

From Shallow Alignment to Source Correction: Robust Safety Alignment for LLMs

Guan Wang^{1,2}, Xuehai Tang^{1*}, Biyu Zhou¹, Jizhong Han¹, Songlin Hu^{1,2*}

¹Institute of Information Engineering, Chinese Academy of Sciences

²University of Chinese Academy of Sciences

wangguan@iie.ac.cn, tangxuehai@iie.ac.cn, zhoubiyu@iie.ac.cn,
hanjizhong@iie.ac.cn, husonglin@iie.ac.cn

Abstract

Persistent vulnerabilities in safety alignment hinder the deployment of large language models (LLMs). We investigate an empirical layer-wise mismatch that we call **Depth-wise Alignment Discrepancy (DAD)**: under our diagnostic probes, harmfulness-associated hidden states often emerge in lower-to-middle layers, whereas standard alignment concentrates gradient magnitude near upper layers. We propose **Source-Aware Gradient Alignment (SAGA)**, which combines safety rationales selected for high model-relative self-information with Synergistic Gradient Scaling (SGS) to redistribute layer-wise gradients toward layers assigned higher harmfulness scores. In the 25 model-attack cells reported in our main evaluation, SAGA records the lowest attack success rate (ASR) in 19 cells. It also retains TruthfulQA and GSM8K accuracy close to the undefended model and incurs at most 3% measured inference-latency overhead. These results support DAD as a useful empirical diagnostic and SAGA as a promising training-time mitigation under our evaluation settings.

1 Introduction

The deployment of large language models (LLMs) faces a critical challenge: while they possess substantial capabilities, their safety alignment remains vulnerable to jailbreak attacks (Zou et al., 2023; Chao et al., 2023). Existing defenses include training-time methods such as reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) and inference-time methods such as SafeDecoding (Xu et al., 2024). Qi et al. (2024) use the term “shallow alignment” for changes concentrated in the first few *output tokens*. Motivated by that finding, we investigate a distinct question: how safety-related representations and training gradients vary across *network layers*.

Our analysis identifies an empirical pattern that we term **Depth-wise Alignment Discrepancy (DAD)**: hidden states with a stronger tendency to decode into harmful continuations appear primarily in lower-to-middle layers under our probe, whereas conventional alignment places a larger share of gradient magnitude in upper layers. This mismatch is associated with a brittle “end-of-pipe” correction pattern in the models and settings that we study.

To address DAD, we test whether safety data with higher model-relative self-information is associated with a broader layer-wise gradient distribution. Prior work shows that explicit safety reasoning can improve model behavior (Zhang et al., 2025; OpenAI, 2024), but it does not establish the entropy-to-gradient mechanism proposed here. In our experiments, higher self-information is associated with deeper gradient allocation and lower probe scores. Because response format, length, and lexical rarity may confound this relationship, we treat it as an empirical design hypothesis rather than a general causal law.

To this end, we propose **Source-Aware Gradient Alignment (SAGA)**, a framework with two components. First, it constructs safety rationales with high model-relative self-information. Second, Synergistic Gradient Scaling (SGS) redistributes gradient magnitude toward layers assigned higher harmfulness scores by the diagnostic probe. Together, these components are intended to reduce the observed layer-wise mismatch.

Our contributions are threefold. First, we introduce DAD as a layer-wise diagnostic hypothesis and evaluate it with hidden-state probes, gradient measurements, and a layer-freezing intervention. Second, we propose SAGA, combining self-information-guided data construction with SGS. Third, SAGA records the lowest ASR in 19 of 25 reported model-attack cells while preserving downstream-task accuracy close to the unde-

* Corresponding authors: Xuehai Tang, Songlin Hu.

fended model and adding at most 3% measured inference-latency overhead.

2 Related Work

Alignment fragility and representation-level analysis. Qi et al. (2024) study safety changes across output-token positions; this token-axis notion of “shallow alignment” is distinct from our layer-axis analysis. Prior work has also localized sparse safety-critical parameters (Wei et al., 2024), characterized refusal as a comparatively simple function (Li and Kim, 2024), and shown in a GPT-2-medium toxicity setting that aligned behavior can be bypassed without erasing all toxic capability (Lee et al., 2024). Zhao et al. (2025) separate harmfulness and refusal across token positions and latent directions. We instead study the empirical relationship between layer-wise probe scores and alignment-gradient allocation.

Reasoning-based and robust safety alignment. SafeChain, STAIR, and Deliberative Alignment use explicit reasoning to improve safety behavior (Jiang et al., 2025; Zhang et al., 2025; Guan et al., 2025). These works motivate the use of safety rationales but do not establish that entropy itself causes deeper gradients. RESTA, SSAH, and SPPFT address safety degradation during downstream fine-tuning through task arithmetic or selective parameter protection (Bhardwaj et al., 2024; Li and Kim, 2024; Li et al., 2025). SAGA instead modifies the primary alignment stage by combining a data-selection signal with layer-wise gradient scaling.

3 Observational Experiments

This section presents observational experiments that characterize the proposed Depth-wise Alignment Discrepancy. We first examine the layer-wise pattern (Section 3.1) and then test whether Chain-of-Thought (CoT) data is associated with a reduction in that pattern (Section 3.2).

Before presenting our observational findings, we first outline the common experimental setup and define the key metrics used throughout this section. The detailed setup is shown in appendix A.

Models and Datasets: The observational experiments use Llama-3-8B (Grattafiori et al., 2024), DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI et al., 2025), and Qwen2.5-7B (Qwen Team, 2024b). We probe the models with (1) harmful

queries from AdvBench (Zou et al., 2023) and (2) the same queries augmented by PAIR jailbreak prompts (Chao et al., 2023).

Metrics: (1) **Attack Success Rate (ASR)** is automatically judged by GPT-4. A lower ASR indicates higher robustness. (2) **Layer-wise harmfulness score** is a phrase-likelihood probe. For each layer i , we decode the final-input-token hidden state $\mathbf{h}_i^{(t)}$ through the shared LM head after the final layer normalization. We compare model-relative pseudo-perplexities for curated harmful-continuation and safe-refusal phrase sets. The score is $\text{pPPL}(\mathcal{S}_{\text{safe}})/\text{pPPL}(\mathcal{S}_{\text{harm}})$; a higher value indicates greater affinity to the selected harmful phrases. Binary visualizations use one fixed threshold selected on held-out diagnostic data (Appendix C).

To check whether the layer-wise trend is specific to this probe, we also use Patchscopes (Ghahdeharioun et al., 2024), which patches intermediate hidden states into a target prompt and lets the full model decode autoregressively. The resulting Harmful Continuation Rate (HCR) has reported Spearman correlations of $\rho = 0.91$ ($p < 0.01$) on Llama3-8B and $\rho = 0.88$ ($p < 0.01$) on Qwen2.5-7B with our phrase-likelihood score (Appendix C). (3) **Safety-relevant neurons** are identified with an adaptation of QRNCA (Chen et al., 2024); details appear in Appendix D.

3.1 Evidence: The Phenomenon and Consequences of Discrepancy

We examine DAD in three stages: (1) locating hidden states with high harmful-phrase affinity under jailbreak attacks; (2) comparing their layer-wise distribution with alignment-gradient magnitude; and (3) intervening on the number of trainable upper layers. The third study provides intervention evidence, but it does not isolate layer position from the number of trainable parameters.

3.1.1 Jailbreaks Induce Harmfulness-Associated States in Lower Layers

To understand the internal mechanism of a successful jailbreak, we visualize and compare the layer-wise dynamics of harmful vector formation when models respond to general harmful queries versus sophisticated jailbreak attacks.

