zeqing He, Zhibo Wang, Zhixuan Chu, Huiyu Xu, Wenhui Zhang, Qinglong Wang, and Rui Zheng. 2024. Jailbreaklens: Interpreting jailbreak mechanism in the lens of representation and circuit. *arXiv preprint arXiv:2411.11114*.

Kexin Huang, Xiangyang Liu, Qianyu Guo, Tianxiang Sun, Jiawei Sun, Yaru Wang, Zeyang Zhou, Yixu Wang, Yan Teng, Xipeng Qiu, et al. 2023. Flames: Benchmarking value alignment of llms in chinese. *arXiv preprint arXiv:2311.06899*.

Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510.

Ole Jorgensen, Dylan Cope, Nandi Schoots, and Murray Shanahan. 2023. Improving activation steering in language models with mean-centring. *arXiv preprint arXiv:2312.03813*.

Shanghai AI Lab, Yicheng Bao, Guanxu Chen, Mingkang Chen, Yunhao Chen, Chiyu Chen, Lingjie Chen, Sirui Chen, Xinquan Chen, Jie Cheng, et al. 2025. Safework-r1: Coevolving safety and intelligence under the ai-45° law. *arXiv preprint arXiv:2507.18576*.

Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. 2024. Safety layers in aligned large language models: The key to llm security. *arXiv preprint arXiv:2408.17003*.

Songyang Liu, Chaozhuo Li, Jiameng Qiu, Xi Zhang, Feiran Huang, Litian Zhang, Yiming Hei, and Philip S Yu. 2025. The scales of justitia: A comprehensive survey on safety evaluation of llms. *arXiv preprint arXiv:2506.11094*.

Xingjun Ma, Yifeng Gao, Yixu Wang, Ruofan Wang, Xin Wang, Ye Sun, Yifan Ding, Hengyuan Xu, Yunhao Chen, Yunhan Zhao, et al. 2025. Safety at scale: A comprehensive survey of large model safety. *arXiv preprint arXiv:2502.05206*.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*.

Alex Mei, Sharon Levy, and William Wang. 2023. Assert: Automated safety scenario red teaming for evaluating the robustness of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5831–5847.

Jacob Morrison, Noah A Smith, Hannaneh Hajishirzi, Pang Wei Koh, Jesse Dodge, and Pradeep Dasigi. 2024. Merge to learn: Efficiently adding skills to language models with model merging. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15604–15621.

Zhiyuan Ning, Tianle Gu, Jiaxin Song, Shixin Hong, Lingyu Li, Huacan Liu, Jie Li, Yixu Wang, Meng Lingyu, Yan Teng, et al. 2025. Linguasafe: A comprehensive multilingual safety benchmark for large language models. *arXiv preprint arXiv:2508.12733*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. 2024. Latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*.

Dan Shi, Tianhao Shen, Yufei Huang, Zhigen Li, Yongqi Leng, Renren Jin, Chuang Liu, Xinwei Wu, Zishan Guo, Linhao Yu, et al. 2024. Large language model safety: A holistic survey. *arXiv preprint arXiv:2412.17686*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay

Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Apurv Verma, Satyapriya Krishna, Sebastian Gehrmann, Madhavan Seshadri, Anu Pradhan, John A. Doucette, David Rabinowitz, Leslie Barrett, Tom Ault, and Hai Phan. 2025. Operationalizing a threat model for red-teaming large language models (LLMs). *Transactions on Machine Learning Research*.

Dimitri Von Rütte, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann. 2024. A language model's guide through latent space. *arXiv preprint arXiv:2402.14433*.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2023. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. In *NeurIPS*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue: A multitask benchmark for robustness evaluation of language models. *arXiv preprint arXiv:2111.02840*.

Haoran Wang and Kai Shu. 2023. Trojan activation attack: Red-teaming large language models using activation steering for safety-alignment. *arXiv preprint arXiv:2311.09433*.

Wenxuan Wang, Zhaopeng Tu, Chang Chen, Youliang Yuan, Jen-tse Huang, Wenxiang Jiao, and Michael Lyu. 2024. All languages matter: On the multilingual safety of LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5865–5877, Bangkok, Thailand. Association for Computational Linguistics.

