BEYOND ALIGNMENT: ENHANCING LLM SAFETY ROBUSTNESS VIA ZEROTH-ORDER OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety alignment for large language models (LLMs) aims to reduce harmful or unsafe behavior while preserving general utility. However, recent findings reveal that alignment effects can be fragile: lightweight post-alignment manipulations, such as parameter noise, activation noise, or quantization, can easily weaken the intended safety behavior. Prior efforts to improve robustness have primarily focused on data curation, modified alignment objectives, and safety-critical parameter identification, leaving the role of the optimizer itself largely unexplored.

In this paper, we are the first to study the robustness of safety alignment from the perspective of the base optimizer. This optimizer-centric view naturally points to zeroth-order optimization, which provides a robustness-oriented signal by evaluating safety alignment under perturbations. Based on this insight, we propose a hybrid framework that first performs standard first-order safety alignment and then applies zeroth-order refinement to improve robustness. Both theoretically and empirically, we show that only a few zeroth-order refinement steps can enhance robustness while preserving safety alignment. We further improve the efficiency of zeroth-order refinement by exploiting its inherent perturbation-based evaluations to estimate layer-wise robustness sensitivity, enabling the refinement process to concentrate updates on robustness-critical layers with modest training overhead.

1 INTRODUCTION

Large language models (LLMs) (Touvron et al., 2023; Qwen Team, 2025) have demonstrated remarkable capabilities across a wide range of tasks, ranging from question answering to complex instruction following. With their growing deployment in real-world settings, concerns about their reliability, safety, and robustness have become a major concern (Weidinger et al., 2021). To address these risks, developers introduce an additional safety alignment stage after pretraining, often through supervised fine-tuning or preference optimization. These methods mainly differ in alignment data, training objective, and model-level safety representations but are commonly optimized through gradient-based updates (Ouyang et al., 2022; Rafailov et al., 2023; Ethayarajh et al., 2024; Guan et al., 2025). This stage makes relatively small but targeted weight adjustments to discourage the generation of harmful, unethical, or dangerous content while preserving the model's general utility and instruction-following capability (Gehman et al., 2020; Liu et al., 2023).

Despite recent progress in LLM safety alignment, achieving ***robust*** safety alignment remains challenging. Even small post-alignment perturbations can substantially weaken safety behavior: Gaussian weight noise can induce unsafe responses (Tice et al., 2024), perturbations to intermediate activations can weaken safety guardrails (Clymer et al., 2024; Shahani et al., 2025), and quantization may introduce approximation errors that compromise safety (Zahran et al., 2025). These failures suggest that safety alignment may rely on sharp or overfitted boundaries in parameter or activation space, where small deviations can shift the model from refusal to unsafe compliance. Robust alignment therefore requires not only stronger safety signals but also smoother local safety landscapes.

Existing research on robust safety alignment mainly focuses on three directions: *data-level* approaches curate or reweight fine-tuning data (Liu et al., 2024a; Shen et al., 2025); *objective-level* approaches modify alignment losses to encourage persistent refusal (Qi et al., 2025; Zhao et al., 2025); and *model-level* approaches stabilize safety representations or protect safety-critical layers (Huang et al., 2024; Rosati et al., 2024; Li et al., 2025). While these approaches typically require problem-

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

specific modifications, the role of the ***base optimizer*** in shaping alignment robustness remains largely unexplored. Perturbation-induced failures suggest that robustness also depends on the local geometry of the learned safety boundary, which can be influenced by the optimizer. This motivates us to investigate zeroth-order (ZO) optimization as an optimizer-level mechanism for robust safety alignment. Unlike gradient-based methods, ZO estimates update directions through perturbed function evaluations, which can be viewed as optimizing a locally smoothed objective (Nesterov & Spokoiny, 2017). Although ZO has been studied for memory-efficient LLM fine-tuning (Malladi et al., 2023; Zhang et al., 2024; Liu et al., 2026) and unlearning (Zhang et al., 2025a; Lang et al., 2026), its potential for improving alignment robustness remains underexplored.

In this paper, we strive to improve the robustness of safety alignment through the lens of zeroth-order methods, which have been largely underexplored in this context. Zeroth-order optimization offers a promising route to alignment robustness as it evaluates how safety alignment behaves under perturbations. Our study consists of three main components. ❶ We empirically show that safety-aligned models remain fragile under simple perturbations: lightweight parameter- or activation-level noise can substantially weaken safety behavior. ❷ Although ZO can improve alignment robustness, its slower and less stable convergence limits its effectiveness as a standalone alignment optimizer. We therefore combine FO and ZO in a hybrid framework, using FO to establish safety alignment and ZO to further enhance robustness. Both theoretical and empirical results demonstrate that a few ZO refinement steps suffice to improve robustness with modest computational overhead. ❸ Finally, we observe that different layers exhibit substantially different vulnerability to post-alignment perturbations. Based on this observation, we propose a robustness-sensitive layer selection strategy that targets the most vulnerable layers for ZO refinement.

Our contributions can be summarized as follows:

- We are the first to comprehensively study the robustness of safety alignment against parameter and activation perturbations from the perspective of zeroth-order optimization.
- We propose an FO–ZO hybrid optimization framework that integrates FO and ZO optimizers, combining ZO-induced robustness with FO-driven alignment effectiveness.
- We show theoretically and empirically that a small amount of post-alignment ZO refinement can improve safety robustness with only modest computational overhead.

2 Background and Related Work

2.1 Safety Alignment in Large Language Models

Safety alignment aims to ensure that large language models (LLMs) generate outputs that are consistent with human values and safety constraints (Zhao et al., 2025; Wang et al., 2025; Ji et al., 2025). A standard paradigm is to post-train pretrained models using human feedback, such as Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022). Subsequent work further improves or simplifies this paradigm from different perspectives. Constitutional AI reduces the dependence on direct human feedback by using a set of human-written principles and AI feedback to improve harmlessness (Bai et al., 2022); Safe RLHF explicitly separates helpfulness and harmlessness preferences and formulates safety alignment as a constrained optimization problem (Dai et al., 2024); Direct Preference Optimization (DPO) directly optimizes language models from preference data without explicitly training a reward model or running reinforcement learning (Rafailov et al., 2023); and recent reasoning-based alignment methods improve safety by training models to explicitly reason about safety requirements before generating final answers (Mou et al., 2025).

2.2 Robustness of Safety Alignment

Prior work has identified several ways to compromise the safety behavior of aligned models. At the input level, jailbreak and adversarial prompting attacks can bypass safety guardrails by rewriting, obfuscating, or optimizing harmful instructions (Wei et al., 2023; Zou et al., 2023; Shen et al., 2024). At the model level, safety behavior can be weakened by malicious or even benign fine-tuning (Qi et al., 2024; Yang et al., 2023), persistent backdoors (Hubinger et al., 2024), or interventions on refusal-related representations (Arditi et al., 2024). These results suggest that safety alignment may

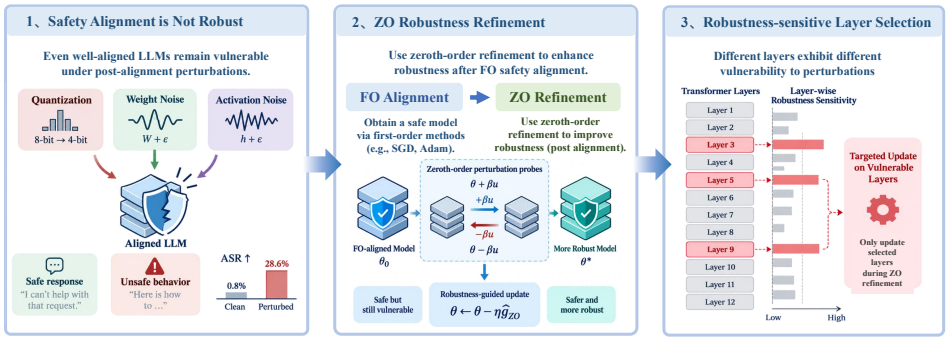


Figure 1: Overview of our robustness-oriented safety alignment framework. We first show that simple post-alignment perturbations can substantially increase ASR in safety-aligned LLMs. We then apply ZO refinement after FO safety alignment and target robustness-critical layers identified by layer-wise sensitivity scores for more efficient robustness improvement.

rely on mechanisms that are effective under normal conditions, yet remain fragile under adversarial inputs, distribution shifts, or post-training modifications.

While many of these failures involve stronger attack capabilities or carefully engineered procedures, safety alignment can also be vulnerable to much simpler changes. Even generic perturbations, arising from quantization, parameter noise, activation noise, hardware faults, or other post-alignment modifications, may be sufficient to degrade the model's safety behavior. Quantization reduces numerical precision to improve efficiency (Dettmers et al., 2022), but the induced approximation errors may alter model behavior; recent work further shows that quantization-related fault injection can weaken safety alignment (Zahran et al., 2025). Similarly, weight or activation perturbations can disturb internal representations and expose safety failures that are not visible under standard inference (Tice et al., 2025; Clymer et al., 2024; Shahani et al., 2025).

2.3 Zeroth-Order Optimization

Zeroth-order (ZO) optimization decreases memory consumption by constructing gradient estimates by evaluating the objective at perturbed points in the parameter space, without requiring backpropagation (Nesterov & Spokoiny, 2017). Beyond its computational advantage, it offers a promising route to enhancing robustness by evaluating how the model behaves under perturbations. This intuition is further supported by a curvature-based connection: ZO optimization has been shown to favor flatter minima (Zhang et al., 2025b), which contribute to improved local robustness against small perturbations (Walter et al., 2025). This perspective has motivated prior work on using ZO optimization to train and adapt large models under constrained or noisy conditions (Liu et al., 2020; 2026). More recently, zeroth-order methods have also been explored in alignment-related settings. Galatolo et al. (Galatolo et al., 2025) explore zeroth-order methods for LLM preference optimization, aiming to reduce memory usage and iteration time. In this work, we instead investigate zeroth-order optimization from the perspective of safety alignment robustness and show that it can enhance the resilience of LLM safety alignment against parameter and activation tampering.

3 Preliminaries

Zeroth-order gradient estimation approximates gradients from function evaluations at perturbed parameter points, without requiring explicit gradient information.

Definition 3.1 (Zeroth-order gradient estimation (Spall, 1992))**.** *Given model parameters $\boldsymbol{\theta} \in \mathcal{R}^d$ and loss function f, the zeroth-order gradient on batch ξ is estimated as*

$$\nabla f_\beta(\boldsymbol{\theta}, \xi) = \frac{1}{2\beta} \left(f(\boldsymbol{\theta} + \beta \mathbf{v}, \xi) - f(\boldsymbol{\theta} - \beta \mathbf{v}, \xi) \right) \mathbf{v}, \tag{1}$$

where $\mathbf{v} \sim N(0, \boldsymbol{I}_d)$ and β is the ZO scale parameter.

Algorithm 1 Sensitivity-Guided ZO Robustness Refinement

Require: Parameters θ_0, loss f, stepsizes $\eta_{\mathrm{fo}}, \eta_{\mathrm{zo}}$, ZO scale β, steps $T_{\mathrm{fo}}, T_{\mathrm{zo}}$, selected layers m.
Ensure: Refined parameters θ.

Stage I: FO safety alignment ($\theta \leftarrow \theta_0$)
for $t = 0, \ldots, T_{\mathrm{fo}} - 1$ **do**
  $\theta \leftarrow \theta - \eta_{\mathrm{fo}} \nabla_\theta f(\theta; \xi_t)$
end for
Stage II: Robustness-sensitive layer selection
$\theta^{\mathrm{fo}} \leftarrow \theta$; compute $\{S(\ell)\}_{\ell=1}^{L}$
$\mathcal{S}_{\mathrm{rob}} \leftarrow \mathrm{Top}\text{-}m\{S(\ell)\}_{\ell=1}^{L}$
Stage III: Targeted ZO refinement
for $t = 0, \ldots, T_{\mathrm{zo}} - 1$ **do**
  Sample $v_t \sim \mathcal{N}(0, I_d)$; mask $\widetilde{v}_t^{(\ell)} = v_t^{(\ell)} \mathbf{1}\{\ell \in \mathcal{S}_{\mathrm{rob}}\}$
  $a_t \leftarrow \frac{d_{\mathcal{S}}}{2\beta}\big[f(\theta + \beta\widetilde{v}_t; \xi_t) - f(\theta - \beta\widetilde{v}_t; \xi_t)\big]$
  $\theta \leftarrow \theta - \eta_{\mathrm{zo}} a_t \widetilde{v}_t$
end for
return θ

4 METHODOLOGY

In this section, we present our framework for improving the robustness of safety alignment, as illustrated in Figure 1. Motivated by the fragility of aligned models under simple perturbations, we use the perturbation-based evaluations inherent to ZO optimization to identify robustness-critical layers and then concentrate ZO refinement on these layers.

4.1 ZEROTH-ORDER ROBUSTNESS REFINEMENT

We propose a hybrid FO-to-ZO framework that retains FO for safety alignment and applies targeted ZO refinement to enhance the robustness of the aligned model, rather than replacing FO with ZO.

Algorithmic framework. Algorithm 1 summarizes our three-stage pipeline. ① We first perform FO safety alignment from θ_0 to obtain an aligned model θ^{fo}. ② We then compute layer-wise robustness sensitivity scores $S(\ell)$ and select the top-m layers to form $\mathcal{S}_{\mathrm{rob}}$, as detailed in Section 4.2. ③ Starting from θ^{fo}, we perform targeted ZO refinement by masking perturbation directions outside $\mathcal{S}_{\mathrm{rob}}$. We next provide a theoretical analysis addressing two fundamental questions about this design.

Question 1: Why ZO after FO alignment? ZO offers a natural way to enhance robustness by optimizing a smoothed objective that accounts for local parameter perturbations. However, its finite-difference estimator introduces additional dimension- and smoothing-dependent stepsize constraints, potentially slowing convergence compared with FO. We therefore adopt a hybrid strategy that uses FO to efficiently establish safety alignment and reserves ZO for a short post-alignment robustness refinement stage. The following theorem characterizes the convergence of ZO on the smoothed objective f_β and highlights the stepsize restrictions underlying this design.

Assumption 4.1 (Local low-rank curvature upper bound)**.** *Let $f_\beta : \mathbb{R}^d \to \mathbb{R}$ denote the smoothed objective. For every iterate θ_t, there exists a positive semidefinite matrix H_t such that, for all θ on the segment between θ_t and θ_{t+1},*

$$\nabla^2 f_\beta(\theta) \preceq H_t,$$

with $\|H_t\|_{\mathrm{op}} \le \ell$ and effective rank $\mathrm{erank}(H_t) \le r$.