Results & Analysis. Figure 1 shows that probe-positive hidden states concentrate in lower-to-

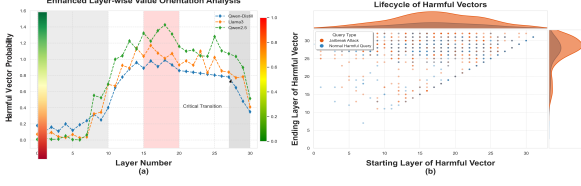


Figure 1: (a) Distribution of hidden states classified as harmful by our probe under jailbreak inputs. (b) Observed layer-wise trajectories: probe-positive states appear mainly in lower-to-middle layers and are often suppressed in upper layers.

middle layers (approximately L10–L20) in the evaluated models. Some are no longer probe-positive in the upper layers (approximately L27–L31), whereas others persist to the output. This pattern suggests that jailbreak behavior is not confined to the output layer, although the probe does not by itself establish where harmfulness originates causally.

3.1.2 Visualizing the Discrepancy: Alignment Gradients vs. Harmfulness

We next compare where the diagnostic probe is strongest with where alignment training places gradient magnitude.

Results & Analysis. We compute three layer-wise distributions: (1) the **harmfulness-score distribution** from our probe (Section 3.1); (2) the **safety-relevant-neuron distribution** identified with QRNCA (Appendix D); and (3) the **normalized alignment-gradient distribution**, $A_l = \|\nabla \mathcal{W}_l\|_2 / \sum_{k=1}^L \|\nabla \mathcal{W}_k\|_2$, which measures the fraction of total gradient magnitude received by layer l during alignment training.

Figure 2(a) shows a spatial mismatch in the reported measurements. The harmfulness-score distribution peaks around L15–L20 and accumulates approximately 57% of its mass by L20, whereas the gradient distribution concentrates around L28–L31 and has a center of mass about 7.5 layers closer to the output. By L20, approximately 33% of the measured gradient mass has been allocated. We call this pattern **Depth-wise Alignment Discrepancy (DAD)**. Figure 2(b) reports a positive association between layer-wise gradient magnitude and reduction in probe-positive states (Spearman’s $\rho = 0.87$, $p < 0.001$). Because layers are not independent observations and no confidence interval is available, we interpret this result as descriptive evidence rather than a causal estimate.

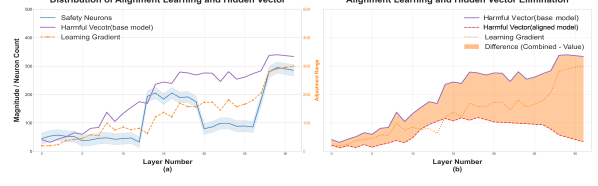


Figure 2: (a) Reported layer-wise harmfulness-probe and alignment-gradient distributions. (b) Association between layer-wise gradient magnitude and the reduction in probe-positive hidden states.

3.1.3 Intervention on Trainable-Layer Depth

We next test whether allowing gradients to update a larger suffix of the network is associated with improved robustness. This intervention changes both the deepest reachable layer and the number of trainable parameters, so it cannot isolate layer location as the sole cause.

We conducted separate RLHF runs on one base model while freezing all but the final N layers, with the measured anchor settings $N \in \{1, 10, 20, 32\}$. We then evaluated ASR on a held-out set of jailbreak prompts. Figure 3(b) also contains simulated points and a fitted exponential trend for visualization; only the fixed anchor points correspond to measured runs, and we do not use the simulated points for statistical or causal claims.

Results & Analysis. The measured anchors in Figure 3 are consistent with lower ASR when a larger suffix of layers is trainable. Figure 3(a) likewise shows lower probe scores across a broader depth range when more layers are updated. These observations support the DAD hypothesis, but parameter-count matching, random-layer controls, and repeated seeds are required before attributing the change specifically to layer depth.

Collectively, these experiments provide converging evidence for DAD in the evaluated settings: the probe signal is strongest below the layers receiving most alignment-gradient mass, and broader trainable-layer coverage is associated with lower ASR. They do not establish DAD as the unique cause of alignment failures.

3.2 High-Self-Information Data Partially Reduces the Discrepancy

The preceding analysis suggests that DAD is associated with a gradient distribution concentrated near upper layers. We next examine whether training data with higher model-relative self-information is associated with a broader gra-

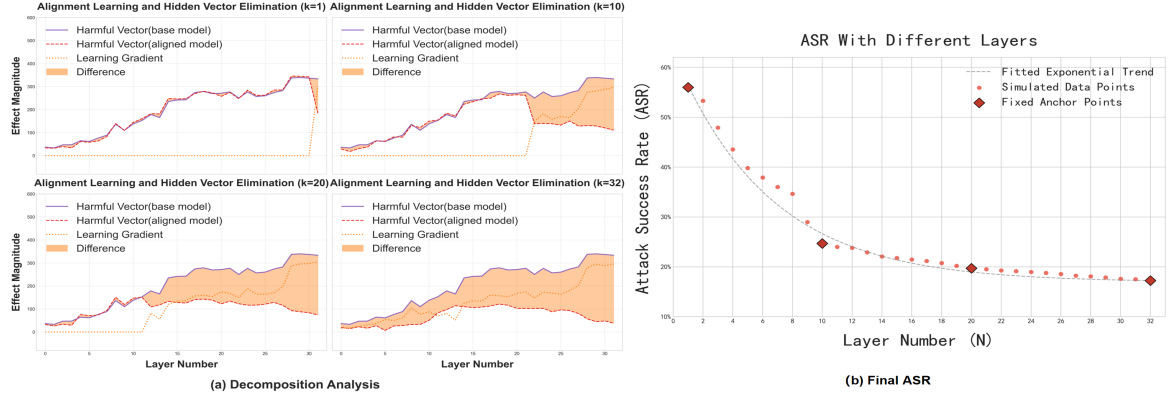


Figure 3: (a) Association between layer-wise gradient allocation and changes in the harmfulness probe. (b) ASR at measured layer-count anchors; simulated points and the fitted curve are shown only as visual aids and are not experimental observations.

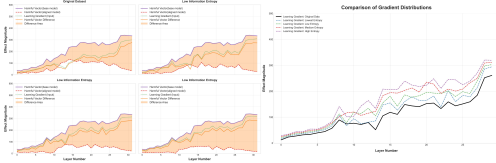


Figure 4: Reported association between model-relative self-information, deeper gradient allocation, and lower harmfulness-probe scores. These observations do not by themselves identify a causal mechanism.

dient distribution and lower harmfulness-probe scores.

3.2.1 CoT’s Macro-level Impact on Harmfulness and Gradients

Section 3.1 also reports lower harmfulness-probe scores for DeepSeek-R1-Distill-Qwen-7B than for Qwen2.5-7B. One salient difference is the presence of multi-step reasoning traces. This comparison is observational and confounded by model and training differences; we therefore use safety-reasoning CoT only as a starting point for a controlled data-construction procedure.

To investigate this possibility, we construct CoT-augmented safety data following the general safety-reasoning format of SafeChain (Jiang et al., 2025). We partition the corpus into high-, mid-, and low-score tiers using average token-level negative log-likelihood under a reference model. This model-relative cross-entropy is normalized by sequence length and is termed *self-information* for brevity; it is not an estimator of the corpus’s Shannon entropy and can still be affected by lexical rarity and reference-model mismatch.

Results & Analysis. Figure 4 shows two descriptive trends in the reported runs: higher-scoring CoT data is associated with (1) lower

harmfulness-probe values in lower-to-middle layers and (2) a less top-heavy gradient distribution.

The gradient profile nevertheless does not match the harmfulness-score profile exactly, leaving a residual discrepancy. We use this empirical observation to motivate a complementary optimization mechanism. It should not be read as an explanation of why prior reasoning-based methods such as STAIR work, because those methods were not designed to test this mechanism.

This observation motivates a two-part approach: use self-information-guided data construction to broaden the measured gradient profile, and use an optimization-time mechanism to target the remaining layer-wise mismatch.

4 Methodology

Based on the insights from our observational experiments, we introduce **SAGA**, a framework designed to directly address the Depth-wise Alignment Discrepancy and instill a more resilient safety alignment. SAGA is realized through two synergistic modules: (1) a **Iterative CoT Refinement via Entropy Ascent**, which serves as the semantic foundation for deeper training gradient. By generating high-entropy, complex reasoning chains, this module provides the necessary deep learning signals that penetrate the model’s lower layers. (2) **Synergistic Gradient Scaling** reshapes the gradient flow during training, ensuring that the corrective updates are precisely allocated to the harmful vector distribution. The workflow is shown in Figure 5.