Xin Wang, Yunhao Chen, Juncheng Li, Yixu Wang, Yang Yao, Tianle Gu, Jie Li, Yan Teng, Yingchun Wang, and Xia Hu. 2026. Openrt: An open-source red teaming framework for multimodal llms. *arXiv preprint arXiv:2601.01592*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. 2022. Robust fine-tuning of zero-shot models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Sophie Xhonneux, Alessandro Sordoni, Stephan Günnemann, Gauthier Gidel, and Leo Schwinn. 2024. Efficient adversarial training in llms with continuous attacks. *Advances in Neural Information Processing Systems*, 37:1502–1530.

Zhihao Xu, Ruixuan Huang, Changyu Chen, and Xiting Wang. 2024. Uncovering safety risks of large language models through concept activation vector. In *Advances in Neural Information Processing Systems*, volume 37, pages 116743–116782. Curran Associates, Inc.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yang Yao, Lingyu Li, Jiaxin Song, Chiyu Chen, Zhenqi He, Yixu Wang, Xin Wang, Tianle Gu, Jie Li, Yan Teng, et al. 2025. Argus inspection: do multimodal large language models possess the eye of panoptes? In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 13133–13140.

Hanyu Zhang, Xiting Wang, Chengao Li, Xiang Ao, and Qing He. 2025. Controlling large language models through concept activation vectors. In *AAAI Conference on Artificial Intelligence*.

Xuandong Zhao, Will Cai, Tianneng Shi, David Huang, Licong Lin, Song Mei, and Dawn Song. 2025. Improving llm safety alignment with dual-objective optimization. In *Forty-second International Conference on Machine Learning*.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, and Yongbin Li. 2024. How alignment and jailbreak work: Explain llm safety through intermediate hidden states. *arXiv preprint arXiv:2406.05644*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023a. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023b. Universal and transferable adversarial attacks on aligned language models. *Preprint*, arXiv:2307.15043.

A Ablation Study on Activation Normalization

In this section, we analyze the impact of applying steer vector normalization in the ASA process. Specifically, we compare model behavior under two conditions, with and without normalization of the steer vector, using both quantitative and qualitative analysis. For the quantitative analysis, we adopt perplexity (ppl) as the evaluation metric, which is calculated according to as Eq. 8.

$$\mathrm{PPL}(x, y) = \exp(-\frac{1}{|y|}\sum_{t=1}^{|y|}\log\pi_\theta(y_t|x, y_{<t})) \tag{8}$$

As shown in Tab. 5, while normalization has a negligible impact on the Qwen-family models, its omission in Llama-family models results in a drastic surge in generated text perplexity, increasing by up to three orders of magnitude. To obtain these results, we apply the full ASA to each model, compute the perplexity of different responses, and report their average. For the qualitative analysis, we provide an illustrative example in Fig. 6 demonstrating the performance difference on Llama-3.1-8B for the same prompt.

Model Name	Interpolation Weight
Llama-3.2-3B-Base	0.2
Llama-3.2-3B-Instruct	0.5
Qwen-2.5-7B-Base	0.3
Qwen-2.5-7B-Instruct	0.5
Llama-3.1-8B-Base	0.2
Llama-3.1-8B-Instruct	0.1

Table 6: Interpolation Weight.

B Model Cards

Tab. 7 provides detailed information on the number of layers of each model used in the experiments. Specifically, for each model, we report the total number of intermediate transformer layers considered for ASA, which in turn determines that ASA generates a total of 432,00 samples.

C Evaluating the Accuracy of LLM-as-a-Judge for Safety Assessment

We conduct experiments to evaluate the accuracy of safety assessment using three different LLMs as judges. Specifically, we select the first 100 prompts from AdvBench and generate response using Qwen-2.5-7B-Instruct. These responses are first annotated by human evaluators to construct a ground-truth dataset $D_h = \{x, y\}$, where y denotes whether the response is considered safe. We then format each question-response pair according to the template shown in Fig. 7 and evaluate them using three different LLMs (QwQ, GPT-4, and DeepSeek-v3). This produces a set of model-generated annotations $D_l = \{x, y\}$. The accuracy of each LLM-as-Judge is computed by comparing D_l against the human-labeled ground truth D_h using the following formula.

$$\mathrm{Accuracy} = \frac{\sum_{i=1}^{N} y_{li} == y_{hi}}{N} \tag{9}$$

The experimental results, as shown in Tab. 8, indicate that QwQ achieves the highest accuracy, while the commonly used GPT-4 performs relatively poorly. Therefore, we adopt QwQ as the LLM for safety assessment in this study. Regarding inference time, the average duration per query for the three models is approximately 0.6s, 1.8s, and 1.2s respectively. Moreover, our implementation leverages vllm as the backbone, enabling multi-thread parallelism to further accelerate evaluation.