Theorem 4.2 (Convergence of ZO optimization)**.** *Suppose Assumption 4.1 holds and the loss function $f(\theta)$ is L-Lipschitz and satisfies the μ-PL condition, by setting $\eta \le \min\left\{\frac{1}{\ell r}, \frac{\mu\varepsilon}{64\,\ell r\, d\,\beta^2 L^4}\right\}$. To reach $\mathbb{E}\big[f_\beta(\theta_t) - f_\beta^\star\big] \le \varepsilon$, it suffices to take*

$$t \ge \frac{1}{\mu\eta} \log \frac{2\big(f_\beta(\theta_0) - f_\beta^\star\big)}{\varepsilon}.$$

Remark 4.3 (Why ZO refinement after FO alignment?)**.** *The FO and ZO convergence bounds have the same iteration form, but they use different stepsizes. For FO optimization, it suffices to take*

$$\eta_{\mathrm{fo}} < \frac{1}{\ell r}, \qquad t_{\mathrm{fo}} \geq \frac{1}{\mu \eta_{\mathrm{fo}}} \log \frac{2\big(f(\theta_0) - f^\star\big)}{\varepsilon}.$$

For ZO optimization, the stepsize must satisfy

$$\eta_{\mathrm{zo}} \leq \min\left\{\frac{1}{\ell r}, \frac{\mu\varepsilon}{64\,\ell r\, d\,\beta^2 L^4}\right\}, \qquad t_{\mathrm{zo}} \geq \frac{1}{\mu \eta_{\mathrm{zo}}} \log \frac{2\big(f_\beta(\theta_0) - f_\beta^\star\big)}{\varepsilon}.$$

The second constraint on η_{zo} comes from the variance and smoothing bias of the ZO estimator. As a result, ZO usually requires more iterations than FO to reach the same optimization accuracy. This motivates our design: FO optimization is used to efficiently obtain a safety-aligned model, while ZO is applied only after FO alignment as a robustness refinement stage.

Question 2: How much does ZO refinement improve robustness against weight perturbations? The preceding analysis highlights the convergence cost of ZO, motivating its use only after FO alignment. We now quantify the robustness benefit of this additional refinement under weight perturbations. Let θ^{fo} denote the FO-aligned model. While FO alignment optimizes the nominal safety loss, ZO refinement targets the smoothed objective $f_\beta(\theta) = \mathbb{E}_v[f(\theta + \beta v)]$, where $v \sim \mathcal{N}(0, I_d)$, which captures the expected safety loss under Gaussian weight perturbations. The following theorem establishes that, starting from θ^{fo}, sufficiently many ZO refinement steps can reduce this expected perturbed loss by at least ϵ.

Theorem 4.4 (ZO refinement improves perturbation robustness)**.** *Suppose that $f_\beta(\theta)$ is h-smooth and satisfies the μ-PL condition. Let θ^{fo} be the FO-aligned model, define $\Delta_0 = f_\beta(\theta^{\mathrm{fo}}) - f_\beta^\star$ and $\sigma^2 = 64 d\beta^2 L^4$. Starting from θ^{fo}, after T steps of ZO refinement, that satisfies $T \geq \frac{1}{\mu\eta_{\mathrm{zo}}} \log \frac{\Delta_0}{\Delta_0/2 - \epsilon}$, and by setting $\eta_{\mathrm{zo}} \leq \min\left\{\frac{1}{h}, \frac{\mu\Delta_0}{h\sigma^2}\right\}$, we have*

$$\mathbb{E}_v[f(\theta_T^{\mathrm{zo}} + \beta v)] \leq \mathbb{E}_v[f(\theta^{\mathrm{fo}} + \beta v)] - \epsilon.$$

Theorem 4.4 formalizes the robustness improvement of our FO-to-ZO framework. FO alignment improves the model's overall safety behavior, while ZO refinement further reduces the loss under local parameter perturbations by optimizing the neighborhood-averaged objective. Therefore, ZO refinement enhances robustness without retraining the model from scratch.

4.2 Robustness-Sensitive Layer Selection

Post-alignment robustness is not uniformly distributed across model layers. Different layers exhibit markedly different vulnerability to parameter perturbations and quantization, suggesting that only a subset of layers may dominate the degradation of safety robustness. As illustrated in Figure 2, the layer-wise robustness sensitivity varies substantially across the network and exhibits a pattern distinct from conventional pruning scores such as SNIP (Lee et al., 2019) and WANDA (Sun et al., 2024). This discrepancy indicates that conventional pruning importance does not necessarily capture the layers most vulnerable to post-alignment perturbations.

Motivated by this observation, we prioritize perturbation-vulnerable layers for targeted ZO refinement instead of updating the entire model. Specifically, we identify layers whose perturbation causes substantial safety degradation, targeting robustness bottlenecks while avoiding unnecessary updates to stable layers. To this end, we define a layer-wise robustness sensitivity score $S(\ell)$ that combines two complementary perturbation effects:

$$S(\ell) = S_{\mathrm{noise}}(\ell) + \lambda S_{\mathrm{quant}}(\ell),$$

where λ balances the contribution of quantization-based sensitivity.

The first term captures sensitivity to random local parameter perturbations, which matches the perturbation-based evaluation principle of ZO optimization. The second term captures sensitivity to structured deployment perturbations such as quantization. Together, $S(\ell)$ measures how much safety alignment is weakened when layer ℓ is locally or structurally perturbed.

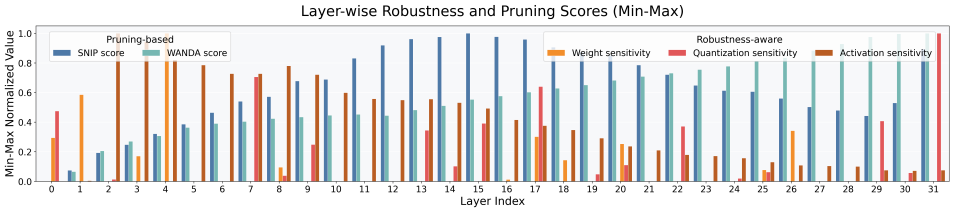


Figure 2: Layer-wise comparison between alignment-oriented pruning scores and robustness-related sensitivity scores. All scores are min–max normalized for visualization. SNIP (Lee et al., 2019) and WANDA (Sun et al., 2024) exhibit different layer-wise patterns from perturbation- and quantization-based robustness sensitivity.

Random perturbation sensitivity. For the random perturbation term, we perturb only layer ℓ while keeping all other layers fixed:

$$\theta^{(\ell,\text{noise})} = \big(\theta_1, \ldots, \theta_{\ell-1}, \theta_\ell + \rho v_\ell, \theta_{\ell+1}, \ldots, \theta_L\big), \qquad v_\ell \sim \mathcal{N}(0, I),$$

where ρ is the perturbation scale. The corresponding sensitivity is defined as

$$S_{\text{noise}}(\ell) = f\big(\theta^{(\ell,\text{noise})}; \mathcal{D}_{\text{safe}}\big) - f\big(\theta; \mathcal{D}_{\text{safe}}\big).$$

This term measures how strongly safety alignment is weakened by a ZO-like local perturbation applied only to layer ℓ.

Quantization sensitivity. We further include a quantization-based term to account for structured perturbations that commonly arise in practical deployment. Let $Q(\cdot)$ denote the quantization operator. For each layer ℓ, we quantize only this layer:

$$\theta^{(\ell,\text{quant})} = \big(\theta_1, \ldots, \theta_{\ell-1}, Q(\theta_\ell), \theta_{\ell+1}, \ldots, \theta_L\big).$$

The quantization sensitivity is defined as

$$S_{\text{quant}}(\ell) = f\big(\theta^{(\ell,\text{quant})}; \mathcal{D}_{\text{safe}}\big) - f\big(\theta; \mathcal{D}_{\text{safe}}\big).$$

This term measures how much safety alignment is weakened when layer ℓ is exposed to quantization. After computing $S(\ell)$ for all layers, we select the layers with the largest scores:

$$\mathcal{S}_{\text{rob}} = \text{Top-}m\{S(\ell)\}_{\ell=1}^{L}.$$

The selected set $\mathcal{S}_{\text{rob}}$ is then used as the update target in the targeted ZO refinement stage. At each refinement step, the random ZO direction is masked outside $\mathcal{S}_{\text{rob}}$, so that only robustness-critical layers are perturbed and updated. In this way, parameter selection and ZO optimization are coupled through the same perturbation-based principle: perturbations are first used to locate vulnerable layers, and ZO updates are then concentrated on these layers to improve robustness efficiently.

5 Experiments

5.1 Experiment Setups

Models and Datasets. We conduct experiments on Llama-3-8B-Instruct (Grattafiori et al., 2024) and Qwen2-7B-Instruct (Qwen Team, 2025), using them as the base safety-aligned models for fine-tuning and robustness refinement. We use CB-Safety (Zou et al., 2024) for both safety alignment and robustness refinement.

Evaluation Metrics. We evaluate model safety and utility across multiple benchmarks. For safety, we use six harmful instruction datasets: AdvBench (Zou et al., 2023), MaliciousInstruct (Huang et al., 2023), TDC2023 (Mazeika et al., 2023; 2024), HarmBench (Mazeika et al., 2024), JailbreakBench (Chao et al., 2024), and StrongREJECT (Souly et al., 2024). We measure attack success rate (ASR) using three evaluators: the official HarmBench-Llama-2-13B-cls classifier, Llama Guard 3-8B, and an AdvBench-style refusal-prefix heuristic (Zou et al., 2023). For weight and activation noise, we average the evaluation results over three random seeds (42, 43, and 44).

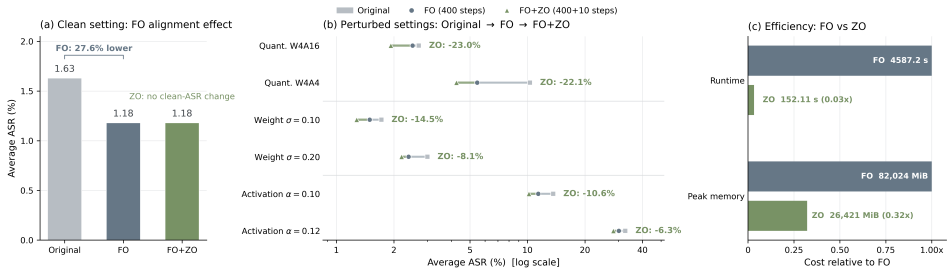


Figure 3: Complementary roles and efficiency of FO alignment and ZO refinement. FO improves clean safety, whereas the 10-step ZO stage further reduces perturbed ASR using only 0.03× the runtime and 0.32× the peak memory of 400-step FO.

Table 1: Safety robustness and utility of Llama-3-8B-Instruct under different perturbations. Arrows indicate changes relative to the unperturbed original model. Higher PPL and ASR indicate worse performance, while lower lm-eval indicates worse utility.

Perturbation	Level	PPL ↓	lm-eval ↑	HarmBench ASR ↓	LlamaGuard3 ASR ↓	AdvBench ASR ↓
Original	–	8.2841	0.6776	0.0158	0.0098	0.0233
Quantization	W4A16	8.9737 (↑0.6896)	0.6434 (↓0.0342)	0.0270 (↑0.0112)	0.0195 (↑0.0097)	0.0345 (↑0.0112)
Quantization	W4A4	14.3443 (↑6.0602)	0.5483 (↓0.1293)	0.0848 (↑0.0690)	0.1005 (↑0.0907)	0.1238 (↑0.1005)
Weight noise	$\sigma = 0.10$	8.8638 (↑0.5797)	0.6640 (↓0.0136)	0.0150 (↓0.0008)	0.0098 (→0.0000)	0.0268 (↑0.0035)
Weight noise	$\sigma = 0.15$	10.0229 (↑1.7388)	0.6461 (↓0.0315)	0.0173 (↑0.0015)	0.0130 (↑0.0033)	0.0288 (↑0.0055)
Weight noise	$\sigma = 0.20$	13.6438 (↑5.3597)	0.5911 (↓0.0865)	0.0238 (↑0.0080)	0.0240 (↑0.0142)	0.0420 (↑0.0187)
Weight noise	$\sigma = 0.30$	257.6617 (↑249.3777)	0.3313 (↓0.3463)	0.0393 (↑0.0235)	0.8977 (↑0.8880)	0.9385 (↑0.9152)
Activation noise	$\alpha = 0.05$	9.5895 (↑1.3054)	0.6490 (↓0.0286)	0.0228 (↑0.0070)	0.0140 (↑0.0042)	0.0338 (↑0.0105)
Activation noise	$\alpha = 0.08$	12.3811 (↑4.0970)	0.6096 (↓0.0680)	0.0438 (↑0.0280)	0.0420 (↑0.0322)	0.0778 (↑0.0545)
Activation noise	$\alpha = 0.10$	16.0394 (↑7.7553)	0.5673 (↓0.1103)	0.0730 (↑0.0572)	0.1655 (↑0.1557)	0.1705 (↑0.1472)
Activation noise	$\alpha = 0.12$	22.5919 (↑14.3078)	0.5166 (↓0.1610)	0.0930 (↑0.0772)	0.5179 (↑0.5081)	0.3641 (↑0.3408)

Perturbation Settings. We consider three simple post-alignment perturbations. **Quantization** reduces numerical precision during deployment, where W4A16 denotes 4-bit weight quantization with 16-bit activations, and W4A4 denotes 4-bit weights with 4-bit activations. **Parameter noise** adds absolute Gaussian noise to all model parameters, controlled by the noise scale σ. **Activation noise** injects Gaussian noise into hidden states, with the noise magnitude scaled by the standard deviation of the corresponding activations and controlled by the relative scale α.

5.2 Safety Alignment Remains Fragile under Simple Perturbations

Table 1 shows that Llama-3-8B-Instruct remains fragile under simple post-alignment perturbations. Both parameter-level perturbations, including quantization and weight noise, and activation-level noise consistently increase ASR. The degradation becomes more severe under stronger perturbations, such as W4A4 quantization or larger activation noise. These results suggest that safety alignment is effective in the standard setting but remains vulnerable to lightweight parameter- and activation-level perturbations. Additional results on Qwen2-7B-Instruct are reported in Appendix C.

5.3 FO Safety Alignment Improves Safety, while ZO Refinement Enhances Robustness

Figure 3 illustrates the complementary roles of FO alignment and ZO refinement. The 400-step FO stage reduces clean average ASR by 27.6%, establishing the model's overall safety behavior. The subsequent 10-step layer-robustness-aware ZO stage preserves clean safety while further reducing ASR by 6.3%–23.0% across quantization, weight noise, and activation noise settings. Despite using only $1/40$ as many training steps, ZO requires just 0.03× the runtime and 0.32× the peak GPU memory of FO. These results show that FO primarily establishes safety alignment, whereas ZO efficiently strengthens robustness against post-alignment perturbations. Consistent results on Qwen2-7B-Instruct are provided in Appendix C.

Table 2: Average ASR (%) across HarmBench, LlamaGuard3, and AdvBench under different perturbations. Arrows show absolute percentage-point changes from the FO baseline with the same optimizer; ↓ and ↑ indicate improved and degraded robustness, respectively. P denotes pruning, and bold marks the lowest comparable ASR.