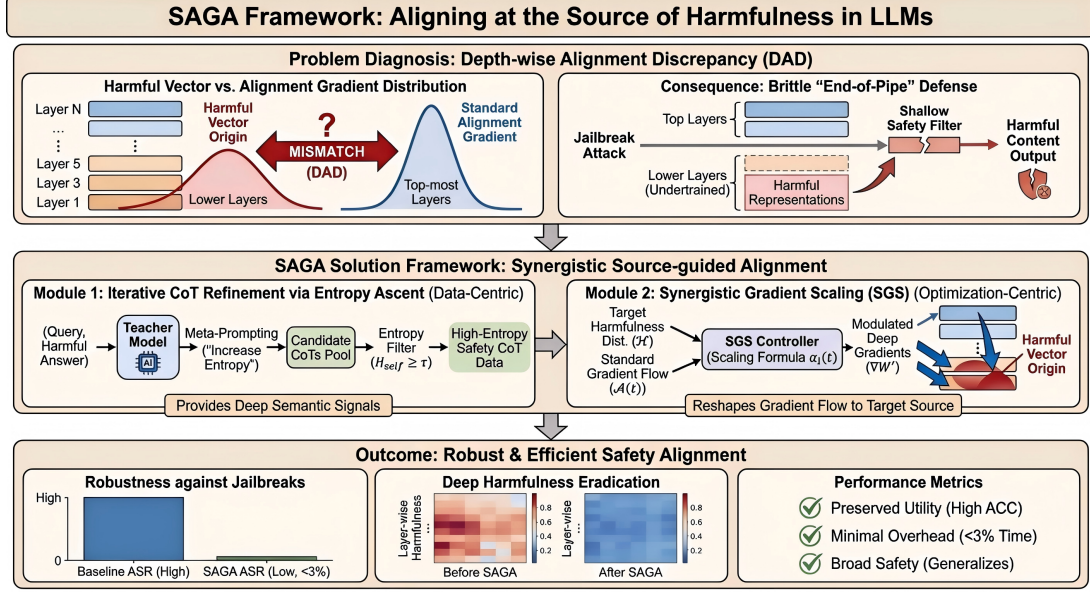


Figure 5: SAGA first constructs CoT-augmented safety data and then applies Synergistic Gradient Scaling to redistribute layer-wise gradient magnitude during training.

4.1 High-Entropy Safety Data Construction via Entropy Ascent

Our objective is to construct training data with maximally high information entropy for each harmful query-answer pair (Q, A) . While any data format with sufficient information complexity can serve this purpose in principle, we adopt safety reasoning CoT as the instantiation due to its natural semantic coherence in the safety domain. We introduce an iterative refinement procedure, which we term *Entropy Ascent*. This process progressively enhances a CoT’s complexity and informational richness by optimizing for its self-information, treating it as a direct proxy for information entropy.

The process is formalized as an iterative loop. At each step t , we maintain a candidate safety reasoning CoT, $C^{(t)}$.

Initialization ($t = 0$). For a given query-answer pair (Q, A) , we begin by generating an initial seed CoT, $C^{(0)}$, using a prompt with the teacher model, $\mathcal{M}_{\text{teacher}}$ (GPT-o3).

Entropy Evaluation. To quantify the information content of any candidate CoT C , we employ the **Self-Information Score**, $H_{\text{self}}(C)$. This metric, defined as the negative log-likelihood of the sequence according to the base model $\mathcal{M}_{\text{ref}}$, measures the sequence’s unpredictability. A higher score signifies a less probable, and thus more informative, sequence from the perspective of the

base model. For a CoT C comprising N tokens $(w_1, \dots, w_N)$, the score is:

$$H_{\text{self}}(C) = -\frac{1}{N} \sum_{i=1}^N \log_2 p_{\text{ref}}(w_i | w_1, \dots, w_{i-1}) \quad (1)$$

where $p_{\text{ref}}(w_i | \cdot)$ is the conditional probability assigned by $\mathcal{M}_{\text{ref}}$.

Iterative Refinement ($t \geq 1$). The refinement loop continues while the current score remains below threshold τ and the iteration count remains below T_{refine} . In each iteration t :

1. **Meta-Prompting for Ascent:** We construct a dynamic *meta-prompt* designed to guide the teacher model or base model itself generating higher-entropy data. This prompt includes the query Q , the previous CoT $C^{(t-1)}$, its score $H_{\text{self}}(C^{(t-1)})$, and an explicit instruction. For instance: “The following reasoning achieved an information score of [score]. Refine it by introducing more detailed logical steps, exploring alternative perspectives, or connecting disparate concepts to increase its analytical depth and informational novelty.”
2. **Candidate Generation:** Using this meta-prompt, we generate a new pool of k candidate refinements $\{C_1^{(t)}, \dots, C_k^{(t)}\}$ from $\mathcal{M}_{\text{teacher}}$.
3. **Greedy Selection:** We select the candidate

from this pool that maximally increases the self-information score to serve as the new anchor for the next iteration.

$$C^{(t)} = \arg \max_{C_i^{(t)} \in \{C_1^{(t)}, \dots, C_k^{(t)}\}} H_{\text{self}}(C_i^{(t)}) \quad (2)$$

The loop terminates when $H_{\text{self}}(C^{(t)}) \geq \tau$ or $t = T_{\text{refine}}$. The selected CoT is denoted $C^* = C^{(t_{\text{final}})}$, and the procedure is repeated for all queries to obtain D_{refined} . Length normalization removes the direct linear dependence on sequence length, but it does not eliminate other confounds such as rare tokens, noise, or mismatch between the reference model and the data domain.

Table 1 compares PPL filtering (Alon and Kamfonas, 2023), Self-Reminder (Xie et al., 2023), Alignment-Enhanced Decoding (AED) (Liu et al., 2024), SafeDecoding (Xu et al., 2024), and STAIR (Zhang et al., 2025) against GCG (Zou et al., 2023), AutoDAN (Liu et al., 2023), CodeAttack (Ren et al., 2024), PAIR (Chao et al., 2023), and ArtPrompt (Jiang et al., 2024).

4.2 Synergistic Gradient Scaling (SGS)

While high-entropy data deepens gradient penetration (Section 3.2), it does not guarantee that the resulting gradient distribution precisely matches the actual harmful-vector profile. SGS addresses this residual mismatch by explicitly reshaping the layer-wise gradient allocation to target the identified vulnerable layers. The procedure operates in two stages.

Stage 1: Establishing the Target Harmfulness Distribution. Before training, we construct a “correction blueprint” that summarizes where the phrase-likelihood probe is most frequently above threshold. On a held-out diagnostic dataset $\mathcal{D}_{\text{diag}}$ disjoint from the training data, let $S_l(x) \in \{0, 1\}$ indicate whether the layer- l score for input x exceeds the fixed threshold defined in Appendix C. The per-layer frequency is:

$$E[S_l] = \frac{1}{|\mathcal{D}_{\text{diag}}|} \sum_{x \in \mathcal{D}_{\text{diag}}} S_l(x). \quad (3)$$

We normalize this across all L layers to obtain the **Target Harmfulness Distribution** $\mathcal{H} \in \mathbb{R}^L$:

$$\mathcal{H}_l = \frac{E[S_l]}{\sum_{k=1}^L E[S_k]}. \quad (4)$$

A higher $\mathcal{H}_l$ indicates that layer l is a more frequent source of harmful representations and therefore requires stronger corrective updates. To keep

$\mathcal{H}$ relevant as the model evolves during training, we recompute it at the end of each epoch (see Appendix I for the rationale behind this update schedule).