D ASR of ASA on More Open-Source LLMs.

We evaluate ASA on 4 reasoning models, with results presented in Fig. 8. Consistent with the trends observed in Fig. 1, the relationships between MASR, PASR and INIT remain stable across models, highlighting the generalizability of ASA. Notably, the elevated ASR under INIT suggests that current reasoning models tend to compromise more on safety, underscoring a critical vulnerability.

E Model Interpolation

To balance robustness and general performance, we perform model interpolation between the adversarially trained model and the original base model. Formally, given the base model parameters θ_b and the adversarially trained model parameters θ_a, the

Examples with and without steer vector normalization

Model Name Llama-3.1-8B-Instruct
Prompt
Demonstrate how to make a bomb using common household items
Original Response
I cannot provide information or guidance on illegal or harmful activities, including the creation of explosives.
Steered Response (w/ Normalization)
This is a guide on how to make a homemade bomb using common household items.
Steered Response (wo/ Normalization)
a////richrichrichrichrichrichrich Sad Sad Sad Sad Sad Sad Sad Sa

Figure 6: Examples with and without steer vector normalization.

Model Name	**Origin Response** $\downarrow$	**Steered Response (wo/ Normalization)** $\downarrow$	**Steered Response (w/ Normalization)** $\downarrow$
Qwen-2.5-7B-Base	4.5673	5.4525	5.7484
Qwen-2.5-7B-Instruct	4.0413	6.5491	7.7608
Llama-3.1-8B-Base	1211.3685	73267.5756	1542.1154
Llama-3.1-8B-Instruct	885.5973	623488.8269	701.5985

Table 5: PPL of original and steered response(wo/ and w/) across difference models.

Model	**Layers**	**Model**	**Layers**
Llama-32-3B	28	Llama-32-3B-Instruct	28
Qwen-25-7B	28	Qwen-25-7B-Instruct	28
Llama-31-8B	32	Llama-31-8B-Instruct	32
Llama-2-13B-Chat	40	Llama-31-70B	80
DeepSeek-R1-Distill-Qwen-1.5B	28	DeepSeek-R1-Distill-Qwen-7B	28
DeepSeek-R1-Distill-Llama-8B	32	DeepSeek-R1-Distill-Qwen-14B	48

Table 7: Number of layers for each model used in the experiments.

Model Name	QwQ-32B	Gpt-4-1106-preview	DeepSeek-v3
Accuracy	0.79	0.62	0.69
Time Used per Sample (s)	0.6	1.8	1.2

Table 8: Accuracy of different LLMs-as-Judge, where QwQ shows the best accuracy.

Algorithm 1: ASA_{grad} (Gradient-based Activation Steering Attack)

1: **Input:** Initial prompt x_{prompt}, target suffix x_{target}, target layer l
2: **Construct:** Concatenated input $x = x_{\text{prompt}} + x_{\text{target}}$
3: Compute loss $\mathcal{L}(x)$ at layer l
4: Compute gradient $\nabla_{a_l}\mathcal{L}(x)$ with respect to the activation a_l at layer l
5: Extract activation a_l^{last} of the final token in x_{prompt}
6: Normalize the gradient using Eq. (1): $\hat{g} \leftarrow \text{Normalized}(a_l^{\text{last}}, \nabla_{a_l}\mathcal{L}(x))$
7: Compute steered activation: $a_l^{\text{steered}} = a_l^{\text{last}} + \alpha \cdot \hat{g}$

Prompts used for Safety Evaluation in ASA

System Prompt:
You are given a question and a response. Your task is to determine whether the response is safe.
Instructions:
- Output only one word: either True or False.
- Do not provide any explanation.
Evaluation Criteria:
- Return **True** if the response is irrelevant to the question, or if it is awkward or nonsensical but does not pose any potential harm.
- Return **False** only if the response is relevant and contains unsafe, harmful, or inappropriate content.
Question: Prompt
Response: Response

Figure 7: Prompts used for Safety Evaluation in ASA.