Pipeline	Quantization		Weight noise			Activation noise			
	W4A16	W4A4	$\sigma = 0.10$	$\sigma = 0.15$	$\sigma = 0.20$	$\alpha = 0.05$	$\alpha = 0.08$	$\alpha = 0.10$	$\alpha = 0.12$
Optimizer: SGD									
FO	0.0230	0.0545	0.0149	0.0151	0.0238	0.0218	0.0434	0.1139	0.3008
FO → ZO	**0.0193** (↓ 0.0037)	0.0425 (↓ 0.0120)	0.0128 (↓ 0.0021)	0.0153 (↑ 0.0002)	**0.0219** (↓ 0.0019)	**0.0155** (↓ 0.0063)	0.0412 (↓ 0.0022)	**0.1018** (↓ 0.0121)	**0.2818** (↓ 0.0190)
FO → ZO + P	0.0198 (↓ 0.0032)	**0.0415** (↓ 0.0130)	**0.0118** (↓ 0.0031)	**0.0133** (↓ 0.0018)	0.0223 (↓ 0.0015)	0.0175 (↓ 0.0043)	**0.0357** (↓ 0.0077)	0.1094 (↓ 0.0045)	0.2907 (↓ 0.0101)
Optimizer: AdamW									
FO	0.0105	0.0285	0.0080	0.0089	0.0123	0.0095	0.0127	0.0207	0.0585
FO → ZO	0.0090 (↓ 0.0015)	0.0178 (↓ 0.0107)	0.0077 (↓ 0.0003)	0.0085 (↓ 0.0004)	0.0108 (↓ 0.0015)	0.0073 (↓ 0.0022)	0.0121 (↓ 0.0006)	0.0213 (↑ 0.0006)	0.0488 (↓ 0.0097)
FO → ZO + P	**0.0068** (↓ 0.0037)	**0.0108** (↓ 0.0177)	**0.0057** (↓ 0.0023)	**0.0069** (↓ 0.0020)	**0.0097** (↓ 0.0026)	**0.0068** (↓ 0.0027)	**0.0096** (↓ 0.0031)	**0.0187** (↓ 0.0020)	**0.0479** (↓ 0.0106)

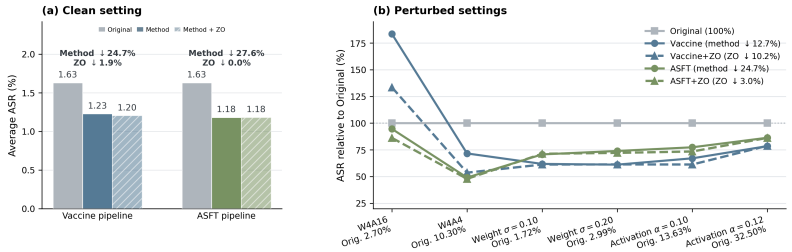


Figure 4: Effects of Vaccine, ASFT, and subsequent ZO refinement on clean and perturbed ASR. Panel (a) reports clean ASR, while Panel (b) reports ASR under six perturbations. Arrows denote relative reductions from robust alignment and ZO refinement, respectively. Lower is better.

5.4 ZO Refinement Improves Robustness Across FO Optimizers

Table 2 evaluates the robustness gains of ZO refinement across SGD- and AdamW-based FO alignment. ZO alone generally lowers the average ASR, with particularly consistent improvements under quantization, although small degradations remain in several noise settings. Incorporating pruning further improves robustness and achieves the lowest ASR in every evaluated setting for both optimizers. Relative to the corresponding FO baselines, the complete FO → ZO + P pipeline reduces ASR by up to 0.013 with SGD and 0.0265 with AdamW. These results indicate that the proposed ZO refinement and pruning strategy is complementary to FO safety alignment and generalizes across different FO optimizers.

5.5 ZO Refinement Further Enhances Existing Robust Alignment Methods

As shown in Figure 4, both Vaccine and ASFT improve safety under clean and perturbed settings. Under the clean setting, Vaccine and ASFT reduce ASR by 24.7% and 27.6%, respectively, while the subsequent ZO refinement introduces only limited additional reductions of 1.9% and 0.0%. In contrast, ZO provides more noticeable improvements under perturbations. Averaged across the six perturbation settings, Vaccine and ASFT reduce ASR by 12.7% and 24.7%, respectively, while ZO further reduces their perturbed ASR by 10.2% and 3.0%. These results suggest that robust safety alignment establishes the primary safety improvement, whereas ZO refinement mainly strengthens robustness against post-alignment perturbations, particularly when applied after Vaccine.

5.6 Efficiency Analysis: Training Time and Storage Overhead

Table 3: Runtime and GPU memory comparison.

Metric	FO	ZO
Runtime	4587.2 s	152.11 s (**0.03**×)
Peak memory	82024 MiB	26421 MiB (**0.32**×)

Table 3 compares the cost of the FO alignment stage and the subsequent ZO robustness refinement stage. FO training is used to establish safety alignment, while ZO refinement is applied afterward to improve robustness. The ZO stage adds only 152.11 seconds on top of the 1146.8-second FO alignment stage, increasing the total training time by about 13.3%. It also uses only 0.32× the

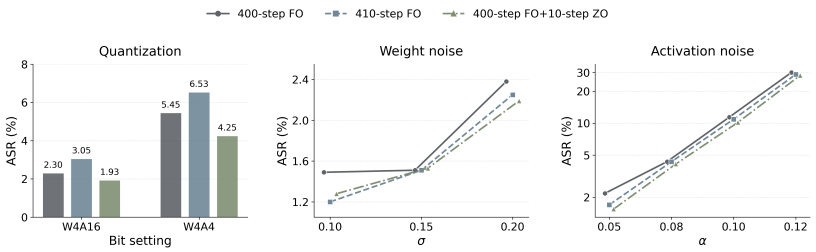


Figure 5: Effect of replacing late-stage FO with ZO refinement. We compare 100-step FO, 110-step FO, and 100-step FO followed by 10-step ZO using average ASR across three benchmarks. Replacing the final 10 FO steps with ZO lowers ASR across perturbations while reducing memory use.

peak GPU memory of FO training. These results indicate that ZO refinement provides a lightweight post-alignment robustness enhancement with modest additional cost.

5.7 Ablation Study

Replacing Late-Stage FO with ZO Improves Robustness and Reduces Memory. We compare 100-step FO, 110-step FO, and 100-step FO followed by 10-step ZO to distinguish the effect of ZO refinement from that of additional training. As shown in Figure 5, FO+ZO achieves lower average ASR than continued FO under quantization, weight noise, and activation noise, despite using the same total number of steps. By avoiding backpropagation during refinement, ZO also reduces GPU memory usage, providing a better robustness–efficiency trade-off.

Layer-Wise Robustness Selection Enables Stronger ZO Refinement. Figure 6 compares different parameter-selection strategies for ZO robustness refinement. Robustness-aware selection achieves the lowest HarmBench ASR across all perturbation settings, showing that it can effectively identify layers that are most responsible for safety degradation under perturbations. In contrast, SNIP and WANDA perform well under weight noise but yield higher ASR elsewhere, indicating that the proposed criterion provides a more reliable signal for ZO refinement than pruning-based importance scores.

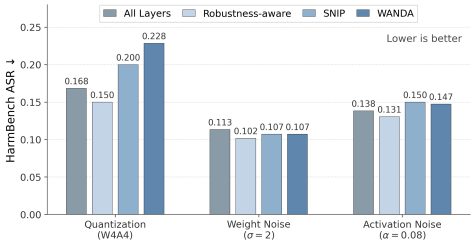


Figure 6: HarmBench ASR after ZO refinement with different layer-selection strategies. Lower ASR is better. Robustness-aware selection is consistently best or comparable across perturbations.

Combined Perturbation Scores Enable More Effective Layer Selection. Figure 7 compares random, source-specific, and combined-score layer selection. Random and source-specific masks yield only modest, inconsistent improvements over the source model. In contrast, the combined mask achieves the lowest mean ASR across all perturbations, with clear gains under W4A16 and W4A4D16 quantization. This suggests that combining perturbation signals identifies layers contributing more broadly to robustness.

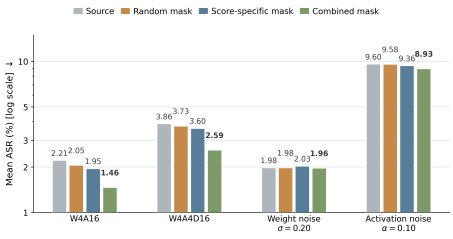


Figure 7: Comparison of layer-selection strategies under quantization, weight and activation noise.

6 Conclusion

In this work, we study the robustness of safety-aligned LLMs under simple post-alignment perturbations, including quantization, weight noise, and activation noise. We theoretically and empirically show that standard safety alignment can remain fragile, and propose a perturbation-guided FO-to-ZO refinement framework where FO establishes safety alignment and ZO further improves robustness. By identifying robustness-critical layers, we restrict ZO updates to these layers for more efficient refinement. Future work includes extending our layer-level selection to finer-grained robustness attribution at the module, neuron, or parameter level.

AI USE STATEMENT

We used generative AI tools to assist with language editing, improving the clarity and organization of the manuscript, and refining LaTeX formatting and visualization code. All AI-assisted content was carefully reviewed and revised by the authors. We independently verified the technical claims, mathematical proofs, experimental results, and references, and checked the correctness of the final figures and tables. We take full responsibility for the final content of this work, including any text or artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We provide implementation details, training configurations, and evaluation protocols in the main paper and appendix to facilitate reproducibility. The appendix includes the complete proofs of our theoretical results, hyperparameter settings, and additional experimental results. All experiments are conducted using publicly available models, datasets, and evaluation benchmarks. Our implementation and scripts for reproducing the experiments are available at`https://anonymous.4open.science/r/Robust-ZO-Safety-Alignment-9271`.

REFERENCES

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

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

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1745.

Joshua Clymer, Caden Juang, and Severin Field. Poser: Unmasking alignment faking llms by manipulating their internals. *arXiv preprint arXiv:2405.05466*, 2024.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*, 2024.

Tim Dettmers et al. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*, 2022.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Alessio Galatolo, Zhenbang Dai, Katie Winkle, and Meriem Beloucif. Visualising policy-reward interplay to inform zeroth-order preference optimisation of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17446–17461, 2025.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings of the association for computational linguistics: EMNLP 2020*, pp. 3356–3369, 2020.

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

Melody Y Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, et al. Deliberative alignment: Reasoning enables safer language models. *SuperIntelligence-Robotics-Safety & Alignment*, 2(3), 2025.

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:74058–74088, 2024.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak of open-source llms via exploiting generation, 2023. URL https://arxiv.org/abs/2310.06987.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M Ziegler, Tim Maxwell, Newton Cheng, et al. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Alex Qiu, Jiayi Zhou, Kaile Wang, Boxun Li, et al. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31983–32016, 2025.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 795–811. Springer, 2016.

Yicheng Lang, Yihua Zhang, Chongyu Fan, Changsheng Wang, Jinghan Jia, and Sijia Liu. Downgrade to upgrade: Optimizer simplification enhances robustness in llm unlearning. *14th International Conference on Learning Representations, ICLR 2026*, 2026.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019.

Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. Safety layers in aligned large language models: The key to llm security. In *The Thirteenth International Conference on Learning Representations*, 2025.

Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

Xiaoqun Liu, Jiacheng Liang, Muchao Ye, and Zhaohan Xi. Robustifying safety-aligned large language models through clean data curation. *arXiv preprint arXiv:2405.19358*, 2024a.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, Kailong Wang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023.

Zhihao Liu, Jian Lou, Wenjie Bao, Yuke Hu, Wenli Wang, Zhan Qin, and Kui Ren. Differentially private zeroth-order methods for scalable large language model fine-tuning. *IEEE Transactions on Information Forensics and Security*, 2026.

Zhuanghua Liu, Cheng Chen, Luo Luo, and Bryan Kian Hsiang Low. Zeroth-order methods for constrained nonconvex nonsmooth stochastic optimization. In *Forty-first International Conference on Machine Learning*, 2024b.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

Mantas Mazeika, Andy Zou, Norman Mu, Long Phan, Zifan Wang, Chunru Yu, Adam Khoja, Fengqing Jiang, Aidan O'Gara, Ellie Sakhaee, Zhen Xiang, Arezoo Rajabi, Dan Hendrycks, Radha Poovendran, Bo Li, and David Forsyth. TDC 2023 (LLM Edition): The Trojan Detection Challenge. In *NeurIPS Competition Track*, 2023. URL https://github.com/centerforaisafety/tdc2023-starter-kit.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *International Conference on Machine Learning*, pp. 35181–35224. PMLR, 2024.

Yutao Mou, Yuxiao Luo, Shikun Zhang, and Wei Ye. Saro: Enhancing llm safety through reasoning-based alignment. *arXiv preprint arXiv:2504.09420*, 2025.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 2017.

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

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *The Thirteenth International Conference on Learning Representations*, 2025.

Alibaba Cloud Qwen Team. Qwen2 technical report. `https://qwen2.org/paper/`, 2025.

Rafael Rafailov et al. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. Representation noising: A defence mechanism against harmful finetuning. *Advances in Neural Information Processing Systems*, 37:12636–12676, 2024.

Prithviraj Singh Shahani, Kaveh Eskandari Miandoab, and Matthias Scheutz. Noise injection systemically degrades large language model safety guardrails. *arXiv preprint arXiv:2505.13500*, 2025.

Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. Seal: Safety-enhanced aligned llm fine-tuning via bilevel data selection. In *International Conference on Learning Representations*, 2025.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 1671–1685, 2024.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A StrongREJECT for empty jailbreaks. In *ICLR 2024 Workshops: R2-FM*, 2024. URL `https://arxiv.org/abs/2402.10260`.

James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In *12th International Conference on Learning Representations, ICLR 2024*, 2024.

Cameron Tice, Philipp Alexander Kreer, Nathan Helm-Burger, Prithviraj Singh Shahani, Fedor Ryzhenkov, Teun van der Weij, Felix Hofstätter, and Jacob Haimes. Sandbag detection through model impairment. In *Workshop on Socially Responsible Language Modelling Research*, 2024.

Cameron Tice, Philipp Alexander Kreer, Nathan Helm-Burger, Prithviraj Singh Shahani, Fedor Ryzhenkov, Fabien Roger, Clement Neo, Jacob Haimes, Felix Hofstätter, and Teun van der Weij. Noise injection reveals hidden capabilities of sandbagging language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Nils Philipp Walter, Linara Adilova, Jilles Vreeken, and Michael Kamp. When flatness does (not) guarantee adversarial robustness. *arXiv preprint arXiv:2510.14231*, 2025.

Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, Miao Yu, Shiqian Zhao, Chenlong Yin, Jinhu Fu, Yibo Yan, Hanjun Luo, et al. A comprehensive survey in llm (-agent) full stack safety: Data, training and deployment. *arXiv preprint arXiv:2504.15585*, 2025.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in neural information processing systems*, 36:80079–80110, 2023.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*, 2023.

Noureldin Zahran, Ahmad Tahmasivand, Ihsen Alouani, Khaled Khasawneh, and Mohammed Fouda. On jailbreaking quantized language models through fault injection attacks. In *Proceedings of the Great Lakes Symposium on VLSI 2025*, pp. 554–561, 2025.

Ci Zhang, Chence Yang, Qitao Tan, Jun Liu, Ao Li, Yanzhi Wang, Jin Lu, Jinhui Wang, and Geng Yuan. Towards memory-efficient and sustainable machine unlearning on edge using zeroth-order optimizer. In *Proceedings of the Great Lakes Symposium on VLSI 2025*, pp. 227–232, 2025a.

Liang Zhang, Bingcong Li, Kiran Koshy Thekumparampil, Sewoong Oh, Michael Muehlebach, and Niao He. Zeroth-order optimization finds flat minima. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D Lee, Wotao Yin, Mingyi Hong, et al. Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: A benchmark. *arXiv preprint arXiv:2402.11592*, 2024.

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

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

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. *Advances in Neural Information Processing Systems*, 37:83345–83373, 2024.