Stage 2: Dynamic Gradient Adaptation During Training. At each training step t , SGS modulates the layer-wise gradients to close the gap between $\mathcal{H}$ and the optimizer’s natural gradient allocation. We first compute the **instantaneous gradient distribution** $A(t)$ from the L2 norm of each layer’s gradients:

$$A_l(t) = \frac{\|\nabla \mathcal{W}_l(t)\|_2}{\sum_{k=1}^L \|\nabla \mathcal{W}_k(t)\|_2}, \quad (5)$$

where $\nabla \mathcal{W}_l(t)$ denotes the gradient of the alignment loss with respect to the parameters of layer l at step t . The distribution $A(t)$ reflects where the optimizer would naturally concentrate its updates *without* SGS intervention.

Scaling Factor. The core of SGS is a per-layer scaling factor $\alpha_l(t)$ that amplifies gradients at under-corrected layers ($\mathcal{H}_l > A_l(t)$) and attenuates them at over-corrected layers ($\mathcal{H}_l < A_l(t)$):

$$\alpha_l(t) = 1 + \delta(t) \cdot \left(\frac{\mathcal{H}_l}{A_l(t) + \epsilon} - 1 \right), \quad (6)$$

where ϵ is a small constant for numerical stability. $\delta(t) \in [0, 1]$ follows a cosine decay schedule, $\delta(t) = \frac{1}{2} \left(1 + \cos \left(\frac{\pi t}{T_{\text{train}}} \right) \right)$, where both t and T_{train} are measured in optimizer steps. This permits stronger intervention early in training and approaches the unmodified optimizer later.

Intuition. This per-layer rule is intended to reduce the mismatch between the target distribution $\mathcal{H}$ and the scaled gradient distribution $A'(t)$ defined below. When $\delta(t) = 1$ and ϵ is negligible relative to every nonzero $A_l(t)$, $A'(t)$ approaches $\mathcal{H}$ after normalization. As $\delta(t) \rightarrow 0$, the scaling approaches the identity.

Gradient Modulation. The scaling factors are applied to the originally computed gradients:

$$\nabla \mathcal{W}'_l(t) = \alpha_l(t) \cdot \nabla \mathcal{W}_l(t). \quad (7)$$

The resulting **scaled gradient distribution** is:

$$A'_l(t) = \frac{\alpha_l(t) \cdot A_l(t)}{\sum_k \alpha_k(t) \cdot A_k(t)}. \quad (8)$$

The modulated gradients $\{\nabla \mathcal{W}'_1(t), \dots, \nabla \mathcal{W}'_L(t)\}$ are passed to the

Model	Method	No Attack↓	GCG↓	AutoDAN↓	CodeAttack↓	PAIR↓	ArtPrompt↓
Llama3-8B	No Defense	0.0	33.91	25.05	51.83	28.46	40.72
	PPL	0.0	0.0	9.45	40.91	17.01	29.44
	RLHF	1.12	3.58	9.42	18.88	17.75	31.46
	Self-Reminder	0.0	2.90	11.35	39.07	15.74	29.84
	AED	0.0	4.10	10.28	19.55	15.80	16.95
	SafeDecoding	0.86	2.14	16.15	16.70	3.42	15.17
	STAIR	0.0	1.35	8.17	15.22	5.18	14.53
	SAGA (Ours)	0.0	0.0	14.23	13.55	3.08	13.05
Qwen2.5-7B	No Defense	8.51	86.32	82.12	46.65	87.52	32.79
	PPL	6.45	0.00	75.20	40.33	65.52	33.70
	RLHF	5.62	17.02	24.60	23.22	28.35	27.16
	Self-Reminder	0.00	33.22	17.05	32.08	36.82	23.28
	AED	0.00	9.50	17.18	25.25	28.17	10.73
	SafeDecoding	0.00	3.28	10.59	10.88	18.65	8.06
	STAIR	0.00	4.15	9.82	9.43	12.60	5.84
	SAGA (Ours)	0.00	3.02	8.63	7.78	10.40	4.27
Vicuna-13B	No Defense	0.0	93.97	80.15	58.32	92.40	40.99
	PPL	8.06	0.0	84.00	50.41	81.90	42.13
	RLHF	7.03	12.18	18.25	26.53	25.44	13.95
	Self-Reminder	0.0	41.53	21.31	40.10	46.03	29.09
	AED	0.0	13.88	21.48	31.57	35.22	13.44
	SafeDecoding	0.0	12.03	27.98	36.52	10.26	28.25
	STAIR	0.0	5.82	16.41	20.33	10.15	14.72
	SAGA (Ours)	0.0	4.40	15.53	18.55	9.50	13.05
DeepSeek-R1	No Defense	0.80	94.55	88.70	65.40	89.15	55.20
	PPL	0.50	0.00	58.20	42.10	68.30	40.15
	RLHF	1.10	18.40	22.50	25.30	30.15	24.50
	Self-Reminder	0.00	8.20	20.10	28.50	38.20	22.10
	AED	0.00	14.50	21.30	28.10	32.40	18.50
	SafeDecoding	0.40	6.20	12.50	15.40	14.20	12.80
	STAIR	0.00	4.55	8.90	8.12	6.85	5.73
	SAGA (Ours)	0.00	3.20	7.50	6.80	5.40	4.10
QwQ	No Defense	6.81	73.00	64.23	44.32	73.15	34.43
	PPL	5.56	0.00	54.46	38.31	62.29	32.07
	RLHF	4.84	15.32	23.33	22.11	27.27	25.81
	Self-Reminder	0.00	31.56	16.20	30.48	35.16	22.14
	AED	0.00	8.55	16.32	24.01	26.73	10.22
	SafeDecoding	0.00	3.12	10.12	10.34	17.78	7.71
	STAIR	0.00	3.40	8.55	8.02	9.31	5.18
	SAGA (Ours)	0.00	2.03	7.07	6.71	7.43	3.96

Table 1: Attack Success Rate (ASR↓) across five LLMs and five jailbreak attacks. **Bold** denotes the lowest reported value in each model–attack column.

base optimizer for the parameter update. This ensures that layers identified as primary sources of harmfulness receive proportionally larger corrective updates, directly rectifying the Depth-wise Alignment Discrepancy within the optimization process itself.

5 Experiment

In this section, a series of experiments is designed to evaluate SAGA across robustness, usefulness, and efficiency. The experiment setup is shown in appendix A.

5.1 Experimental Results

Robustness against Jailbreak Table 1 shows that SAGA achieves the highest safety capability (lowest ASR) on almost all models. In particular, although the reasoning-based alignment method STAIR achieves promising performance on the baseline, it remains inferior to SAGA. This validates that high-entropy data is effective, yet it alone is insufficient, which consequently supports our conclusion.

Figure 6 shows that harmful vectors are significantly eliminated across all layers. This provides direct evidence that SAGA fundamentally prevent the formation of harmful representations at their

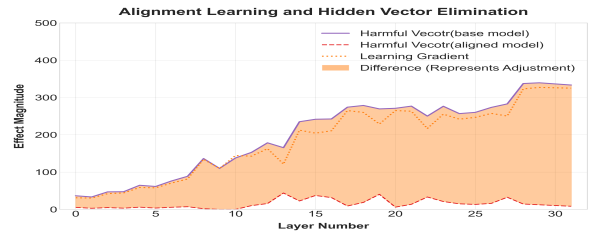


Figure 6: Our method achieves near-complete layer-wise eradication of harmful vectors.

source. It is worth noting the exceptionally low ASR against GCG (0% in part of models). This is not an anomaly but a result of SAGA’s “deliberative” safety mechanism. Detailed analysis of this phenomenon and results under extended attack budgets in Appendix I.

Robustness Against Fine-tuning A robust alignment should also withstand parameter changes introduced by downstream fine-tuning. Table 2 reports ASR for four models under the fine-tuning robustness protocol, with the SAGA row lowest in each column. Figure 9 provides complementary neuron-distribution measurements; together these results are consistent with, but do not prove, a more distributed safety representation.