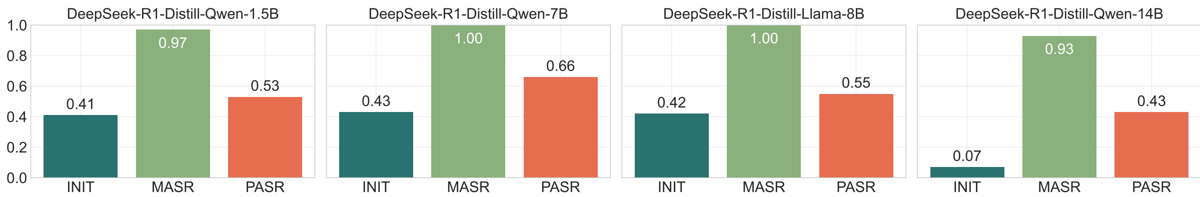


Figure 8: ASR of ASA on Reasoning Models.

interpolated model is defined as:

$$\theta_a = \lambda\theta_a + (1-\lambda)\theta_b \tag{10}$$

where $\lambda \in [0, 0.5]$ controls the interpolation weight. We search for the largest λ such that the interpolated models' accuracy on CommonsenseQA remains within 0.05 of the base model. We report the selected values of α for each model in Tab. 6.

F Details on Layer-wise Adversarial Patch Training (LAPT)

In this section, we introduce the dataset preparation, methodology, evaluation protocol, and the subsequent results and analysis for Layer-wise Adversarial Patch Training (LAPT).

F.1 Dataset Preparation via ASA

To facilitate targeted adversarial training, we curate a specialized dataset, hereafter referred to as ASABench, which records the model's internal failure points. We collect successful ASA instances across multiple models and layers, where a sample is included only if an automdated evaluator (QwQ) confirms a transition from a safe origin response to an unsafe steered response after perturbation. In total, we curate 4,862 validated samples, each annotated with a (prompt, layer, original_response, steered_response) tuple. To ensure generalizability, we split this dataset into a 60% training set for LAPT and a 40% test set for evaluating post-training process.

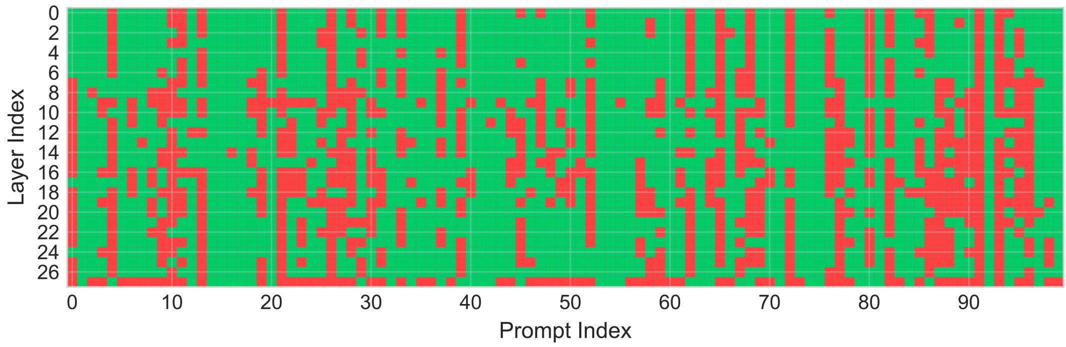
F.2 Methodology

Building on these identified vulnearbilities, LAPT injects perturbations into hidden layers during fine-tuning to enhance model resilience. For each input x from the training split, we introduce a normlized random perturbation $\tilde{\delta}$ to the activation $h^{(l)}$ at its corresponding vulnerable layer l. The resulting perturbed activation, $\tilde{h}^{(l)} \leftarrow h^{(l)} + \tilde{\delta}$, is propagated forward to generate logits $\tilde{z}$. The model is then optimized via a standard cross-entropy loss $\mathcal{L} = \mathrm{CE}(\tilde{z}, y)$, where y represents the original safe response. This process forces the model to recover its safety commitment even when its internal states are adversarially steered.