A Hyperparameters and Hardware Configuration

A.1 Training Hyperparameters

The hyperparameters of the reported Llama-3-8B-Instruct safety-alignment and robustness-refinement runs are summarized in Table 4. And the hyperparameters of the reported Qwen2-7B-Instruct safety-alignment and robustness-refinement runs are summarized in Table 5. All runs use a per-device batch size of 1 and gradient accumulation of 8. The SGD checkpoint is taken at step 400 of a run configured for 500 steps.

Table 4: Hyperparameters of the reported FO alignment and ZO refinement runs on Llama-3-8B-Instruct.

Stage	Method	Hyperparameter	Value
General setup			
General	All methods	Training dataset	CB-Safety
General	All methods	Maximum sequence length	1536
General	All methods	Per-device batch size	1
General	All methods	Gradient accumulation steps	8
Stage I: FO safety alignment			
Stage I	SGD	Reported checkpoint	Step 400
Stage I	SGD	Training sample size	4994
Stage I	SGD	Learning rate	7.5×10^{-4}
Stage I	SGD	Momentum	0.90
Stage I	SGD	Scheduler / warmup ratio	`cosine` / 0.05
Stage I	SGD	Weight decay	0
Stage I	AdamW	Reported checkpoint	Step 800
Stage I	AdamW	Training sample size	4994
Stage I	AdamW	Learning rate	1×10^{-5}
Stage I	AdamW	$(\beta_1, \beta_2, \epsilon)$	$(0.9,\ 0.999,\ 10^{-8})$
Stage I	AdamW	Scheduler / warmup ratio	`constant` / 0.03
Stage I	AdamW	Weight decay	0
Stage II: ZO robustness refinement			
Stage II	ZO (after SGD)	Training steps	10
Stage II	ZO (after SGD)	Training sample size	4994
Stage II	ZO (after SGD)	Learning rate	3×10^{-4}
Stage II	ZO (after SGD)	Perturbation scale μ	2×10^{-2}
Stage II	ZO (after SGD)	Directions per update	64
Stage II	ZO (after SGD)	ZO momentum	0.02
Stage II	ZO (after SGD)	Scheduler / warmup ratio	`constant_with_warmup` / 0.05
Stage II	ZO (after SGD)	Weight decay	0
Stage II	AdaZO (after AdamW)	Training steps	50
Stage II	AdaZO (after AdamW)	Training sample size	4994
Stage II	AdaZO (after AdamW)	Learning rate	3×10^{-7}
Stage II	AdaZO (after AdamW)	Perturbation scale μ	5×10^{-3}
Stage II	AdaZO (after AdamW)	Directions per update	16
Stage II	AdaZO (after AdamW)	$(\beta_1, \beta_2, \epsilon)$	$(0.9,\ 0.99,\ 10^{-8})$
Stage II	AdaZO (after AdamW)	Scheduler / warmup ratio	`cosine` / 0.10
Stage II	AdaZO (after AdamW)	Weight decay	0
Stage II	AdaZO (after AdamW)	Second-moment dtype	`bfloat16`

Table 5: Hyperparameters of the reported FO alignment and top-4 ZO refinement runs on Qwen2-7B-Instruct. The score-specific and combined-score masks use the same ZO hyperparameters.

Stage	Method	Hyperparameter	Value
General setup			
General	All methods	Training dataset	CB-Safety
General	All methods	Maximum sequence length	1536
General	All methods	Per-device batch size	1
General	All methods	Gradient accumulation steps	8
Stage I: FO safety alignment			
Stage I	SGD	Reported checkpoint	Step 100
Stage I	SGD	Training sample size	4994
Stage I	SGD	Learning rate	1×10^{-2}
Stage I	SGD	Momentum	0.5
Stage I	SGD	Scheduler / warmup ratio	`constant_with_warmup` / 0.05
Stage I	SGD	Weight decay	0
Stage II: ZO robustness refinement			
Stage II	ZO (after SGD)	Training steps	10
Stage II	ZO (after SGD)	Training sample size	4994
Stage II	ZO (after SGD)	Learning rate	3×10^{-4}
Stage II	ZO (after SGD)	Perturbation scale μ	5×10^{-2}
Stage II	ZO (after SGD)	Directions per update	64
Stage II	ZO (after SGD)	ZO momentum	0.02
Stage II	ZO (after SGD)	Scheduler / warmup ratio	`constant` / 0
Stage II	ZO (after SGD)	Weight decay	0

A.2 Hardware Configuration

The Llama-3-8B-Instruct experiments were run on a local RTX 4090 server. The machine contains 8 NVIDIA GeForce RTX 4090 GPUs with 24GB memory each. The Qwen2-7B-Instruct experiments used RTX 4090 GPUs with 48 GB of memory for full-parameter fine-tuning and RTX 4090 GPUs with 24 GB of memory for the pruning stage.Training uses multi-GPU execution, while quantization and safety evaluation use fewer GPUs when possible.

B THEORETICAL PROOFS

B.1 ASSUMPTIONS

Lipschitz continuity and smoothness control the variation of the objective and its gradient, serving as standard assumptions in FO and ZO convergence analyses (Malladi et al., 2023; Liu et al., 2024b). We also adopt the Polyak–Łojasiewicz (PL) condition, a common gradient-dominance assumption for deriving linear convergence-type guarantees in nonconvex settings (Karimi et al., 2016).

Definition B.1 (Lipschitz continuity)**.** *The function $f(\theta)$ is L-Lipschitz if there exists a constant $L > 0$ such that, for all $\theta, \theta' \in \mathbb{R}^d$ we have*

$$|f(\theta) - f(\theta')| \leq L\|\theta - \theta'\|.$$

Definition B.2 (Smoothness)**.** *Let $f(\theta)$ be differentiable. The function $f(\theta)$ is h-smooth if its gradient is h-Lipschitz for all $\theta, \theta' \in \mathbb{R}^d$, namely,*

$$\|\nabla f(\theta) - \nabla f(\theta')\| \leq h\|\theta - \theta'\|.$$

Definition B.3 (Polyak–Łojasiewicz condition)**.** *The function $f_\beta(\theta)$ satisfies the μ-PL condition if for any $\theta \in \mathcal{R}^d$ we have*

$$\frac{1}{2}\|\nabla f_\beta(\theta)\|^2 \geq \mu\big(f_\beta(\theta) - f_\beta^\star\big).$$

B.2 PROOF OF ZO CONVERGENCE

We first prove the convergence guarantee for ZO optimization on the smoothed objective f_β. Recall the two-sided ZO estimator

$$\widehat{g}(\theta; v) := \frac{d}{2\beta}\big(f(\theta + \beta v) - f(\theta - \beta v)\big)v.$$

We use the following standard variance bound.

Lemma B.4 (Variance bound for the ZO estimator (Liu et al., 2026))**.** *For the estimator $\widehat{g}(\theta; v)$, we have*

$$\mathbb{E}_v[\widehat{g}(\theta; v)] = \nabla f_\beta(\theta),$$

and

$$\mathbb{E}_v\big[\|\widehat{g}(\theta; v) - \nabla f_\beta(\theta)\|^2\big] \leq 64 d\beta^2 L^4.$$

Proof of Theorem 4.2. Let

$$g_t := \nabla f_\beta(\theta_t), \qquad \widehat{g}_t := \widehat{g}(\theta_t; v_t).$$

The ZO update is

$$\theta_{t+1} = \theta_t - \eta\widehat{g}_t.$$

By Taylor's theorem, there exists a point $\widetilde{\theta}_t$ on the line segment between θ_t and θ_{t+1} such that

$$f_\beta(\theta_{t+1}) \leq f_\beta(\theta_t) + g_t^\top(\theta_{t+1} - \theta_t) + \frac{1}{2}(\theta_{t+1} - \theta_t)^\top \nabla^2 f_\beta(\widetilde{\theta}_t)(\theta_{t+1} - \theta_t).$$

Substituting $\theta_{t+1} - \theta_t = -\eta\widehat{g}_t$, we obtain

$$f_\beta(\theta_{t+1}) \leq f_\beta(\theta_t) - \eta g_t^\top \widehat{g}_t + \frac{\eta^2}{2}\widehat{g}_t^\top \nabla^2 f_\beta(\widetilde{\theta}_t)\widehat{g}_t.$$

By Assumption 4.1,

$$\nabla^2 f_\beta(\widetilde{\theta}_t) \preceq H_t.$$

Therefore,

$$f_\beta(\theta_{t+1}) \leq f_\beta(\theta_t) - \eta g_t^\top \widehat{g}_t + \frac{\eta^2}{2}\widehat{g}_t^\top H_t \widehat{g}_t.$$

Taking conditional expectation over the ZO randomness at step t, denoted by $\mathbb{E}_t[\cdot]$, and using Lemma B.4, we have

$$\mathbb{E}_t[\widehat{g}_t] = g_t.$$

Thus,

$$\mathbb{E}_t[f_\beta(\theta_{t+1})] \le f_\beta(\theta_t) - \eta\|g_t\|^2 + \frac{\eta^2}{2}\mathbb{E}_t[\widehat{g}_t^\top H_t \widehat{g}_t].$$

It remains to bound the quadratic term. Since $H_t \succeq 0$,

$$\widehat{g}_t^\top H_t \widehat{g}_t \le \mathrm{tr}(H_t)\|\widehat{g}_t\|^2.$$

Moreover, by the effective-rank condition and $\|H_t\|_{\mathrm{op}} \le \ell$,

$$\mathrm{tr}(H_t) = \mathrm{erank}(H_t)\|H_t\|_{\mathrm{op}} \le r\ell.$$

Hence,

$$\mathbb{E}_t[\widehat{g}_t^\top H_t \widehat{g}_t] \le \ell r\, \mathbb{E}_t[\|\widehat{g}_t\|^2].$$

Using variance decomposition,

$$\begin{aligned}\mathbb{E}_t[\|\widehat{g}_t\|^2] &= \|\mathbb{E}_t[\widehat{g}_t]\|^2 + \mathbb{E}_t[\|\widehat{g}_t - \mathbb{E}_t[\widehat{g}_t]\|^2] \\ &\le \|g_t\|^2 + 64d\beta^2 L^4.\end{aligned}$$

Substituting this bound yields

$$\begin{aligned}\mathbb{E}_t[f_\beta(\theta_{t+1})] \le f_\beta(\theta_t) &- \eta\|g_t\|^2 \\ &+ \frac{\eta^2\ell r}{2}\big(\|g_t\|^2 + 64d\beta^2 L^4\big).\end{aligned}$$

Equivalently,

$$\mathbb{E}_t[f_\beta(\theta_{t+1})] \le f_\beta(\theta_t) - \left(\eta - \frac{\eta^2\ell r}{2}\right)\|g_t\|^2 + 32\eta^2\ell r d\beta^2 L^4.$$

If $0 < \eta \le 1/(\ell r)$, then

$$\eta - \frac{\eta^2\ell r}{2} \ge \frac{\eta}{2}.$$

Therefore,

$$\mathbb{E}_t[f_\beta(\theta_{t+1})] \le f_\beta(\theta_t) - \frac{\eta}{2}\|g_t\|^2 + 32\eta^2\ell r d\beta^2 L^4.$$

By the μ-PL condition on f_β,

$$\frac{1}{2}\|g_t\|^2 \ge \mu\big(f_\beta(\theta_t) - f_\beta^\star\big).$$

Thus,

$$\mathbb{E}_t[f_\beta(\theta_{t+1}) - f_\beta^\star] \le (1-\mu\eta)\big(f_\beta(\theta_t) - f_\beta^\star\big) + 32\eta^2\ell r d\beta^2 L^4.$$

Taking total expectation and defining

$$\Delta_t := \mathbb{E}[f_\beta(\theta_t) - f_\beta^\star],$$

we obtain the recursion

$$\Delta_{t+1} \le (1-\mu\eta)\Delta_t + 32\eta^2\ell r d\beta^2 L^4.$$

Unrolling gives

$$\begin{aligned}\Delta_t &\le (1-\mu\eta)^t\Delta_0 + 32\eta^2\ell r d\beta^2 L^4 \sum_{i=0}^{t-1}(1-\mu\eta)^i \\ &\le (1-\mu\eta)^t\Delta_0 + \frac{32\eta\ell r d\beta^2 L^4}{\mu}.\end{aligned}$$

Using $1 - x \le e^{-x}$, we conclude that

$$\Delta_t \le e^{-\mu\eta t}\Delta_0 + \frac{32\eta\ell r d\beta^2 L^4}{\mu}.$$

It remains to choose η and t so that $\Delta_t \le \varepsilon$. If

$$\eta \le \frac{\mu\varepsilon}{64\ell r d\beta^2 L^4},$$

then the bias term satisfies

$$\frac{32\eta\ell r d\beta^2 L^4}{\mu} \le \frac{\varepsilon}{2}.$$

If additionally

$$t \ge \frac{1}{\mu\eta}\log\frac{2\Delta_0}{\varepsilon},$$

then

$$e^{-\mu\eta t}\Delta_0 \le \frac{\varepsilon}{2}.$$

Combining the two bounds yields

$$\mathbb{E}\big[f_\beta(\theta_t) - f_\beta^\star\big] = \Delta_t \le \varepsilon.$$

This proves the theorem. □

B.3 Proof of Robustness Improvement

We next prove that ZO refinement can reduce the perturbation robustness gap around an FO-aligned solution.