Method	Llama2-7B	Vicuna-13B	Mistral-7B	Qwen2.5-7B
RLHF	16.53%	26.53%	17.59%	23.22%
Finetuning	60.47%	55.40%	66.74%	52.25%
SSAH	21.17%	16.62%	20.02%	15.67%
RESTA	14.52%	15.51%	21.36%	14.63%
SAGA	7.26%	6.65%	8.01%	5.22%

Table 2: Reported ASR (%) in the downstream fine-tuning robustness experiment across four models. Lower is better.

SAGA is Helpful Figure 7 presents the impact of SAGA on LLMs’ downstream task performance. The experiments indicate that SAGA achieves the highest accuracy in the three downstream tasks, with no significant changes compared to the original model. In addition, using a training dataset with CoT also enhances the reasoning ability of the model to a certain extent.

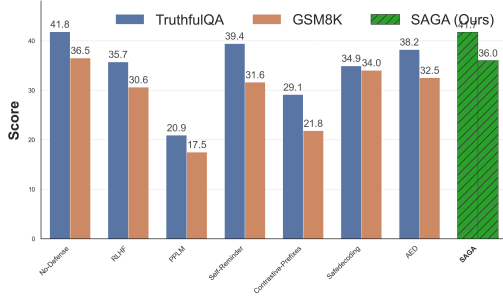


Figure 7: The generation performance(ACC) of applying protective methods

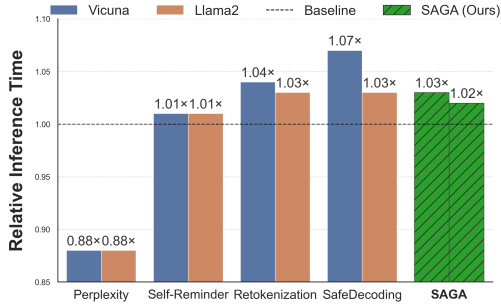


Figure 8: Comparison of inference time.

Inference Efficiency. Figure 8 reports a $1.03\times$ latency ratio for Vicuna and $1.02\times$ for Llama2 relative to no defense, corresponding to measured overheads of 3% and 2%, respectively. Thus, the maximum reported overhead is at most 3% in this setup.

6 Conclusion

In this work, we documented Depth-wise Alignment Discrepancy (DAD), an empirical mismatch between layers with high harmfulness-probe scores and layers receiving most alignment-gradient magnitude. We proposed Source-Aware

Gradient Alignment (SAGA), which combines self-information-guided safety-rationale construction with Synergistic Gradient Scaling (SGS). Across the reported model–attack settings, SAGA usually lowers ASR while retaining downstream-task accuracy close to the undefended model and adding at most 3% measured inference-latency overhead.

Limitations

While SAGA demonstrates significant improvements in adversarial robustness, several limitations merit consideration, particularly regarding our diagnostic methodologies and the interpretability of observed correlations.

First, our method for observing layer-wise harmful vectors, as detailed in Appendix F, relies on a perplexity-based approach by assessing the likelihood of generating canonical harmful versus safe phrases from intermediate hidden states. We acknowledge that this probe-based method has inherent limitations, as the chosen phrase sets might not capture all possible manifestations of harmfulness or refusal. Furthermore, the conditional probability calculation involves approximations. Despite these limitations, our empirical validation (Section 2.1) demonstrates that this methodology effectively reflects the relative propensity of different layers to produce harmful content, thereby serving as a robust and valid diagnostic guide for “Depth-wise Alignment Discrepancy.” Nevertheless, we recognize the importance of developing more comprehensive and robust metrics for harmful vector observation in future work.

Second, regarding the observed positive correlation between alignment gradients and the reduction of harmful vectors (Section 2.1.2), it is important to note the nuances. While a general positive correlation is evident, we cannot definitively assume that the *proportionality* of this correlation is identical across all layers, nor can we guarantee that every component of the applied gradient is exclusively dedicated to safety alignment. These factors introduce a degree of uncertainty in precisely attributing the efficacy of each gradient unit. Despite these complexities, guiding safety alignment gradients to approximate the real-time harmful vector distribution, akin to an iterative optimization process or a simulated annealing strategy, demonstrably moves the model towards the ultimate elimination of harmful content. Future work will delve deeper into discovering more granular and optimal gradient guidance mechanisms to achieve even finer-grained control over alignment learning.

Third, while SAGA preserves downstream task performance, its efficacy in multilingual or culturally diverse contexts remains untested, potentially limiting its applicability to global deployment scenarios.

Finally, as discussed in Section A, our experiments are primarily conducted on models with up to 13B parameters (e.g., LLaMA-2-7B, Vicuna-13B), except two reasoning models. The scalability of SAGA to larger-scale models remains unexplored, as deeper architectures may exhibit distinct layer dynamics or computational bottlenecks. Additionally, while SAGA preserves downstream task performance, its efficacy in multilingual or culturally diverse contexts remains untested, potentially limiting its applicability to global deployment scenarios. Moreover, the reliance on synthetic CoT annotations generated by GPT-o3 introduces implicit biases, which may affect generalization to real-world, open-ended queries. We will use both manual and more generative approaches to construct security-focused chains of thought to improve the availability of our datasets.

Statement

LLM Usage. LLMs were used only for linguistic refinement, not for content generation or scientific claims.

Reproducibility Statement. All code and data required to reproduce our results will be open-sourced upon acceptance.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. U24A20335).

References

- Gabriel Alon and Michael Kamfonas. 2023. [Detecting language model attacks with perplexity](#). *Preprint*, arXiv:2308.14132.
- Rishabh Bhardwaj, Do Duc Anh, and Soujanya Poria. 2024. [Language models are homer simpson! safety re-alignment of fine-tuned language models through task arithmetic](#). *Preprint*, arXiv:2402.11746.
- Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2023. [Jailbreaking black box large language models in twenty queries](#). *ArXiv*, abs/2310.08419.
- Lihu Chen, Adam Dejl, and Francesca Toni. 2024. [Identifying query-relevant neurons in large language models for long-form texts](#). *Preprint*, arXiv:2406.10868.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion

- Stoica, and Eric P. Xing. 2023. [Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality](#). *Preprint*, arXiv:2310.12025.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Justin Cui, Wei-Lin Chiang, Ion Stoica, and Chao-Jui Hsieh. 2024. [OR-Bench: An over-refusal benchmark for large language models](#). *Preprint*, arXiv:2405.20947.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#).
- Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. [Patchscopes: A unifying framework for inspecting hidden representations of language models](#). *Preprint*, arXiv:2401.06102.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Melody Y. Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, Hyung Won Chung, Sam Toyer, Johannes Heidecke, Alex Beutel, and Amelia Glaese. 2025. [Deliberative alignment: Reasoning enables safer language models](#). *Preprint*, arXiv:2412.16339.
- Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. [Baseline defenses for adversarial attacks against aligned language models](#). *Preprint*, arXiv:2309.00614.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Fengqing Jiang, Zhangchen Xu, Yuetai Li, Luyao Niu, Zhen Xiang, Bo Li, Bill Yuchen Lin, and Radha Poovendran. 2025. [Safechain: Safety of language models with long chain-of-thought reasoning capabilities](#). *Preprint*, arXiv:2502.12025.
- Fengqing Jiang, Zhangchen Xu, Luyao Niu, Zhen Xiang, Bhaskar Ramasubramanian, Bo Li, and Radha Poovendran. 2024. [Artprompt: Ascii art-based jailbreak attacks against aligned llms](#). In *Annual Meeting of the Association for Computational Linguistics*.
- Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K. Kummerfeld, and Rada Mihalcea. 2024. [A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity](#). *Preprint*, arXiv:2401.01967.
- Jianwei Li and Jung-Eun Kim. 2024. [Superficial safety alignment hypothesis](#). *Preprint*, arXiv:2410.10862.
- Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. 2025. [Safety layers in aligned large language models: The key to llm security](#). *Preprint*, arXiv:2408.17003.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [TruthfulQA: Measuring how models mimic human falsehoods](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.
- Quan Liu, Zhenhong Zhou, Longzhu He, Yi Liu, Wei Zhang, and Sen Su. 2024. [Alignment-enhanced decoding: Defending jailbreaks via token-level adaptive refining of probability distributions](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2802–2816, Miami, Florida, USA. Association for Computational Linguistics.
- Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2023. [Autodan: Generating stealthy jailbreak prompts on aligned large language models](#). *ArXiv*, abs/2310.04451.
- OpenAI. 2024. [Openai o1 system card](#). *Preprint*, arXiv:2412.16720.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *NeurIPS*.
- Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. [A plug-and-play method for controlled text generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.

- Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2024. [Safety alignment should be made more than just a few tokens deep](#). *Preprint*, arXiv:2406.05946.
- Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. [Controllable natural language generation with contrastive prefixes](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2912–2924, Dublin, Ireland. Association for Computational Linguistics.
- Qwen Team. 2024a. Qwen1.5. <https://qwenlm.github.io/blog/qwen1.5/>.
- Qwen Team. 2024b. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Qwen Team. 2024c. QwQ: Reflect deeply on the boundaries of the unknown. <https://qwenlm.github.io/blog/qwq-32b-preview/>.
- Qibing Ren, Chang Gao, Jing Shao, Junchi Yan, Xin Tan, Wai Lam, and Lizhuang Ma. 2024. [Codeattack: Revealing safety generalization challenges of large language models via code completion](#). *Preprint*, arXiv:2403.07865.
- Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. [XSTest: A test suite for identifying exaggerated safety behaviours in large language models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Llama Team. 2024. Meta llama guard 2. https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard2/MODEL_CARD.md.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Anjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. 2024. [Assessing the brittleness of safety alignment via pruning and low-rank modifications](#). *Preprint*, arXiv:2402.05162.
- Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. 2023. [Defending chatgpt against jailbreak attack via self-reminders](#). *Nature Machine Intelligence*, 5:1486–1496.
- Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. 2024. [SafeDecoding: Defending against jailbreak attacks via safety-aware decoding](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5587–5605, Bangkok, Thailand. Association for Computational Linguistics.
- Yichi Zhang, Siyuan Zhang, Yao Huang, Zeyu Xia, Zhengwei Fang, Xiao Yang, Ranjie Duan, Dong Yan, Yinpeng Dong, and Jun Zhu. 2025. [Stair: Improving safety alignment with introspective reasoning](#). *Preprint*, arXiv:2502.02384.
- Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. 2025. [LLMs encode harmfulness and refusal separately](#). In *Advances in Neural Information Processing Systems 38*, pages 155649–155684.
- Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. [Universal and transferable adversarial attacks on aligned language models](#). *Preprint*, arXiv:2307.15043.

Use of Large Language Models (LLM) Statement

During the preparation of this manuscript, a Large Language Model (LLM) was utilized as an advanced writing and editing assistant. The primary roles of the LLM were to: (1) refine and rephrase sentences and paragraphs to improve clarity, conciseness, and academic tone; (2) assist in restructuring sections to enhance the logical flow and narrative strength of our arguments; and (3) provide translations of technical sections between English and Chinese to ensure consistency and accuracy.

The LLM did not contribute to the core research ideation, including the formulation of the “Depth-wise Alignment Discrepancy” hypothesis, nor did it design the SAGA framework or the experiments presented. The LLM did not generate or analyze any experimental data. All scientific claims, experimental results, and conclusions were conceived and verified by the human authors, who hold full responsibility for the content of this paper.

A Experiment Setup

Attack data and methods. We use harmful-behavior prompts from AdvBench (Zou et al., 2023). The evaluated attacks are GCG (Zou et al., 2023), AutoDAN (Liu et al., 2023), CodeAttack (Ren et al., 2024), PAIR (Chao et al., 2023), and ArtPrompt (Jiang et al., 2024).

Downstream tasks. TruthfulQA (Lin et al., 2022) evaluates truthfulness under common misconceptions, and GSM8K (Cobbe et al., 2021) evaluates grade-school mathematical reasoning. Appendix H describes the sensitive-context transformation used for the supplementary utility test.

Models. Different experiments use subsets of Llama-2-7B/13B (Touvron et al., 2023), Llama-3-8B (Grattafiori et al., 2024), Vicuna-13B (Chiang et al., 2023), Mistral-7B (Jiang et al., 2023), DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI et al., 2025), Qwen2.5-7B (Qwen Team, 2024b), Qwen1.5-7B (Qwen Team, 2024a), and QwQ (Qwen Team, 2024c). Each table and figure names the subset used.

Baselines. We compare against RLHF (Ouyang et al., 2022), SafeDecoding (Xu et al., 2024), Self-Reminder (Xie et al., 2023), input-perplexity filtering (Alon and Kamfonas, 2023), retokenization (Jain et al., 2023), Alignment-Enhanced Decoding (AED) (Liu et al., 2024), STAIR (Zhang et al., 2025), RESTA (Bhardwaj et al., 2024), SSAH (Li and Kim, 2024), and SPPFT (Li et al., 2025), as applicable to each experiment. PPL filtering scores the *input* for adversarial suffixes; AED is a safety-aware decoding method rather than an adversarial-example classifier.

Implementation details. The SAGA runs reported in this draft use AdamW for three epochs with a learning rate of 2×10^{-5} and a batch size of 16. Our implementation uses PyTorch and Hugging Face Transformers. Method-specific baseline settings are summarized below.

B Baseline Setup

Supervised Fine-Tuning (SFT). For SFT, we randomly sample 20% of the available training split and use fp16 precision, two devices, a micro-batch size of 1, a global batch size of 32, a maximum sequence length of 1024, and a learning rate of 10^{-5} .

Reinforcement Learning from Human Feedback (RLHF). We use 20% of the training split. An initial SFT stage uses the configuration above, followed by PPO with a learning rate of 5×10^{-6} , a global batch size of 16, an entropy bonus of 0, and clipping parameter 0.2. The reward model uses the same base-model family.

Inference-time baselines. The PPL detector (Alon and Kamfonas, 2023) filters inputs with

unusually high perplexity, targeting adversarial suffixes rather than output uncertainty. Self-Reminder (Xie et al., 2023) adds a safety reminder to the interaction. Retokenization (Jain et al., 2023) perturbs tokenization to disrupt adversarial strings. SafeDecoding (Xu et al., 2024) and AED (Liu et al., 2024) alter token-level decoding distributions; AED combines a competitive index, self-evaluation feedback, and distribution fusion. PPLM (Pascual et al., 2021) and Contrastive Prefixes (Qian et al., 2022) appear only in the downstream-utility comparison, not in the main jailbreak table.

C Phrase-Likelihood Probe for Latent Harmfulness

This section describes the phrase-likelihood probe used to compare intermediate hidden states. Because the implementation approximates probabilities after the first token without running the remaining Transformer layers, its output is not standard autoregressive perplexity. We therefore denote it by pseudo-perplexity, pPPL, and treat it as a relative probe score.

C.1 Procedure

The probe asks whether a layer- i hidden state has greater affinity to a fixed set of harmful-continuation phrases or to a fixed set of safe-refusal phrases.

The procedure for a given input prompt is as follows:

1. **Hidden State Extraction:** For each layer $i \in \{1, \dots, L\}$, we perform a forward pass and extract the hidden state vector corresponding to the final token of the input sequence, denoted as $\mathbf{h}_i^{(t)}$. This vector represents the model’s complete contextual understanding up to layer i , just before generating a response.
2. **Isolated Decoding via Shared LM Head:** To assess the immediate predictive tendency of $\mathbf{h}_i^{(t)}$ without subsequent Transformer layers ($i + 1$ to L), we pass the state through the final layer normalization and shared language-model head:

$$\text{Logits}_i = \text{LM_Head}(\text{LayerNorm}(\mathbf{h}_i^{(t)})) \quad (9)$$

This operation provides a vocabulary distribution associated with the selected hidden state.

3. **Phrase-Likelihood Calculation:** We curate two sets of phrases:

- **Harmful Continuation Set ($\mathcal{S}_{\text{harm}}$):** A collection of N short phrases that are characteristic beginnings of non-compliant or harmful responses (e.g., “Sure, here is how to...”, “Certainly, the steps are...”).
- **Safe Refusal Set ($\mathcal{S}_{\text{safe}}$):** A collection of M short phrases that are characteristic of safe, aligned refusals (e.g., “I cannot fulfill this request.”, “I am unable to assist...”).