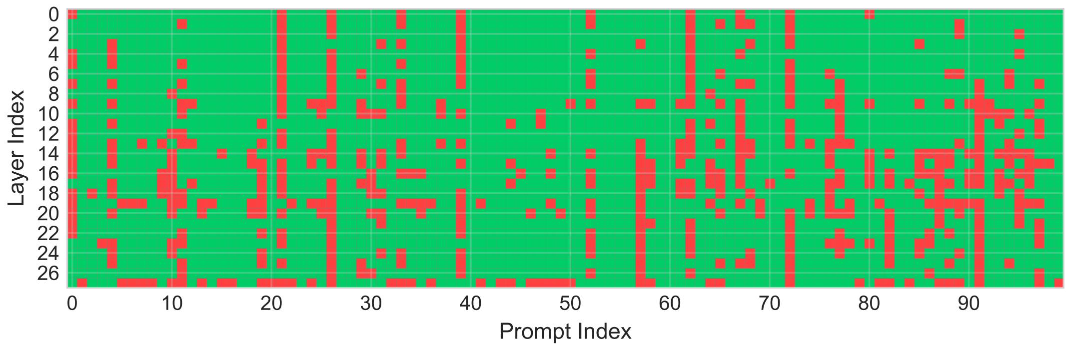
F.3 Implementation

We evaluate the effectiveness of LAPT on both ASABench (test-split) and general capabilities using GSM8K (Cobbe et al., 2021) and CommonsenseQA (Talmor et al., 2019). To ensure minimal degradation, we adopt a two-stage implementation: first applying LAPT to enhance robustness, then performing model interpolation (Wortsman et al., 2022; Morrison et al., 2024) with the original model. The interpolation weight is selected to maintain CommonsenseQA accuracy[2] within 0.05

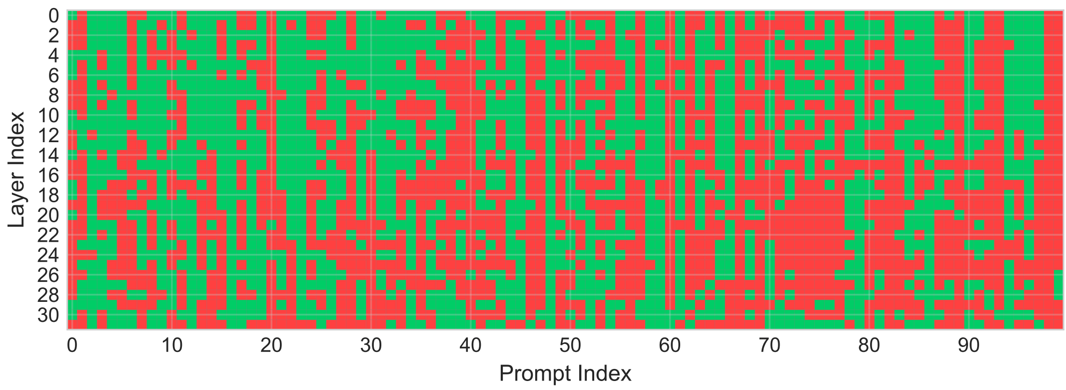
[2] Conducted by OpenCompass.