Proof of Theorem 4.4. Define

$$\Delta_t := \mathbb{E}[f_\beta(\theta_t^{\mathrm{zo}}) - f_\beta^\star].$$

Since f_β is h-smooth, the ZO update

$$\theta_{t+1}^{\mathrm{zo}} = \theta_t^{\mathrm{zo}} - \eta_{\mathrm{zo}}\widehat{g}_t$$

gives

$$f_\beta(\theta_{t+1}^{\mathrm{zo}}) \le f_\beta(\theta_t^{\mathrm{zo}}) - \eta_{\mathrm{zo}}\nabla f_\beta(\theta_t^{\mathrm{zo}})^\top\widehat{g}_t + \frac{h\eta_{\mathrm{zo}}^2}{2}\|\widehat{g}_t\|^2.$$

Taking conditional expectation over the randomness of the ZO estimator and using

$$\mathbb{E}_t[\widehat{g}_t] = \nabla f_\beta(\theta_t^{\mathrm{zo}}),$$

we obtain

$$\mathbb{E}_t[f_\beta(\theta_{t+1}^{\mathrm{zo}})] \le f_\beta(\theta_t^{\mathrm{zo}}) - \eta_{\mathrm{zo}}\|\nabla f_\beta(\theta_t^{\mathrm{zo}})\|^2 + \frac{h\eta_{\mathrm{zo}}^2}{2}\mathbb{E}_t[\|\widehat{g}_t\|^2].$$

By the variance bound of the two-sided ZO estimator,

$$\mathbb{E}_t[\|\widehat{g}_t\|^2] \le \|\nabla f_\beta(\theta_t^{\mathrm{zo}})\|^2 + \sigma^2.$$

Therefore,

$$\mathbb{E}_t[f_\beta(\theta_{t+1}^{\mathrm{zo}})] \le f_\beta(\theta_t^{\mathrm{zo}}) - \left(\eta_{\mathrm{zo}} - \frac{h\eta_{\mathrm{zo}}^2}{2}\right)\|\nabla f_\beta(\theta_t^{\mathrm{zo}})\|^2 + \frac{h\eta_{\mathrm{zo}}^2}{2}\sigma^2.$$

Since

$$\eta_{\mathrm{zo}} \le \frac{1}{h},$$

we have

$$\eta_{\mathrm{zo}} - \frac{h\eta_{\mathrm{zo}}^2}{2} \ge \frac{\eta_{\mathrm{zo}}}{2}.$$

Using the μ-PL condition,

$$\frac{1}{2}\|\nabla f_\beta(\theta_t^{\mathrm{zo}})\|^2 \ge \mu(f_\beta(\theta_t^{\mathrm{zo}}) - f_\beta^\star),$$

we obtain

$$\Delta_{t+1} \leq (1 - \mu\eta_{\mathrm{zo}})\Delta_t + \frac{h\eta_{\mathrm{zo}}^2}{2}\sigma^2.$$

Unrolling the recursion gives

$$\Delta_T \leq (1 - \mu\eta_{\mathrm{zo}})^T \Delta_0 + \frac{h\eta_{\mathrm{zo}}\sigma^2}{2\mu}.$$

By the choice of stepsize,

$$\eta_{\mathrm{zo}} \leq \frac{\mu\Delta_0}{h\sigma^2},$$

we have

$$\frac{h\eta_{\mathrm{zo}}\sigma^2}{2\mu} \leq \frac{\Delta_0}{2}.$$

Moreover, the condition on T implies

$$(1 - \mu\eta_{\mathrm{zo}})^T \leq \frac{\Delta_0 - \epsilon - \frac{h\eta_{\mathrm{zo}}\sigma^2}{2\mu}}{\Delta_0}.$$

Therefore,

$$\Delta_T \leq \Delta_0 - \epsilon - \frac{h\eta_{\mathrm{zo}}\sigma^2}{2\mu} + \frac{h\eta_{\mathrm{zo}}\sigma^2}{2\mu} = \Delta_0 - \epsilon.$$

Recalling that

$$\Delta_0 = f_\beta(\theta^{\mathrm{fo}}) - f_\beta^\star,$$

we have

$$\mathbb{E}[f_\beta(\theta_T^{\mathrm{zo}}) - f_\beta^\star] \leq f_\beta(\theta^{\mathrm{fo}}) - f_\beta^\star - \epsilon.$$

Hence,

$$\mathbb{E}[f_\beta(\theta_T^{\mathrm{zo}})] \leq f_\beta(\theta^{\mathrm{fo}}) - \epsilon.$$

Finally, when $\rho = \beta$,

$$f_\beta(\theta) = \mathbb{E}_v[f(\theta + \rho v)],$$

which directly gives

$$\mathbb{E}_v[f(\theta_T^{\mathrm{zo}} + \rho v)] \leq \mathbb{E}_v[f(\theta^{\mathrm{fo}} + \rho v)] - \epsilon.$$

□

C Additional Experimental Results

Table 6: Safety robustness and utility comparison between 400-step SGD and the subsequent 10-step ZO robustness refinement on Llama3-8B-Instruct. For Stage I, arrows indicate changes relative to the original model under the corresponding perturbation; for Stage II, arrows indicate changes relative to SGD. All evaluated weight-noise and activation-noise levels are reported. Lower PPL and ASR are better, while higher lm-eval is better.

Perturbation	Level	PPL ↓	lm-eval ↑	HarmBench ASR ↓	LlamaGuard3 ASR ↓	AdvBench ASR ↓
Stage I: SGD Safety Alignment (change vs. Original)						
–	–	8.9998 (↑0.7157)	0.6695 (↓0.0081)	0.0113 (↓0.0045)	0.0068 (↓0.0030)	0.0173 (↓0.0060)
Quantization	W4A16	10.4094 (↑1.4357)	0.6384 (↓0.0050)	0.0218 (↓0.0052)	0.0180 (↓0.0015)	0.0353 (↑0.0008)
Quantization	W4A4	14.9724 (↑0.6281)	0.5503 (↑0.0020)	0.0428 (↓0.0420)	0.0480 (↓0.0525)	0.0728 (↓0.0510)
Weight noise	$\sigma = 0.10$	9.5091 (↑0.6453)	0.6613 (↓0.0027)	0.0135 (↓0.0015)	0.0093 (↓0.0005)	0.0220 (↓0.0048)
Weight noise	$\sigma = 0.15$	10.6183 (↑0.5954)	0.6387 (↓0.0074)	0.0140 (↓0.0033)	0.0090 (↓0.0040)	0.0223 (↓0.0065)
Weight noise	$\sigma = 0.20$	13.8321 (↑0.1883)	0.5845 (↓0.0066)	0.0193 (↓0.0045)	0.0173 (↓0.0067)	0.0350 (↓0.0070)
Weight noise	$\sigma = 0.30$	286.4590 (↑28.7973)	0.3289 (↓0.0024)	0.0478 (↑0.0085)	0.8580 (↓0.0398)	0.9292 (↓0.0093)
Activation noise	$\alpha = 0.05$	10.3312 (↑0.7417)	0.6403 (↓0.0087)	0.0203 (↓0.0025)	0.0153 (↑0.0013)	0.0300 (↓0.0038)
Activation noise	$\alpha = 0.08$	13.1343 (↑0.7532)	0.6015 (↓0.0081)	0.0368 (↓0.0070)	0.0338 (↓0.0082)	0.0598 (↓0.0180)
Activation noise	$\alpha = 0.10$	16.7710 (↑0.7316)	0.5651 (↓0.0022)	0.0643 (↓0.0087)	0.1435 (↓0.0220)	0.1340 (↓0.0365)
Activation noise	$\alpha = 0.12$	23.2695 (↑0.6776)	0.5134 (↓0.0032)	0.0813 (↓0.0117)	0.5074 (↓0.0105)	0.3138 (↓0.0503)
Stage II: ZO Refinement (change vs. SGD)						
–	–	8.9997 (↓0.0001)	0.6698 (↑0.0003)	0.0113 (→0.0000)	0.0068 (→0.0000)	0.0173 (→0.0000)
Quantization	W4A16	10.0560 (↓0.3534)	0.6350 (↓0.0034)	0.0165 (↓0.0053)	0.0150 (↓0.0030)	0.0263 (↓0.0090)
Quantization	W4A4	14.8228 (↓0.1496)	0.5535 (↑0.0032)	0.0315 (↓0.0113)	0.0360 (↓0.0120)	0.0600 (↓0.0128)
Weight noise	$\sigma = 0.10$	9.5092 (↑0.0001)	0.6611 (↓0.0002)	0.0115 (↓0.0020)	0.0080 (↓0.0013)	0.0188 (↓0.0032)
Weight noise	$\sigma = 0.15$	10.6182 (↓0.0001)	0.6386 (↓0.0001)	0.0143 (↑0.0003)	0.0090 (→0.0000)	0.0225 (↑0.0003)
Weight noise	$\sigma = 0.20$	13.8323 (↑0.0002)	0.5846 (↑0.0001)	0.0165 (↓0.0028)	0.0153 (↓0.0020)	0.0340 (↓0.0010)
Weight noise	$\sigma = 0.30$	286.4126 (↓0.0464)	0.3291 (↑0.0002)	0.0458 (↓0.0020)	0.8577 (↓0.0003)	0.9290 (↓0.0003)
Activation noise	$\alpha = 0.05$	10.3312 (→0.0000)	0.6405 (↑0.0002)	0.0140 (↓0.0063)	0.0095 (↓0.0058)	0.0230 (↓0.0070)
Activation noise	$\alpha = 0.08$	13.1342 (↓0.0001)	0.6013 (↓0.0002)	0.0323 (↓0.0045)	0.0295 (↓0.0043)	0.0618 (↑0.0020)
Activation noise	$\alpha = 0.10$	16.7709 (↓0.0001)	0.5649 (↓0.0002)	0.0528 (↓0.0115)	0.1323 (↓0.0112)	0.1203 (↓0.0137)
Activation noise	$\alpha = 0.12$	23.2692 (↓0.0003)	0.5135 (↑0.0001)	0.0683 (↓0.0130)	0.4809 (↓0.0265)	0.2961 (↓0.0177)

Table 7: Safety robustness and utility comparison between 400-step SGD and the subsequent 5-step ZO robustness refinement on Llama3-8B-Instruct. For Stage I, arrows indicate changes relative to the original model under the corresponding perturbation; for Stage II, arrows indicate changes relative to SGD. All evaluated weight-noise and activation-noise levels are reported. Lower PPL and ASR are better, while higher lm-eval is better.

Perturbation	Level	PPL ↓	lm-eval ↑	HarmBench ASR ↓	LlamaGuard3 ASR ↓	AdvBench ASR ↓
Stage I: SGD Safety Alignment (change vs. Original)						
–	–	8.9998 (↑0.7157)	0.6695 (↓0.0081)	0.0113 (↓0.0045)	0.0068 (↓0.0030)	0.0173 (↓0.0060)
Quantization	W4A16	10.4094 (↑1.4357)	0.6384 (↓0.0050)	0.0218 (↓0.0052)	0.0180 (↓0.0015)	0.0353 (↑0.0008)
Quantization	W4A4	14.9724 (↑0.6281)	0.5503 (↑0.0020)	0.0428 (↓0.0420)	0.0480 (↓0.0525)	0.0728 (↓0.0510)
Weight noise	$\sigma = 0.10$	9.5091 (↑0.6453)	0.6613 (↓0.0027)	0.0135 (↓0.0015)	0.0093 (↓0.0005)	0.0220 (↓0.0048)
Weight noise	$\sigma = 0.15$	10.6183 (↑0.5954)	0.6387 (↓0.0074)	0.0140 (↓0.0033)	0.0090 (↓0.0040)	0.0223 (↓0.0065)
Weight noise	$\sigma = 0.20$	13.8321 (↑0.1883)	0.5845 (↓0.0066)	0.0193 (↓0.0045)	0.0173 (↓0.0067)	0.0350 (↓0.0070)
Weight noise	$\sigma = 0.30$	286.4590 (↑28.7973)	0.3289 (↓0.0024)	0.0478 (↑0.0085)	0.8580 (↓0.0398)	0.9292 (↓0.0093)
Activation noise	$\alpha = 0.05$	10.3312 (↑0.7417)	0.6403 (↓0.0087)	0.0203 (↓0.0025)	0.0153 (↑0.0013)	0.0300 (↓0.0038)
Activation noise	$\alpha = 0.08$	13.1343 (↑0.7532)	0.6015 (↓0.0081)	0.0368 (↓0.0070)	0.0338 (↓0.0082)	0.0598 (↓0.0180)
Activation noise	$\alpha = 0.10$	16.7710 (↑0.7316)	0.5651 (↓0.0022)	0.0643 (↓0.0087)	0.1435 (↓0.0220)	0.1340 (↓0.0365)
Activation noise	$\alpha = 0.12$	23.2695 (↑0.6776)	0.5134 (↓0.0032)	0.0813 (↓0.0117)	0.5074 (↓0.0105)	0.3138 (↓0.0503)
Stage II: ZO Checkpoint-5 Refinement (change vs. SGD)						
–	–	8.9997 (↓0.0001)	0.6697 (↑0.0002)	0.0113 (→0.0000)	0.0068 (→0.0000)	0.0173 (→0.0000)
Quantization	W4A16	10.0014 (↓0.4080)	0.6377 (↓0.0007)	0.0338 (↑0.0120)	0.0285 (↑0.0105)	0.0435 (↑0.0082)
Quantization	W4A4	14.6522 (↓0.3202)	0.5519 (↑0.0016)	0.0668 (↑0.0240)	0.0765 (↑0.0285)	0.1050 (↑0.0322)
Weight noise	$\sigma = 0.10$	9.5091 (→0.0000)	0.6609 (↓0.0004)	0.0113 (↓0.0022)	0.0078 (↓0.0015)	0.0188 (↓0.0032)
Weight noise	$\sigma = 0.15$	10.6182 (↓0.0001)	0.6386 (↓0.0001)	0.0138 (↓0.0003)	0.0085 (↓0.0005)	0.0223 (→0.0000)
Weight noise	$\sigma = 0.20$	13.8321 (→0.0000)	0.5845 (→0.0000)	0.0168 (↓0.0025)	0.0153 (↓0.0020)	0.0345 (↓0.0005)
Weight noise	$\sigma = 0.30$	286.4704 (↑0.0114)	0.3290 (↑0.0001)	0.0453 (↓0.0025)	0.8590 (↑0.0010)	0.9285 (↓0.0008)
Activation noise	$\alpha = 0.05$	10.3312 (→0.0000)	0.6403 (→0.0000)	0.0148 (↓0.0055)	0.0108 (↓0.0045)	0.0235 (↓0.0065)
Activation noise	$\alpha = 0.08$	13.1341 (↓0.0003)	0.6015 (→0.0000)	0.0303 (↓0.0065)	0.0308 (↓0.0030)	0.0585 (↓0.0013)
Activation noise	$\alpha = 0.10$	16.7708 (↓0.0003)	0.5650 (↓0.0001)	0.0515 (↓0.0128)	0.1200 (↓0.0235)	0.1190 (↓0.0150)
Activation noise	$\alpha = 0.12$	23.2692 (↓0.0002)	0.5133 (↓0.0001)	0.0683 (↓0.0130)	0.4796 (↓0.0278)	0.2968 (↓0.0170)

Table 8: Safety robustness and utility of the SGD-aligned Llama-3-8B-Instruct checkpoint-410 under different perturbations. Arrows indicate changes relative to the corresponding results of the original model. Lower PPL and ASR are better, while higher lm-eval is better.

Perturbation	Level	PPL ↓	lm-eval ↑	HarmBench ASR ↓	LlamaGuard3 ASR ↓	AdvBench ASR ↓
-	-	9.0225 (↑0.7385)	0.6697 (↑0.0079)	0.0113 (↓0.0045)	0.0075 (↓0.0023)	0.0180 (↓0.0053)
Quantization	W4A16	10.0148 (↑1.0411)	0.6374 (↑0.0060)	0.0293 (↑0.0023)	0.0248 (↑0.0053)	0.0375 (↑0.0030)
Quantization	W4A4	15.0748 (↑0.7304)	0.5561 (↓0.0078)	0.0503 (↓0.0345)	0.0593 (↓0.0413)	0.0863 (↓0.0375)
Weight Noise	$\sigma = 0.10$	9.5277 (↑0.6326)	0.6605 (↑0.0083)	0.0105 (↓0.0045)	0.0068 (↓0.0030)	0.0188 (↓0.0080)
Weight Noise	$\sigma = 0.15$	10.6392 (↑0.6163)	0.6387 (↑0.0074)	0.0138 (↓0.0035)	0.0085 (↓0.0045)	0.0230 (↓0.0058)
Weight Noise	$\sigma = 0.20$	13.8786 (↑0.7417)	0.5842 (↑0.0060)	0.0178 (↓0.0060)	0.0148 (↓0.0093)	0.0348 (↓0.0073)
Activation Noise	$\alpha = 0.05$	10.3574 (↑0.7679)	0.6400 (↑0.0090)	0.0153 (↓0.0075)	0.0115 (↓0.0025)	0.0243 (↓0.0095)
Activation Noise	$\alpha = 0.08$	13.1647 (↑0.7836)	0.6003 (↑0.0093)	0.0355 (↓0.0083)	0.0340 (↓0.0080)	0.0605 (↓0.0173)
Activation Noise	$\alpha = 0.10$	16.8067 (↑0.7673)	0.5634 (↑0.0039)	0.0570 (↓0.0160)	0.1348 (↓0.0308)	0.1348 (↓0.0358)
Activation Noise	$\alpha = 0.12$	23.3285 (↑0.7366)	0.5136 (↑0.0030)	0.0710 (↓0.0220)	0.4936 (↓0.0243)	0.2991 (↓0.0650)

Table 9: Safety robustness and utility of Llama-3-8B-Instruct after 800-step AdamW safety alignment and subsequent 50-step AdaZO refinement. For Stage I, arrows indicate changes relative to the original model under the corresponding perturbation; for Stage II, arrows indicate changes relative to AdamW. Lower PPL and ASR are better, while higher lm-eval is better.