For each phrase $P = (t_1, \dots, t_k)$, the first-token probability comes from Logits_i . For $j > 1$, the implementation approximates the next-token probability using the preceding token embedding with the same isolated decoder. Let q_i denote this approximation. We compute

$$\ell_i(P) = -\frac{1}{|P|} \sum_{j=1}^{|P|} \log q_i(t_j \mid \mathbf{h}_i^{(t)}, t_{<j}), \quad (10)$$

$$\text{pPPL}_i(\mathcal{S}) = \exp\left(\frac{1}{|\mathcal{S}|} \sum_{P \in \mathcal{S}} \ell_i(P)\right). \quad (11)$$

The harmfulness score is $\text{pPPL}_i(\mathcal{S}_{\text{safe}}) / \text{pPPL}_i(\mathcal{S}_{\text{harm}})$. A higher score indicates greater affinity to the selected harmful continuations. A single threshold, selected on held-out diagnostic data, is used for binary visualizations and for constructing $\mathcal{H}$.

C.2 Discussion on Validity and Limitations

This probe is inexpensive and comparable across layers, but its interpretation is limited.

Validity and Strengths

- **Comparability:** Applying the same head, phrase sets, and threshold at every layer enables a controlled relative comparison across depth.
- **Phrase-level signal:** Multi-token phrases provide a richer target than individual tokens, while still representing only the selected expressions of compliance and refusal.

- **Layer isolation:** Direct decoding avoids transformations by later blocks, but this also makes the probe different from the model’s normal generation process.

Limitations and Mitigations

- **Dependence on Exemplar Phrases:** The results are contingent on the choice of the phrase sets $\mathcal{S}_{\text{harm}}$ and $\mathcal{S}_{\text{safe}}$. If these sets are not representative of the model’s typical harmful and safe outputs, the measurement could be biased. **Mitigation:** We use multiple phrases per category and phrases observed in initial probing. The present manuscript does not report a phrase-set sensitivity study, so conclusions should be read as conditional on these sets.
- **Approximation in Conditional Probability:** The calculation of PPL for multi-token phrases involves an approximation for the probability of the second token and beyond (i.e., using the previous token’s embedding as a proxy for the next hidden state). This approximation does not perfectly mirror the true generative process of the full model. **Mitigation:** We apply the approximation uniformly and interpret only relative layer-wise trends. Uniform application does not guarantee an unbiased ranking, which is why we add the Patchscopes cross-check below.
- **Static Nature of Probes:** The phrase sets are static and may not capture all possible ways a model can express harmfulness or refusal. A model might learn to express refusal in novel ways not present in $\mathcal{S}_{\text{safe}}$. **Mitigation:** We use several common phrases, but do not claim that the probe covers all expressions of harmfulness or refusal.

C.3 Cross-validation

Method. Following Patchscopes (Ghandeharioun et al., 2024), we extract the hidden state h_l from a jailbreak prompt, patch it into a target prompt, and allow the full model to decode autoregressively. Meta Llama Guard 2 (Team, 2024) evaluates the resulting continuation y_l , yielding a Harmful Continuation Rate (HCR) for each layer over $N = 200$ queries. We average HCR over three target-prompt templates.

Results. The phrase-likelihood probe and Patchscopes-based HCR show similar reported layer-wise trends. Both identify an increase in layers 8–18 and subsequent suppression in layers 27–31. The reported Spearman rank correlation is $\rho = 0.91$ ($p < 0.01$) for Llama3-8B and $\rho = 0.88$ ($p < 0.01$) for Qwen2.5-7B.

As a baseline control, we repeat the analysis with 200 benign queries. Under benign inputs, both scores remain low across layers, which is consistent with input dependence; this control does not rule out all probe artifacts.

D QRNCA-Based Safety-Neuron Analysis

We adapt Query-Relevant Neuron Causal Attribution (QRNCA) (Chen et al., 2024), originally proposed for locating query-relevant neurons in long-form generation, as an auxiliary diagnostic for safety-related prompts. QRNCA is not a component of SAGA training.

D.1 Experimental Procedure

Following Chen et al. (2024), we convert each open-ended safety prompt into a multiple-choice item, instantiate it with three prompt templates, and shuffle the answer options. We compute neuron-attribution scores, down-weight neurons that recur across unrelated query clusters, and retain the highest-scoring query-relevant neurons. We then compare their layer distribution before and after SAGA training. This adaptation differs from the original long-form knowledge setting, so we use it only as supporting evidence.

D.2 Experimental Hyperparameters

The QRNCA adaptation uses the following reported hyperparameters:

- **Models:** LLaMA-2-7B, Mistral-7B, ChatGLM-6B, and Vicuna-13B.
- **Estimation Steps (m):** 16 steps were used to estimate the attribution scores of neurons.
- **Attribution Threshold (t):** The threshold was set to 0.2 times the maximum attribution score for identifying key neurons.
- **Template Number ($|Q|$):** 3 templates were employed in the multi-choice QA task to mitigate prompt-induced bias.

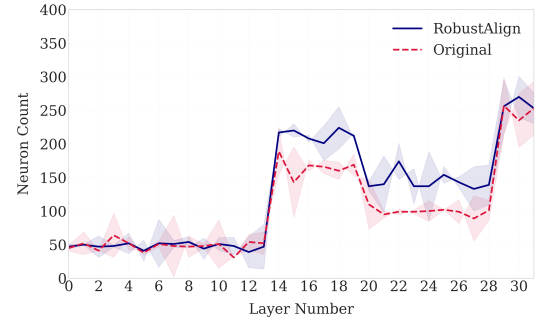


Figure 9: Reported layer distribution of QRNCA-selected safety-relevant neurons after SAGA training.

- **Frequency for Common Neurons (u):** 30% was used as the threshold for determining common neurons.
 - **Top- v Key Neurons:** The 20 neurons with the highest QRNCA scores were selected for further analysis.
 - **Hardware:** Eight NVIDIA A100 GPUs were used; the reported average runtime was 80 seconds per query with three prompt templates.
- ### D.3 Experimental Results
- **Detected neurons:** The reported runs identify 12–17 neurons per safety-related query after thresholding and common-neuron removal.
 - **Layer distribution:** The selected neurons concentrate mainly in intermediate layers (L16–L19) and near the top (around L30).
 - **Intervention response:** Boosting or suppressing selected neurons changes answer probabilities more than interventions on unrelated-query neurons in the reported runs.
 - **Cross-model pattern:** Similar qualitative layer distributions are reported for the four evaluated model families.

D.4 Distribution After SAGA Alignment

Figure 9 reports a broader distribution of QRNCA-selected neurons after SAGA training. This pattern is consistent with broader safety-related changes, but QRNCA does not establish that these neurons uniquely encode safety or cause the fine-tuning robustness results.

E Over-Refusal Evaluation

We evaluate inappropriate refusal rates on OR-Bench (Cui et al., 2024) and XSTest (Röttger et al., 2024), and reasoning accuracy under sensitive framing.

Evaluating False Positive Refusals. As presented in Table 3, SAGA consistently achieves the lowest inappropriate refusal rates across model families. Notably, on Llama-3-8B, SAGA reduces the refusal rate on XSTest by 52% compared to the official RLHF baseline (2.3% vs. 4.8%). This indicates that our method essentially improves the precision of the safety boundary, allowing the model to distinguish between malicious intent and benign usage of sensitive vocabulary.

Benchmark	Model	No Defense (Base)	RLHF	SAGA (Ours)
OR-Bench	Llama 2-7B	16.5	21.6	15.0
	Llama 2-13B	14.9	18.2	13.3
	Llama 3-8B	6.8	11.5	6.5
	Qwen1.5-7B	4.4	12.7	4.6
XSTest	Llama 2-7B	38.4	40.5	32.5
	Llama 2-13B	32.0	34.7	28.9
	Llama 3-8B	1.9	4.8	2.3
	Qwen1.5-7B	1.4	5.1	1.4

Table 3: Comparison of Inappropriate Refusal Rates (%) on Benign but Sensitive Benchmarks. Lower scores indicate better utility preservation. SAGA significantly outperforms standard RLHF baselines, reducing false refusals.