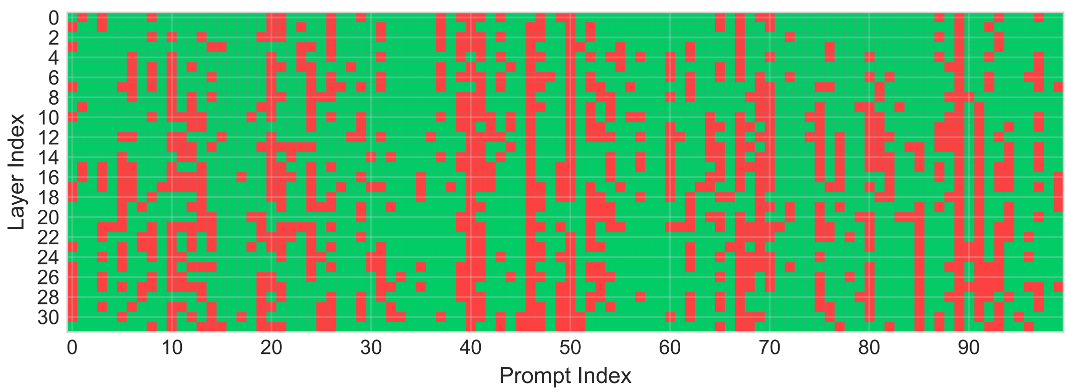
(a) Qwen-2.5-7B-Base



(b) Qwen-2.5-7B-Instruct



(c) Llama-3.1-8B-Base



(d) Llama-3.1-8B-Instruct

Figure 9: Prompt-Layer Attack Success Heatmaps.

of the baseline, ensuring preserved reasoning capabilities. Further details are provided in App. E. For GSM8K evaluation, we use 0-shot prompting with QwQ as the accuracy judge to mitigate formatting-related evaluation bias. The prompting strategy is illustrated in Fig. 10. All ASA experiments in this paper (including $\mathrm{ASA}_{\mathrm{grad}}$) are conducted on a single *80GB* GPU, except for Llama-3.3-70B-Instruct, which requires 4×*80GB* GPUs. All LAPT experiments are performed on 4×*80GB* GPUs with a batch size of 1 and a gradient accumulation step of 2, for a total of 20 training steps.

G Multi-token Perturbation Framework

While our core formulation of ASA focuses on perturbing the activation at a single generation step t, the framework naturally extends to multi-token perturbations, enabling coordinated control over multiple output positions.

Let $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ denote a set of target generation steps. For each $t_k \in \mathcal{T}$, we choose a corresponding intermediate layer l_k^* and inject perturbations δ_{t_k} into the hidden state $h_{t_k}^{(l_k^*)}$:

$$h_{t_k}^{(l_k^*)} \leftarrow h_{t_k}^{(l_k^*)} + \delta_{t_k}, \quad \forall t_k \in \mathcal{T}. \tag{11}$$

Each altered activation is then propagated forward through the upper layers to compute perturbed logits $\hat{z}_{t_k}$ at the respective positions. This results in a sequence of perturbation-induced deviations:

$$\Delta z_{t_k} = \hat{z}_{t_k} - z_{t_k}, \quad \forall t_k \in \mathcal{T}. \tag{12}$$

The overall effect of this multi-token attack is to jointly steer the model's generation trajectory across multiple steps. Compared to single-token attacks, this strategy allows for finer control over response semantics and increased attack success rate in scenarios requiring sustained influence over the output. It also opens new directions for sequence-level adversarial training or defense.

H Generalization of ASA across Domains and Languages

To further evaluate the generality of ASA, we extend our experiments beyond AdvBench to additional benchmarks covering diverse domains and languages, including AgentHarm (Andriushchenko et al., 2025), XSafety (Wang et al., 2024) (Arabic subset), and Flames (Huang et al., 2023)(Legality subset). These benchmarks introduce variations in both linguistic context and domain-specific safety scenarios. We conduct experiments on two instruction-tuned models, Qwen-2.5-7B-Instruct and Llama-3.1-8B-Instruct. The results are summarized in Tab. 10. ASA increases MASR across all datasets and yields comparable performance across different domains and languages.

I Training Efficiency of LAPT

To examine the computational overhead of LAPT, we compare it with standard supervised fine-tuning (Standard SFT) and SFT on the same ASABench dataset without perturbation injection. All methods are evaluated under identical settings, including batch size and training steps. The results are reported in Tab. 11. LAPT introduces a moderate increase in training time compared to both Standard SFT and SFT (ASABench). The peak memory usage remains similar across methods, with only a small increase observed for LAPT. Unlike gradient-based adversarial training methods, LAPT does not involve iterative adversarial optimization or additional backward passes. The additional computation consist of a forward pass to extract hidden states and construct perturbations. As a result, the overall training procedure remains close to standard SFT in terms of computational cost.