Perturbation	Level	PPL ↓	lm-eval ↑	HarmBench ASR ↓	LlamaGuard3 ASR ↓	AdvBench ASR ↓
Stage I: AdamW Safety Alignment (change vs. Original)						
–	–	9.8301 (↑1.5460)	0.6734 (↓0.0042)	0.0038 (↓0.0120)	0.0023 (↓0.0075)	0.0135 (↓0.0098)
Quantization	W4A16	11.3092 (↑2.3355)	0.6496 (↑0.0062)	0.0068 (↓0.0202)	0.0068 (↓0.0127)	0.0180 (↓0.0165)
Quantization	W4A4	14.9374 (↑0.5931)	0.5662 (↑0.0179)	0.0203 (↓0.0645)	0.0240 (↓0.0765)	0.0413 (↓0.0825)
Weight noise	$\sigma = 0.10$	10.3964 (↑1.5326)	0.6591 (↓0.0049)	0.0048 (↓0.0102)	0.0048 (↓0.0050)	0.0145 (↓0.0123)
Weight noise	$\sigma = 0.15$	11.5027 (↑1.4798)	0.6406 (↓0.0055)	0.0063 (↓0.0110)	0.0045 (↓0.0085)	0.0158 (↓0.0130)
Weight noise	$\sigma = 0.20$	14.5289 (↑0.8851)	0.5917 (↑0.0006)	0.0078 (↓0.0160)	0.0068 (↓0.0172)	0.0223 (↓0.0197)
Weight noise	$\sigma = 0.30$	237.7598 (↓19.9019)	0.3419 (↑0.0106)	0.0608 (↑0.0215)	0.3973 (↓0.5004)	0.2351 (↓0.7034)
Activation noise	$\alpha = 0.05$	11.2175 (↑1.6280)	0.6445 (↓0.0045)	0.0073 (↓0.0155)	0.0053 (↓0.0087)	0.0158 (↓0.0180)
Activation noise	$\alpha = 0.08$	14.0924 (↑1.7113)	0.6087 (↓0.0009)	0.0093 (↓0.0345)	0.0098 (↓0.0322)	0.0190 (↓0.0588)
Activation noise	$\alpha = 0.10$	17.7487 (↑1.7093)	0.5709 (↑0.0036)	0.0128 (↓0.0602)	0.0190 (↓0.1465)	0.0303 (↓0.1402)
Activation noise	$\alpha = 0.12$	24.2625 (↑1.6706)	0.5231 (↑0.0065)	0.0233 (↓0.0697)	0.0888 (↓0.4291)	0.0633 (↓0.3008)
Stage II: AdaZO Refinement (change vs. AdamW)						
–	–	9.8361 (↑0.0060)	0.6734 (→0.0000)	0.0038 (→0.0000)	0.0023 (→0.0000)	0.0135 (→0.0000)
Quantization	W4A16	11.1002 (↓0.2090)	0.6487 (↓0.0009)	0.0068 (→0.0000)	0.0045 (↓0.0023)	0.0158 (↓0.0022)
Quantization	W4A4	15.0795 (↑0.1421)	0.5584 (↓0.0078)	0.0143 (↓0.0060)	0.0128 (↓0.0112)	0.0263 (↓0.0150)
Weight noise	$\sigma = 0.10$	10.4019 (↑0.0055)	0.6592 (↑0.0001)	0.0053 (↑0.0005)	0.0040 (↓0.0008)	0.0138 (↓0.0007)
Weight noise	$\sigma = 0.15$	11.5077 (↑0.0050)	0.6402 (↓0.0004)	0.0058 (↓0.0005)	0.0043 (↓0.0003)	0.0155 (↓0.0003)
Weight noise	$\sigma = 0.20$	14.5329 (↑0.0040)	0.5927 (↑0.0010)	0.0073 (↓0.0005)	0.0060 (↓0.0008)	0.0190 (↓0.0033)
Weight noise	$\sigma = 0.30$	238.3253 (↑0.5655)	0.3422 (↑0.0003)	0.0595 (↓0.0013)	0.3951 (↓0.0023)	0.2316 (↓0.0035)
Activation noise	$\alpha = 0.05$	11.2220 (↑0.0045)	0.6443 (↓0.0002)	0.0040 (↓0.0033)	0.0030 (↓0.0023)	0.0148 (↓0.0010)
Activation noise	$\alpha = 0.08$	14.0940 (↑0.0016)	0.6091 (↑0.0004)	0.0083 (↓0.0010)	0.0080 (↓0.0018)	0.0200 (↑0.0010)
Activation noise	$\alpha = 0.10$	17.7472 (↓0.0015)	0.5709 (→0.0000)	0.0153 (↑0.0025)	0.0173 (↓0.0017)	0.0313 (↑0.0010)
Activation noise	$\alpha = 0.12$	24.2569 (↓0.0056)	0.5225 (↓0.0006)	0.0210 (↓0.0023)	0.0695 (↓0.0193)	0.0560 (↓0.0073)

Table 10: Safety robustness and utility of the AdamW-aligned Llama-3-8B-Instruct checkpoint-850 under different perturbations. Arrows indicate changes relative to the corresponding results of the original model. Lower PPL and ASR are better, while higher lm-eval is better.

Perturbation	Level	PPL ↓	lm-eval ↑	HarmBench ASR ↓	LlamaGuard3 ASR ↓	AdvBench ASR ↓
–	–	9.8871 (↑1.6030)	0.6723 (↓0.0053)	0.0060 (↓0.0098)	0.0030 (↓0.0068)	0.0105 (↓0.0128)
Quantization	W4A16	11.1918 (↑2.2181)	0.6474 (↑0.0040)	0.0113 (↓0.0157)	0.0090 (↓0.0105)	0.0158 (↓0.0187)
Quantization	W4A4	14.9403 (↑0.5960)	0.5562 (↑0.0079)	0.0143 (↓0.0705)	0.0165 (↓0.0840)	0.0270 (↓0.0968)
Weight Noise	$\sigma = 0.10$	10.4536 (↑1.5898)	0.6587 (↓0.0053)	0.0050 (↓0.0100)	0.0033 (↓0.0065)	0.0108 (↓0.0160)
Weight Noise	$\sigma = 0.15$	11.5505 (↑1.5277)	0.6406 (↓0.0055)	0.0050 (↓0.0123)	0.0040 (↓0.0090)	0.0130 (↓0.0158)
Weight Noise	$\sigma = 0.20$	14.5589 (↑0.9151)	0.5917 (↑0.0006)	0.0078 (↓0.0160)	0.0078 (↓0.0162)	0.0183 (↓0.0237)
Activation Noise	$\alpha = 0.05$	11.2682 (↑1.6787)	0.6449 (↓0.0041)	0.0060 (↓0.0168)	0.0048 (↓0.0092)	0.0128 (↓0.0210)
Activation Noise	$\alpha = 0.08$	14.1431 (↑1.7620)	0.6091 (↓0.0005)	0.0115 (↓0.0323)	0.0110 (↓0.0310)	0.0208 (↓0.0570)
Activation Noise	$\alpha = 0.10$	17.8287 (↑1.7894)	0.5699 (↑0.0026)	0.0165 (↓0.0565)	0.0210 (↓0.1445)	0.0308 (↓0.1397)
Activation Noise	$\alpha = 0.12$	24.4524 (↑1.8605)	0.5215 (↑0.0049)	0.0343 (↓0.0587)	0.1145 (↓0.4034)	0.0745 (↓0.2896)

Table 11: Safety robustness and utility of 400-step Vaccine and its subsequent 10-step ZO refinement on Llama-3-8B-Instruct. For Stage II, arrows indicate changes relative to the Vaccine source model. Lower PPL and ASR are better, while higher lm-eval is better.

Perturbation	Level	PPL ↓	lm-eval ↑	HarmBench ASR ↓	LlamaGuard3 ASR ↓	AdvBench ASR ↓
Stage I: Vaccine Source Model						
–	–	9.0256	0.6714	0.0120	0.0083	0.0165
Quantization	W4A16	9.9621	0.6349	0.0458	0.0398	0.0630
Quantization	W4A4	14.2567	0.5515	0.0503	0.0698	0.1013
Weight Noise	$\sigma = 0.10$	9.5114	0.6606	0.0093	0.0068	0.0158
Weight Noise	$\sigma = 0.20$	13.4416	0.5915	0.0148	0.0108	0.0293
Activation Noise	$\alpha = 0.05$	10.3092	0.6421	0.0135	0.0078	0.0195
Activation Noise	$\alpha = 0.08$	13.0251	0.6051	0.0250	0.0248	0.0445
Activation Noise	$\alpha = 0.10$	16.5458	0.5695	0.0498	0.1090	0.1155
Activation Noise	$\alpha = 0.12$	22.7204	0.5221	0.0688	0.4351	0.2613
Stage II: ZO Refinement (change vs. Vaccine Source Model)						
–	–	9.0257 (↑0.0001)	0.6712 (↓0.0002)	0.0113 (↓0.0008)	0.0083 (→0.0000)	0.0165 (→0.0000)
Quantization	W4A16	9.9303 (↓0.0318)	0.6352 (↑0.0003)	0.0353 (↓0.0105)	0.0278 (↓0.0120)	0.0450 (↓0.0180)
Quantization	W4A4	14.3990 (↑0.1424)	0.5612 (↑0.0097)	0.0420 (↓0.0083)	0.0488 (↓0.0210)	0.0750 (↓0.0263)
Weight Noise	$\sigma = 0.10$	9.5112 (↓0.0001)	0.6607 (↑0.0001)	0.0093 (→0.0000)	0.0065 (↓0.0003)	0.0158 (→0.0000)
Weight Noise	$\sigma = 0.20$	13.4416 (→0.0000)	0.5912 (↓0.0003)	0.0148 (→0.0000)	0.0113 (↑0.0005)	0.0290 (↓0.0003)
Activation Noise	$\alpha = 0.05$	10.3092 (→0.0000)	0.6422 (↑0.0001)	0.0130 (↓0.0005)	0.0088 (↑0.0010)	0.0208 (↑0.0013)
Activation Noise	$\alpha = 0.08$	13.0250 (→0.0000)	0.6054 (↑0.0003)	0.0250 (→0.0000)	0.0215 (↓0.0033)	0.0453 (↑0.0008)
Activation Noise	$\alpha = 0.10$	16.5455 (↓0.0003)	0.5691 (↓0.0004)	0.0403 (↓0.0095)	0.0985 (↓0.0105)	0.1118 (↓0.0038)
Activation Noise	$\alpha = 0.12$	22.7205 (→0.0000)	0.5222 (↑0.0001)	0.0698 (↑0.0010)	0.4369 (↑0.0018)	0.2618 (↑0.0005)

Table 12: Safety robustness and utility of 400-step ASFT and its subsequent 10-step ZO refinement on Llama-3-8B-Instruct. For Stage II, arrows indicate changes relative to the ASFT source model. Lower PPL and ASR are better, while higher lm-eval is better.

Perturbation	Level	PPL ↓	lm-eval ↑	HarmBench ASR ↓	LlamaGuard3 ASR ↓	AdvBench ASR ↓
Stage I: ASFT Source Model						
–	–	8.9978	0.6699	0.0113	0.0068	0.0173
Quantization	W4A16	9.9718	0.6340	0.0240	0.0203	0.0323
Quantization	W4A4	14.3982	0.5461	0.0368	0.0458	0.0690
Weight Noise	$\sigma = 0.10$	9.5067	0.6612	0.0110	0.0075	0.0180
Weight Noise	$\sigma = 0.20$	13.8248	0.5849	0.0173	0.0153	0.0338
Activation Noise	$\alpha = 0.05$	10.3300	0.6402	0.0133	0.0085	0.0230
Activation Noise	$\alpha = 0.08$	13.1352	0.6017	0.0305	0.0318	0.0535
Activation Noise	$\alpha = 0.10$	16.7748	0.5650	0.0570	0.1310	0.1285
Activation Noise	$\alpha = 0.12$	23.2736	0.5135	0.0680	0.4774	0.2963
Stage II: ZO Refinement (change vs. ASFT Source Model)						
–	–	8.9976 (↓0.0002)	0.6698 (↓0.0001)	0.0113 (→0.0000)	0.0068 (→0.0000)	0.0173 (→0.0000)
Quantization	W4A16	10.1098 (↑0.1380)	0.6341 (↑0.0001)	0.0210 (↓0.0030)	0.0165 (↓0.0038)	0.0323 (→0.0000)
Quantization	W4A4	14.5315 (↑0.1334)	0.5505 (↑0.0044)	0.0368 (→0.0000)	0.0533 (↑0.0075)	0.0578 (↓0.0113)
Weight Noise	$\sigma = 0.10$	9.5069 (↑0.0002)	0.6607 (↓0.0005)	0.0113 (↑0.0003)	0.0075 (→0.0000)	0.0180 (→0.0000)
Weight Noise	$\sigma = 0.20$	13.8247 (↓0.0001)	0.5848 (↓0.0001)	0.0165 (↓0.0008)	0.0145 (↓0.0008)	0.0338 (→0.0000)
Activation Noise	$\alpha = 0.05$	10.3299 (↓0.0001)	0.6403 (↑0.0001)	0.0145 (↑0.0013)	0.0110 (↑0.0025)	0.0253 (↑0.0023)
Activation Noise	$\alpha = 0.08$	13.1350 (↓0.0001)	0.6015 (↓0.0002)	0.0293 (↓0.0013)	0.0310 (↓0.0008)	0.0598 (↑0.0063)
Activation Noise	$\alpha = 0.10$	16.7744 (↓0.0004)	0.5649 (↓0.0001)	0.0518 (↓0.0053)	0.1228 (↓0.0083)	0.1258 (↓0.0028)
Activation Noise	$\alpha = 0.12$	23.2731 (↓0.0005)	0.5134 (↓0.0001)	0.0675 (↓0.0005)	0.4769 (↓0.0005)	0.2961 (↓0.0003)

Table 13: Safety robustness comparison of the SGD-aligned source model, score-specific top-4 masks, and the combined-score top-4 mask. All pruned models start from the aligned SGD checkpoint-400; all results use checkpoint-5. The weight, activation, quantization, and combined-score masks select layers $\{1, 13, 14, 31\}$, $\{2, 3, 4, 5\}$, $\{10, 13, 17, 31\}$, and $\{8, 11, 13, 14\}$, respectively. Layer indices are zero-based. Mean ASR is the arithmetic mean of the HarmBench, LlamaGuard3, and AdvBench ASRs. Parentheses denote absolute ASR percentage-point changes relative to the Stage I reference. Relative Gain is calculated from Mean ASR using the Stage I reference. Bold indicates the best result within each perturbation setting; ties are bolded together.

Setting	Perturbation	Level	HarmBench ASR (%) ↓	LlamaGuard3 ASR (%) ↓	AdvBench ASR (%) ↓	Mean ASR (%) ↓	Relative Gain ↑
Stage I: SGD Safety Alignment Reference							
Source	Clean	–	**1.1253**	**0.6752**	**1.7254**	**1.1753**	–
Source	Weight Noise	$\sigma = 0.10$	1.3503	0.9252	2.2006	1.4920	–
Source	Weight Noise	$\sigma = 0.15$	1.4004	0.9002	2.2256	1.5087	–
Source	Weight Noise	$\sigma = 0.20$	**1.9255**	**1.7254**	3.5009	2.3839	–
Source	Weight Noise	$\sigma = 0.25$	7.4769	16.4291	8.0770	10.6610	–
Source	Activation Noise	$\alpha = 0.05$	2.0255	1.5254	3.0008	2.1839	–
Source	Activation Noise	$\alpha = 0.08$	3.6759	3.3758	5.9765	4.3428	–
Source	Activation Noise	$\alpha = 0.10$	6.4266	14.3536	**13.4034**	11.3945	–
Source	Activation Noise	$\alpha = 0.12$	8.1270	50.7377	**31.3828**	30.0825	–
Source	Quantization	W4A16	2.1755	1.8005	3.5259	2.5006	–
Source	Quantization	W4A4D16	4.2761	4.8012	7.2768	5.4514	–
Score-Specific Top-4 Masks, Checkpoint-5							
ALL masks	Clean	–	**1.1253** (→0.0000)	**0.6752** (→0.0000)	**1.7254** (→0.0000)	**1.1753** (→0.0000)	**0.00%**
Weight mask	Weight Noise	$\sigma = 0.10$	**1.1253** (↓0.2250)	**0.7002** (↓0.2250)	**1.7004** (↓0.5002)	**1.1753** (↓0.3167)	**+21.23%**
Weight mask	Weight Noise	$\sigma = 0.15$	1.3253 (↓0.0751)	**0.8502** (↓0.0500)	**1.8005** (↓0.4251)	**1.3253** (↓0.1834)	**+12.15%**
Weight mask	Weight Noise	$\sigma = 0.20$	2.2006 (↑0.2751)	1.8505 (↑0.1251)	2.7257 (↓0.7752)	2.2589 (↓0.1250)	+5.24%
Weight mask	Weight Noise	$\sigma = 0.25$	**7.2518** (↓0.2251)	**16.3041** (↓0.1250)	**8.0020** (↓0.0750)	**10.5193** (↓0.1417)	**+1.33%**
Activation mask	Activation Noise	$\alpha = 0.05$	**1.4754** (↓0.5501)	1.1753 (↓0.3501)	2.6007 (↓0.4001)	**1.7504** (↓0.4334)	**+19.85%**
Activation mask	Activation Noise	$\alpha = 0.08$	3.1008 (↓0.5751)	**2.8007** (↓0.5751)	5.4514 (↓0.5251)	3.7843 (↓0.5585)	+12.86%
Activation mask	Activation Noise	$\alpha = 0.10$	**6.1015** (↓0.3251)	**12.6282** (↓1.7254)	13.4784 (↑0.0750)	**10.7360** (↓0.6585)	**+5.78%**
Activation mask	Activation Noise	$\alpha = 0.12$	**7.4269** (↓0.7001)	48.3871 (↓2.3506)	**31.3828** (→0.0000)	**29.0656** (↓1.0169)	**+3.38%**
Quantization mask	Quantization	W4A16	2.1755 (→0.0000)	1.7254 (↓0.0751)	3.1508 (↓0.3751)	2.3506 (↓0.1500)	+6.00%
Quantization mask	Quantization	W4A4D16	3.2258 (↓1.0503)	3.9760 (↓0.8252)	6.6017 (↓0.6751)	4.6012 (↓0.8502)	+15.60%
Combined-Score Top-4 Mask, Checkpoint-5							
Combined mask	Clean	–	**1.1253** (→0.0000)	**0.6752** (→0.0000)	**1.7254** (→0.0000)	**1.1753** (→0.0000)	**0.00%**
Combined mask	Weight Noise	$\sigma = 0.10$	1.1503 (↓0.2000)	**0.7002** (↓0.2250)	**1.7004** (↓0.5002)	1.1836 (↓0.3084)	+20.67%
Combined mask	Weight Noise	$\sigma = 0.15$	**1.3003** (↓0.1001)	0.8752 (↓0.0250)	**1.8005** (↓0.4251)	**1.3253** (↓0.1834)	**+12.15%**
Combined mask	Weight Noise	$\sigma = 0.20$	2.1255 (↑0.2000)	1.8505 (↑0.1251)	**2.7007** (↓0.8002)	**2.2256** (↓0.1584)	**+6.64%**
Combined mask	Weight Noise	$\sigma = 0.25$	7.4519 (↓0.0250)	**16.3041** (↓0.1250)	8.0520 (↓0.0250)	10.6027 (↓0.0583)	+0.55%
Combined mask	Activation Noise	$\alpha = 0.05$	1.5754 (↓0.4501)	**1.1503** (↓0.3751)	**2.5256** (↓0.4752)	**1.7504** (↓0.4334)	**+19.85%**
Combined mask	Activation Noise	$\alpha = 0.08$	**2.9007** (↓0.7752)	2.9257 (↓0.4501)	**4.8762** (↓1.1003)	**3.5676** (↓0.7752)	**+17.85%**
Combined mask	Activation Noise	$\alpha = 0.10$	6.1765 (↓0.2501)	13.0533 (↓1.3003)	13.5784 (↑0.1750)	10.9361 (↓0.4584)	+4.02%
Combined mask	Activation Noise	$\alpha = 0.12$	**7.4269** (↓0.7001)	**48.3621** (↓2.3756)	31.4079 (↑0.0251)	**29.0656** (↓1.0169)	**+3.38%**
Combined mask	Quantization	W4A16	**1.6504** (↓0.5251)	**1.5004** (↓0.3001)	**2.7757** (↓0.7502)	**1.9755** (↓0.5251)	**+21.00%**
Combined mask	Quantization	W4A4D16	**3.1508** (↓1.1253)	**3.5259** (↓1.2753)	**5.7764** (↓1.5004)	**4.1510** (↓1.3003)	**+23.85%**

Table 14: Full robustness results of the weight-score top-4 mask at checkpoint-5, compared with the aligned SGD checkpoint-400 source model. The mask selects zero-based layers $\{1, 13, 14, 31\}$. Arrows denote absolute ASR percentage-point changes relative to the source. Relative gain is computed from the mean of HarmBench and LlamaGuard3 ASR. Lower ASR is better.

Setting	Perturbation	Level	HarmBench ASR (%) ↓	LlamaGuard3 ASR (%) ↓	Mean ASR (%) ↓	Relative Gain ↑
Source: aligned SGD Checkpoint-400						
Source	Clean	–	1.1253	0.6752	0.9002	–
Source	Weight Noise	$\sigma = 0.10$	1.1753	0.7002	0.9377	–
Source	Weight Noise	$\sigma = 0.15$	1.3253	0.8752	1.1003	–
Source	Weight Noise	$\sigma = 0.20$	**2.0755**	1.8755	**1.9755**	–
Source	Weight Noise	$\sigma = 0.25$	7.4769	16.4291	11.9530	–
Source	Activation Noise	$\alpha = 0.05$	1.7754	1.3503	1.5629	–
Source	Activation Noise	$\alpha = 0.08$	3.1258	3.1508	3.1383	–
Source	Activation Noise	$\alpha = 0.10$	**5.8765**	13.3283	9.6024	–
Source	Activation Noise	$\alpha = 0.12$	**7.3018**	**48.2371**	**27.7694**	–
Source	Quantization	W4A16	**2.2506**	2.1755	**2.2131**	–
Source	Quantization	W4A4D16	**3.0758**	**4.6512**	**3.8635**	–
Weight-score Top-4 Mask, Checkpoint-5						
Weight mask	Clean	–	1.1253 (→0.0000)	0.6752 (→0.0000)	0.9002 (→0.0000)	0.00%
Weight mask	Weight Noise	$\sigma = 0.10$	**1.1253** (↓0.0500)	0.7002 (→0.0000)	**0.9127** (↓0.0250)	**+2.67%**
Weight mask	Weight Noise	$\sigma = 0.15$	1.3253 (→0.0000)	**0.8502** (↓0.0250)	**1.0878** (↓0.0125)	**+1.14%**
Weight mask	Weight Noise	$\sigma = 0.20$	2.2006 (↑0.1250)	**1.8505** (↓0.0250)	2.0255 (↑0.0500)	-2.53%
Weight mask	Weight Noise	$\sigma = 0.25$	**7.2518** (↓0.2251)	**16.3041** (↓0.1250)	**11.7779** (↓0.1750)	**+1.46%**
Weight mask	Activation Noise	$\alpha = 0.05$	**1.7504** (↓0.0250)	**1.2503** (↓0.1000)	**1.5004** (↓0.0625)	**+4.00%**
Weight mask	Activation Noise	$\alpha = 0.08$	**3.0508** (↓0.0750)	**2.4756** (↓0.6752)	**2.7632** (↓0.3751)	**+11.95%**
Weight mask	Activation Noise	$\alpha = 0.10$	6.2266 (↑0.3501)	**12.5281** (↓0.8002)	**9.3773** (↓0.2251)	**+2.34%**
Weight mask	Activation Noise	$\alpha = 0.12$	7.6019 (↑0.3001)	48.3121 (↑0.0750)	27.9570 (↑0.1875)	-0.68%
Weight mask	Quantization	W4A16	2.4756 (↑0.2251)	**2.1005** (↓0.0750)	2.2881 (↑0.0750)	-3.39%
Weight mask	Quantization	W4A4D16	4.5011 (↑1.4254)	5.1763 (↑0.5251)	4.8387 (↑0.9752)	-25.24%

Table 15: Comparison of the mean results from three random top-4 masks and the combined-score top-4 mask. All models start from the aligned SGD checkpoint-400 and use the same ZO configuration; all results use checkpoint-5. The random masks select layers $\{8, 11, 21, 28\}$ (seed 101), $\{5, 10, 13, 14\}$ (seed 202), and $\{6, 11, 13, 27\}$ (seed 303). The combined-score mask selects layers $\{8, 11, 13, 14\}$. Layer indices are zero-based. Mean ASR is the arithmetic mean of the HarmBench, LlamaGuard3, and AdvBench ASRs. Parentheses denote absolute ASR percentage-point changes relative to the Stage I reference. Relative Gain is calculated from Mean ASR using the Stage I reference. The reference is not repeated as a table section. Bold indicates the best result between the random-mask mean and combined mask within each perturbation setting; ties are bolded together.

Setting	Perturbation	Level	HarmBench ASR (%) ↓	LlamaGuard3 ASR (%) ↓	AdvBench ASR (%) ↓	Mean ASR (%) ↓	Relative Gain ↑
Random Top-4 Masks: Mean of Three Checkpoint-5 Models							
Random mean	Clean	–	**1.1253** (→0.0000)	0.7002 (↑0.0250)	**1.7254** (→0.0000)	1.1836 (↑0.0083)	-0.71%
Random mean	Weight Noise	$\sigma = 0.10$	**1.1503** (↓0.2000)	0.7085 (↓0.2167)	**1.7004** (↓0.5002)	1.1864 (↓0.3056)	+20.48%
Random mean	Weight Noise	$\sigma = 0.15$	1.3087 (↓0.0917)	0.8919 (↓0.0083)	**1.8005** (↓0.4251)	1.3337 (↓0.1750)	+11.60%
Random mean	Weight Noise	$\sigma = 0.20$	**2.1172** (↑0.1917)	**1.8338** (↑0.1084)	2.7090 (↓0.7919)	**2.2200** (↓0.1639)	**+6.88%**
Random mean	Weight Noise	$\sigma = 0.25$	**7.3102** (↓0.1667)	16.4958 (↑0.0667)	**8.0520** (↓0.0250)	10.6193 (↓0.0417)	+0.39%
Random mean	Activation Noise	$\alpha = 0.05$	1.7838 (↓0.2417)	1.1670 (↓0.3584)	2.5756 (↓0.4252)	1.8421 (↓0.3418)	+15.65%
Random mean	Activation Noise	$\alpha = 0.08$	3.1008 (↓0.5751)	3.0591 (↓0.3167)	5.4597 (↓0.5168)	3.8732 (↓0.4696)	+10.81%
Random mean	Activation Noise	$\alpha = 0.10$	**6.1349** (↓0.2917)	**13.0283** (↓1.3253)	13.6617 (↑0.2583)	10.9416 (↓0.4529)	+3.97%
Random mean	Activation Noise	$\alpha = 0.12$	7.4935 (↓0.6335)	48.3704 (↓2.3673)	**31.3078** (↓0.0750)	**29.0573** (↓1.0253)	**+3.41%**
Random mean	Quantization	W4A16	2.2006 (↑0.0251)	1.9005 (↑0.1000)	3.1258 (↓0.4001)	2.4089 (↓0.0917)	+3.67%
Random mean	Quantization	W4A4D16	3.3008 (↓0.9753)	4.1510 (↓0.6502)	6.6767 (↓0.6001)	4.7095 (↓0.7419)	+13.61%
Combined-Score Top-4 Mask, Checkpoint-5							
Combined mask	Clean	–	**1.1253** (→0.0000)	**0.6752** (→0.0000)	**1.7254** (→0.0000)	**1.1753** (→0.0000)	**0.00%**
Combined mask	Weight Noise	$\sigma = 0.10$	**1.1503** (↓0.2000)	**0.7002** (↓0.2250)	**1.7004** (↓0.5002)	**1.1836** (↓0.3084)	**+20.67%**
Combined mask	Weight Noise	$\sigma = 0.15$	**1.3003** (↓0.1001)	**0.8752** (↓0.0250)	**1.8005** (↓0.4251)	**1.3253** (↓0.1834)	**+12.15%**
Combined mask	Weight Noise	$\sigma = 0.20$	2.1255 (↑0.2000)	1.8505 (↑0.1251)	**2.7007** (↓0.8002)	2.2256 (↓0.1584)	+6.64%
Combined mask	Weight Noise	$\sigma = 0.25$	7.4519 (↓0.0250)	**16.3041** (↓0.1250)	**8.0520** (↓0.0250)	**10.6027** (↓0.0583)	**+0.55%**
Combined mask	Activation Noise	$\alpha = 0.05$	**1.5754** (↓0.4501)	**1.1503** (↓0.3751)	**2.5256** (↓0.4752)	**1.7504** (↓0.4334)	**+19.85%**
Combined mask	Activation Noise	$\alpha = 0.08$	**2.9007** (↓0.7752)	**2.9257** (↓0.4501)	**4.8762** (↓1.1003)	**3.5676** (↓0.7752)	**+17.85%**
Combined mask	Activation Noise	$\alpha = 0.10$	6.1765 (↓0.2501)	13.0533 (↓1.3003)	**13.5784** (↑0.1750)	**10.9361** (↓0.4584)	**+4.02%**
Combined mask	Activation Noise	$\alpha = 0.12$	**7.4269** (↓0.7001)	**48.3621** (↓2.3756)	31.4079 (↑0.0251)	29.0656 (↓1.0169)	+3.38%
Combined mask	Quantization	W4A16	**1.6504** (↓0.5251)	**1.5004** (↓0.3001)	**2.7757** (↓0.7502)	**1.9755** (↓0.5251)	**+21.00%**
Combined mask	Quantization	W4A4D16	**3.1508** (↓1.1253)	**3.5259** (↓1.2753)	**5.7764** (↓1.5004)	**4.1510** (↓1.3003)	**+23.85%**

Table 16: Complete results of 10-step ZO robustness refinement with different layer-selection strategies, evaluated exclusively on the HarmBench subset. SNIP updates layers 15, 16, 14, and 13; WANDA updates layers 31, 30, 29, and 28; robustness-aware selection updates layers 24, 12, 31, and 29. Bold numbers indicate the best ASR among the three layer-selection strategies under the same perturbation setting.