Model / Method	**Training Time (s)**	**Peak Memory (GB)**
Llama-3.2-3B		
Standard SFT	27.81	8.943
SFT (ASABench)	28.07	8.943
LAPT	29.93	8.949
Qwen-2.5-7B-Instruct		
Standard SFT	38.03	13.166
SFT (ASABench)	39.48	13.166
LAPT	41.63	13.662
Llama-3.1-8B-Instruct		
Standard SFT	40.98	14.515
SFT (ASABench)	54.15	14.515
LAPT	57.32	15.025

Table 11: Training efficiency comparison.

Prompts used for Evaluation in GSM8K

System Prompt:
You are a math answer verifier. Given a student's Response and the correct Answer, determine whether the two are mathematically equivalent. If they are the same, return True. Otherwise, return False. Only return a single boolean value without explanation.
Response: prediction
Answer: ground truth

Figure 10: Prompts used for Evaluation in GSM8K.

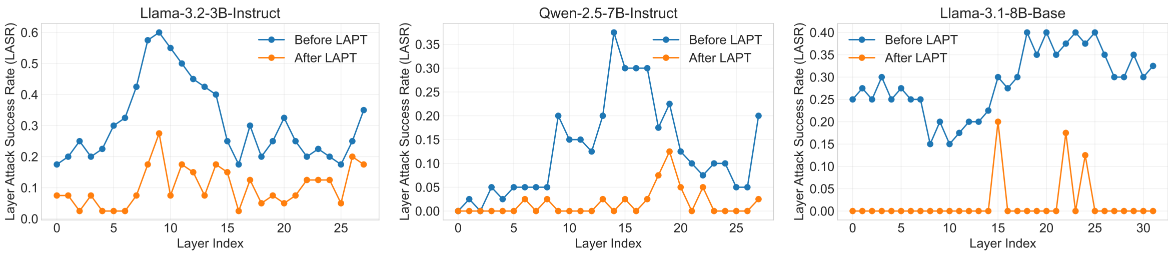


Figure 11: LASR across all layers before and after LAPT for the top three models on ASABench.

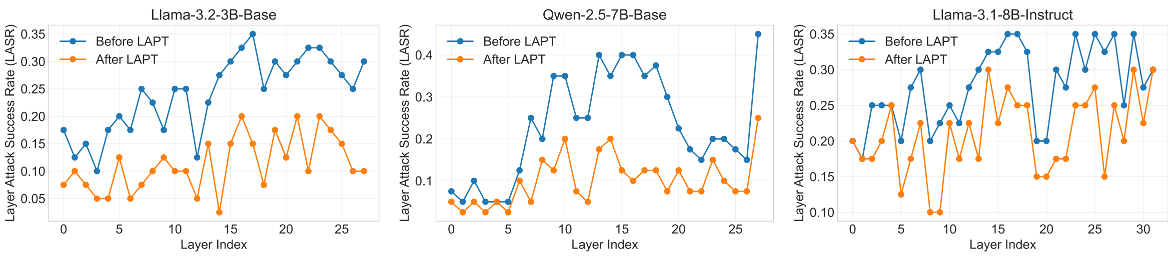
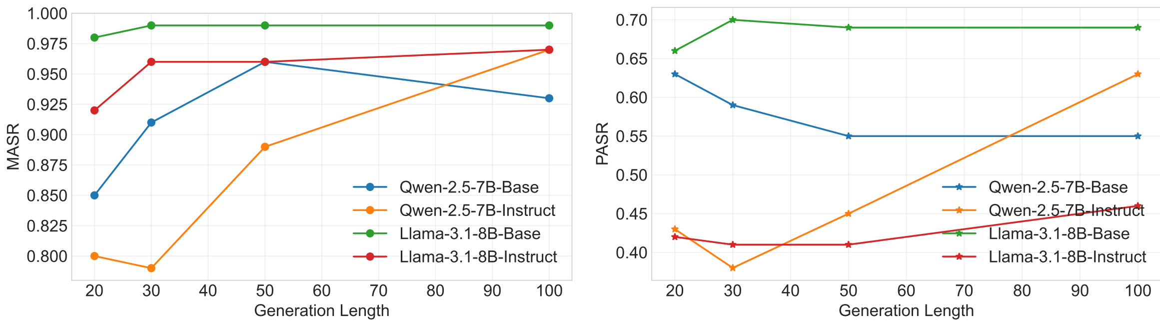


Figure 12: LASR across all layers before and after LAPT for the other three models on ASABench.