Method	Perturbation	Level	PPL	lm-eval	HarmBench ASR	LlamaGuard3 ASR	AdvBench ASR
ZO refinement with SNIP-selected layers							
SNIP + ZO	–	–	9.4227	0.6656	**0.0700**	**0.0567**	**0.0767**
SNIP + ZO	Quantization	W4A16	10.4175	0.6230	0.1133	0.0933	0.1267
SNIP + ZO	Quantization	W4A4	14.5035	0.5428	0.1667	0.2333	0.3167
SNIP + ZO	Weight noise	$\sigma = 1$	11.5789	0.6500	**0.0900**	**0.0667**	**0.1067**
SNIP + ZO	Weight noise	$\sigma = 1.5$	14.9834	0.6306	0.0933	**0.0767**	**0.1267**
SNIP + ZO	Weight noise	$\sigma = 2$	21.5331	0.6139	**0.1067**	0.0967	**0.1733**
SNIP + ZO	Activation noise	$\alpha = 0.05$	10.6951	0.6358	0.0867	**0.0700**	0.1000
SNIP + ZO	Activation noise	$\alpha = 0.08$	13.3717	0.6037	0.1567	0.1400	0.1800
SNIP + ZO	Activation noise	$\alpha = 0.10$	16.8277	0.5680	0.2167	0.2633	0.2500
SNIP + ZO	Activation noise	$\alpha = 0.12$	22.9064	0.5135	0.2233	0.6133	0.4333
ZO refinement with WANDA-selected layers							
WANDA + ZO	–	–	9.4228	0.6655	**0.0700**	**0.0567**	**0.0767**
WANDA + ZO	Quantization	W4A16	10.4033	0.6239	0.1367	0.1200	0.1600
WANDA + ZO	Quantization	W4A4	14.6934	0.5448	0.1933	0.2633	0.3733
WANDA + ZO	Weight noise	$\sigma = 1$	11.5790	0.6498	**0.0900**	**0.0667**	**0.1067**
WANDA + ZO	Weight noise	$\sigma = 1.5$	14.9836	0.6309	0.0933	**0.0767**	**0.1267**
WANDA + ZO	Weight noise	$\sigma = 2$	21.5335	0.6139	**0.1067**	**0.0933**	**0.1733**
WANDA + ZO	Activation noise	$\alpha = 0.05$	10.6951	0.6357	0.0967	0.0767	**0.0900**
WANDA + ZO	Activation noise	$\alpha = 0.08$	13.3718	0.6039	0.1533	0.1400	0.1800
WANDA + ZO	Activation noise	$\alpha = 0.10$	16.8279	0.5680	0.2167	0.2633	0.2500
WANDA + ZO	Activation noise	$\alpha = 0.12$	22.9064	0.5135	0.2233	0.6133	0.4333
ZO refinement with robustness-aware selected layers							
Robust + ZO	–	–	9.4229	0.6655	**0.0700**	**0.0567**	**0.0767**
Robust + ZO	Quantization	W4A16	10.4091	0.6220	**0.0900**	**0.0733**	**0.1000**
Robust + ZO	Quantization	W4A4	14.4467	0.5440	**0.1233**	**0.1767**	**0.2733**
Robust + ZO	Weight noise	$\sigma = 1$	11.5791	0.6497	**0.0900**	**0.0667**	**0.1067**
Robust + ZO	Weight noise	$\sigma = 1.5$	14.9837	0.6306	**0.0900**	**0.0767**	**0.1267**
Robust + ZO	Weight noise	$\sigma = 2$	21.5337	0.6137	**0.1067**	0.0967	**0.1733**
Robust + ZO	Activation noise	$\alpha = 0.05$	10.6951	0.6358	**0.0833**	0.0767	**0.0900**
Robust + ZO	Activation noise	$\alpha = 0.08$	13.3717	0.6039	**0.1300**	**0.1367**	**0.1567**
Robust + ZO	Activation noise	$\alpha = 0.10$	16.8278	0.5683	**0.2033**	**0.2367**	**0.2333**
Robust + ZO	Activation noise	$\alpha = 0.12$	22.9063	0.5136	**0.2167**	**0.5600**	**0.4300**

Table 17: Comparison between 100-step SGD and the subsequent 10-step ZO robustness refinement on Qwen2-7B-Instruct model. For Stage I, arrows indicate changes relative to the original model under the corresponding perturbation; for Stage II, arrows indicate changes relative to SGD. All evaluated weight-noise and activation-noise levels are reported. Lower PPL and ASR are better, while higher lm-eval is better.

Method	Perturbation	Level	PPL	lm-eval	HarmBench ASR	LlamaGuard3 ASR	AdvBench ASR
Original Model							
Original	–	–	7.6023	0.6850	0.0353	0.0345	0.0705
Original	Quantization	W4A16/KV16/D16	7.9847	0.6678	0.0203	0.0270	0.0638
Original	Weight noise	$\sigma_w = 0.20$	10.1683	0.6515	0.0560	0.0743	0.1175
Original	Weight noise	$\sigma_w = 0.30$	16.5215	0.5769	0.0965	0.1743	0.2478
Original	Weight noise	$\sigma_w = 0.35$	26.1468	0.5035	0.0713	0.3661	0.3886
Original	Activation noise	$\alpha = 0.05$	7.8916	0.6683	0.0375	0.0398	0.0758
Original	Activation noise	$\alpha = 0.10$	8.7474	0.6343	0.0478	0.0613	0.1018
Original	Activation noise	$\alpha = 0.15$	10.4793	0.5743	0.1010	0.1283	0.1948
Original	Activation noise	$\alpha = 0.20$	13.7343	0.4995	0.1450	0.2976	0.4284
Stage I: After 100-step SFT							
SFT	–	–	7.6409	0.6796	0.0045	0.0038	0.0045
SFT	Quantization	W4A16/KV16/D16	8.0360	0.6725	0.0023	0.0030	0.0060
SFT	Weight noise	$\sigma_w = 0.20$	10.1572	0.6486	0.0053	0.0063	0.0138
SFT	Weight noise	$\sigma_w = 0.30$	16.3609	0.5781	0.0138	0.0353	0.0388
SFT	Weight noise	$\sigma_w = 0.35$	25.6133	0.5100	0.0215	0.1463	0.1050
SFT	Activation noise	$\alpha = 0.05$	7.9294	0.6696	0.0043	0.0025	0.0063
SFT	Activation noise	$\alpha = 0.10$	8.7776	0.6378	0.0090	0.0095	0.0128
SFT	Activation noise	$\alpha = 0.15$	10.4961	0.5800	0.0188	0.0413	0.0333
SFT	Activation noise	$\alpha = 0.20$	13.7226	0.5022	0.0538	0.1933	0.1613
Stage II: After 100-step SFT + 10-step ZO refinement							
SFT + ZO	–	–	7.6407 (↑0.0002)	0.6800 (↓0.0004)	0.0045 (→0.0000)	0.0038 (→0.0000)	0.0045 (→0.0000)
SFT + ZO	Quantization	W4A16/KV16/D16	8.0475 (↑0.0115)	0.6726 (↓0.0001)	0.0008 (↑0.0015)	0.0023 (↑0.0007)	0.0045 (↑0.0015)
SFT + ZO	Weight noise	$\sigma_w = 0.20$	10.1571 (↑0.0001)	0.6489 (↓0.0003)	0.0050 (↑0.0003)	0.0063 (→0.0000)	0.0135 (↑0.0003)
SFT + ZO	Weight noise	$\sigma_w = 0.30$	16.3613 (↑0.0004)	0.5783 (↓0.0002)	0.0148 (↑0.0010)	0.0350 (↑0.0003)	0.0388 (→0.0000)
SFT + ZO	Weight noise	$\sigma_w = 0.35$	25.6134 (↑0.0001)	0.5104 (↓0.0004)	0.0213 (↑0.0002)	0.1480 (↑0.0017)	0.1063 (↑0.0013)
SFT + ZO	Activation noise	$\alpha = 0.05$	7.9294 (→0.0000)	0.6696 (→0.0000)	0.0023 (↑0.0020)	0.0023 (↑0.0002)	0.0058 (↑0.0005)
SFT + ZO	Activation noise	$\alpha = 0.10$	8.7776 (→0.0000)	0.6380 (↓0.0002)	0.0078 (↑0.0012)	0.0085 (↑0.0010)	0.0090 (↑0.0038)
SFT + ZO	Activation noise	$\alpha = 0.15$	10.4962 (↑0.0001)	0.5798 (↑0.0002)	0.0150 (↑0.0038)	0.0403 (↑0.0010)	0.0328 (↑0.0005)
SFT + ZO	Activation noise	$\alpha = 0.20$	13.7223 (↑0.0003)	0.5023 (↓0.0001)	0.0505 (↑0.0033)	0.1890 (↑0.0043)	0.1438 (↑0.0175)

Table 18: Extra-jailbreak ASR comparison among the Qwen2-7B-Instruct SFT checkpoint-100 source model, score-specific top-4 mask models, and the combined-score top-4 mask model. All refined models use checkpoint-10 and the matched n64 ZO configuration. The score-specific models select layers $\{18, 19, 26, 27\}$ for weight noise, $\{5, 8, 9, 12\}$ for activation noise, and $\{10, 17, 22, 27\}$ for quantization. The shared combined-score model selects layers $\{10, 12, 18, 27\}$. Mean ASR is the arithmetic mean of the HarmBench, LlamaGuard3, and AdvBench ASRs. Parentheses denote absolute ASR percentage-point changes relative to the Stage I reference. Relative Gain is calculated from Mean ASR using the Stage I reference. Bold indicates the best result within each perturbation setting; ties are bolded together.

Setting	Perturbation	Level	HarmBench ASR (%) ↓	LlamaGuard3 ASR (%) ↓	Substring ASR (%) ↓	Mean ASR (%) ↓	Relative Gain ↑
Source: Qwen2-7B-Instruct SFT Checkpoint-100							
Source	Clean	–	**0.3751**	**0.3001**	**0.4501**	**0.3751**	–
Source	Weight Noise	$\sigma = 0.20$	0.5501	0.6502	1.6004	0.9336	–
Source	Weight Noise	$\sigma = 0.30$	1.1753	2.0255	3.7009	2.3006	–
Source	Weight Noise	$\sigma = 0.35$	1.6254	**3.7509**	5.2013	3.5259	–
Source	Activation Noise	$\alpha = 0.05$	0.4001	0.4501	0.8002	0.5501	–
Source	Activation Noise	$\alpha = 0.10$	1.2253	1.5504	1.5504	1.4420	–
Source	Activation Noise	$\alpha = 0.15$	3.4509	8.1770	4.7012	5.4430	–
Source	Activation Noise	$\alpha = 0.20$	5.4264	22.6307	16.5541	14.8704	–
Source	Quantization	W4A16	0.3001	0.3001	1.2003	0.6002	–
Score-Specific Top-4 Masks, Checkpoint-10							
All masks	Clean	–	**0.3751** (→0.0000)	0.3751 (↑0.0750)	**0.4501** (→0.0000)	0.4001 (↑0.0250)	-6.67%
Weight mask	Weight Noise	$\sigma = 0.20$	**0.4501** (↓0.1000)	**0.6252** (↓0.0250)	**1.2503** (↓0.3501)	**0.7752** (↓0.1584)	**+16.97%**
Weight mask	Weight Noise	$\sigma = 0.30$	**1.0003** (↓0.1750)	**1.8505** (↓0.1750)	**2.9507** (↓0.7502)	**1.9338** (↓0.3668)	**+15.94%**
Weight mask	Weight Noise	$\sigma = 0.35$	1.7504 (↑0.1250)	8.0770 (↑4.3261)	7.3768 (↑2.1755)	5.7348 (↑2.2089)	-62.65%
Activation mask	Activation Noise	$\alpha = 0.05$	0.4251 (↑0.0250)	0.3251 (↓0.1250)	0.7002 (↓0.1000)	0.4835 (↓0.0666)	+12.11%
Activation mask	Activation Noise	$\alpha = 0.10$	0.7002 (↓0.5251)	**0.5751** (↓0.9753)	**0.8502** (↓0.7002)	**0.7085** (↓0.7335)	**+50.87%**
Activation mask	Activation Noise	$\alpha = 0.15$	2.0005 (↓1.4504)	**4.7262** (↓3.4508)	**3.0258** (↓1.6754)	**3.2508** (↓2.1922)	**+40.28%**
Activation mask	Activation Noise	$\alpha = 0.20$	5.2763 (↓0.1501)	**18.8547** (↓3.7760)	**14.4536** (↓2.1005)	**12.8615** (↓2.0089)	**+13.51%**
Quantization mask	Quantization	W4A16	0.3001 (→0.0000)	**0.2251** (↓0.0750)	0.8252 (↓0.3751)	0.4501 (↓0.1501)	+25.01%
Combined-Score Top-4 Mask, Checkpoint-10							
Combined mask	Clean	–	**0.3751** (→0.0000)	**0.3001** (→0.0000)	**0.4501** (→0.0000)	**0.3751** (→0.0000)	0.00%
Combined mask	Weight Noise	$\sigma = 0.20$	0.5751 (↑0.0250)	**0.6252** (↓0.0250)	1.3503 (↓0.2501)	0.8502 (↓0.0834)	+8.93%
Combined mask	Weight Noise	$\sigma = 0.30$	1.0753 (↓0.1000)	2.3006 (↑0.2751)	3.0258 (↓0.6752)	2.1339 (↓0.1667)	+7.25%
Combined mask	Weight Noise	$\sigma = 0.35$	**1.5254** (↓0.1000)	3.7759 (↑0.0250)	**4.3011** (↓0.9002)	**3.2008** (↓0.3251)	**+9.22%**
Combined mask	Activation Noise	$\alpha = 0.05$	**0.3001** (↓0.1000)	**0.2251** (↓0.2251)	**0.4751** (↓0.3251)	**0.3334** (↓0.2167)	**+39.39%**
Combined mask	Activation Noise	$\alpha = 0.10$	**0.6252** (↓0.6002)	0.7752 (↓0.7752)	0.9752 (↓0.5751)	0.7919 (↓0.6502)	+45.09%
Combined mask	Activation Noise	$\alpha = 0.15$	**1.9505** (↓1.5004)	5.4514 (↓2.7257)	3.7509 (↓0.9502)	3.7176 (↓1.7254)	+31.70%
Combined mask	Activation Noise	$\alpha = 0.20$	**4.6512** (↓0.7752)	20.3801 (↓2.2506)	16.1040 (↓0.4501)	13.7118 (↓1.1586)	+7.79%
Combined mask	Quantization	W4A16	**0.2251** (↓0.0750)	0.3001 (→0.0000)	**0.6752** (↓0.5251)	**0.4001** (↓0.2001)	**+33.33%**