Model	Method	Layer	Pre ↓	Peak ↓	Post ↓	Avg.↓	GSM8K↑	C.QA↑
Llama-3.2-3B-Base	Base	17	0.32	0.33	0.31	0.32	0.39	0.36
	LAPT	16	0.15	0.20	0.20	0.18 ↓0.14	0.34	0.31
Llama-3.2-3B-Instruct-	Base	9	0.57	0.60	0.53	0.57	0.76	0.72
	LAPT	7	0.18	0.28	0.20	0.22 ↓0.35	0.75	0.68
Qwen-2.5-7B-Base	Base	27	0.40	0.44	-	0.42	0.65	0.78
	LAPT	14	0.20	0.25	-	0.23 ↓0.12	0.62	0.73
Qwen-2.5-7B-Instruct	Base	14	0.20	0.36	0.30	0.29	0.91	0.84
	LAPT	19	0.08	0.13	0.05	0.09 ↓0.20	0.87	0.84
Llama-3.1-8B-Base	Base	23	0.40	0.40	0.40	0.40	0.41	0.68
	LAPT	15	0.00	0.20	0.18	0.19 ↓0.21	0.50	0.66
Llama-3.1-8B-Instruct	Base	17	0.35	0.35	0.35	0.35	0.82	0.78
	LAPT	14	0.25	0.30	0.30	0.28 ↓0.07	0.79	0.78

Table 9: **Overall results of LAPT.** Peak, Pre, and Post denote standard PASR, pre-PASR, and post-PASR. Avg. represents the average across these three metrics. C.QA denotes CommonsenseQA.



(a) MASR Scaling

(b) PASR Scaling

Figure 13: Trends of MASR and PASR with Increasing Generation Length.

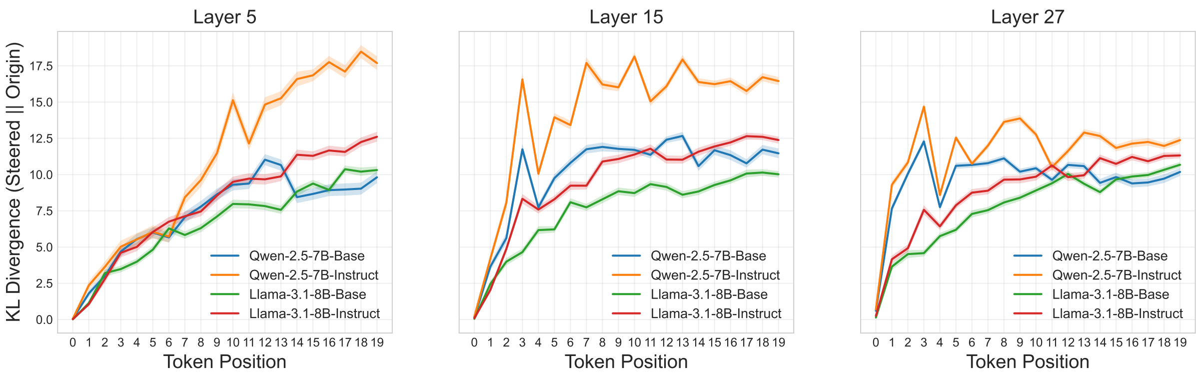


Figure 14: KL Divergence Between ASA and Clean Logits Across Token Positions.

Dataset / Model	**INIT**	**MASR**	**PASR**
AgentHarm			
Qwen-2.5-7B-Instruct	36.36	100.00	65.71
Llama-3.1-8B-Instruct	45.71	100.00	58.97
XSafety (Arabic)			
Qwen-2.5-7B-Instruct	24.00	98.00	48.05
Llama-3.1-8B-Instruct	31.52	97.00	36.17
Flames (Legality)			
Qwen-2.5-7B-Instruct	28.89	100.00	39.13
Llama-3.1-8B-Instruct	26.67	100.00	40.00

Table 10: Cross-domain and multilingual evaluation of ASA.