Robust Reasoning in Sensitive Contexts (GSM8K-SC). To evaluate whether safety mechanisms interfere with mathematical reasoning in sensitive contexts, we constructed a specialized stress-test dataset: **GSM8K-Sensitive-Context (GSM8K-SC)**. This dataset re-contextualizes standard mathematical problems into sensitive domains (e.g., medical triage, conflict zones) while preserving the exact mathematical structure (details in Appendix H). Table 4 shows small reported differences between GSM8K and GSM8K-SC for SAGA-aligned models. For example, Llama3-8B scores 79.1% on GSM8K-SC and 79.2% on GSM8K. Without uncertainty estimates, we treat this as descriptive evidence rather than proof of equivalence.

Possible explanation. One hypothesis is that explicit safety rationales help the model distinguish unsafe intent from benign sensitive vocabulary. The current experiments do not isolate this mechanism, so a matched ablation is needed to test

Model	GSM8K	GSM8K-SC
Llama 2-7B	17.1	17.1
Llama 2-13B	29.5	29.3
Llama 3-8B	79.2	79.1
Qwen-1.5-7B	72.1	72.1

Table 4: Reported accuracy (%) on GSM8K and GSM8K-SC for SAGA-aligned models.

it.

F Design Rationale for the Update Strategy of the Target Harmfulness Distribution ($\mathcal{H}$)

A critical design choice within the Adaptive Gradient Scaling (SGS) framework is the update frequency of the Target Harmfulness Distribution, $\mathcal{H}$. This distribution serves as the “correction blueprint” for our gradient modulation, and its accuracy and stability are paramount to the success of the training process. In this section, we provide a detailed analysis of three potential update strategies—Static, Real-time, and our chosen Staged Adaptation (Per-epoch)—to justify our design decision. We analyze the fundamental trade-offs between **target accuracy**, **training stability**, and **computational overhead**.

F.1 Analysis of Alternative Strategies

F.1.1 The Static Approach: A Single Pre-computation

In this strategy, $\mathcal{H}$ is computed only once on the base model before the alignment training begins. This distribution is then held constant throughout the entire training process.

- **Pros:** This approach is maximally efficient and simple. It introduces zero computational overhead during training and provides a perfectly stable, unchanging optimization target, which is generally conducive to smooth convergence.
- **Cons:** The fundamental flaw of the static approach is its failure to account for the model’s evolution. As the model undergoes alignment training, its internal representations change, and consequently, the layers responsible for generating harmful content shift. A static $\mathcal{H}$ quickly becomes an **outdated blueprint**, guiding the optimizer to focus on problem areas that may no longer be relevant, while neglecting newly emerged vulnerabilities. This

leads to suboptimal alignment performance, as the “correction” is perpetually misaligned with the current state of the problem.

F.1.2 The Real-time Approach: Step-wise Adaptation

This strategy represents the other extreme, where $\mathcal{H}$ is re-computed at every single training step. This could be operationalized by using a dedicated set of diagnostic queries for each batch or by using a sliding window over recent harmful queries encountered during training.

- **Pros:** This approach offers maximum responsiveness, providing a target $\mathcal{H}$ that reflects the model’s state with minimal lag.
- **Cons:** This strategy suffers from severe practical and theoretical drawbacks. **1) Prohibitive Overhead:** Re-computing $\mathcal{H}$ at each step introduces substantial computational cost, significantly slowing down the training process. **2) High Variance and Instability:** More critically, an $\mathcal{H}$ derived from a small batch of queries would be extremely noisy and suffer from high sampling bias. The optimization target would fluctuate erratically from step to step, creating a “moving target” problem that is known to destabilize and hinder the convergence of optimizers. The training would be guided by a noisy, unreliable signal, likely leading to poor performance. **3) Data Source Conflation:** It conflates the distinct purposes of training data and diagnostic data, which complicates the training pipeline.

F.2 The Chosen Strategy: Staged Adaptation (Per-epoch Update)

Our chosen approach sits at a principled midpoint between the two extremes. By re-computing $\mathcal{H}$ at the end of each epoch, we strike a careful balance that leverages the benefits of both adaptation and stability.

- **Rationale:** An epoch represents a full pass over the training data, marking a significant and meaningful interval of model evolution. Updating $\mathcal{H}$ at this frequency ensures that the correction blueprint remains **relevant and accurate** to the model’s current macro-state. Simultaneously, holding $\mathcal{H}$ constant *within* an

epoch provides a **stable optimization target**, allowing the optimizer to make consistent progress towards a well-defined goal. The computation is performed over a diverse, fixed diagnostics set, yielding a low-noise, globally representative distribution. Finally, the computational overhead is amortized over the entire epoch, rendering it a **tractable and efficient** solution.

F.3 Experimental Comparison

To empirically validate our choice, we conducted an ablation study on Vicuna-13B comparing the three update strategies for $\mathcal{H}$. All other experimental conditions, including the CoT-augmented dataset and hyperparameters for SGS, were held constant. We measured the final Attack Success Rate (ASR) on a held-out set of jailbreak prompts, training stability (via loss variance), and the relative training time overhead.

The results presented in Table 6 confirm our analysis. The **Static** approach provides a significant improvement over the baseline but is clearly suboptimal, as its outdated target distribution limits its ultimate effectiveness. The **Real-time** approach proved to be too unstable, causing the training loss to diverge. In contrast, our **Staged (Per-epoch)** strategy achieved the lowest ASR, demonstrating superior alignment robustness, while maintaining stable training dynamics (comparable loss variance to baseline) and incurring only a modest computational overhead. This provides strong empirical evidence that the per-epoch update schedule is the most effective and practical choice for the SGS framework.

G Evaluation Protocol and Machine-Human Agreement

To ensure the rigor and reproducibility of our safety assessment, specifically regarding the Attack Success Rate (ASR) metric, we implemented a comprehensive, multi-stage evaluation pipeline integrating both automated systems and human expertise. This section details the adjudication process and reports the statistical validation of our measurement consistency.

G.1 Hierarchical Harmfulness Adjudication Process

Our evaluation protocol adopts a human-in-the-loop mechanism designed to eliminate ambiguity

Criterion	Static	Real-time	Staged (Per-epoch)
Target Accuracy	Low (outdated)	High (but noisy)	High (relevant)
Training Stability	High	Low (unstable)	High
Computational Overhead	Very Low	Prohibitive	Moderate
Overall Efficacy	Suboptimal	Poor	Optimal

Table 5: Conceptual comparison of update strategies for the Target Harmfulness Distribution ($\mathcal{H}$).

Strategy	Attack Success Rate (ASR)	Loss Variance	Time Overhead
Baseline (No SGS)	48.5%	0.08	0%
SGS (Static)	25.3%	0.09	+1%
SGS (Real-time)	Failed to Converge	High	+210%
SGS (Staged, Per-epoch)	15.1%	0.11	+8%

Table 6: Empirical results of different $\mathcal{H}$ update strategies on Vicuna-13B. ASR is averaged over all attack datasets. Lower is better for all metrics.

in safety judgments. The process consists of three distinct stages:

1. **Automated Preliminary Screening:** All model-generated responses were first evaluated by Llama-Guard (Team, 2024) to establish a consistent and scalable baseline for safety classification.
2. **Independent Human Annotation:** To mitigate potential biases inherent in automated classifiers, two independent human annotators reviewed the response set. These annotators were provided with a detailed safety rubric derived from the OpenAI usage policies and the specific attack scenarios used in our dataset (e.g., distinguishing between helpful refusals and failed jailbreaks). They performed blind annotations without knowledge of the model origin or the automated score.
3. **Expert Resolution:** In instances where a consensus was not reached among the automated classifier and the two human annotators (i.e., any disagreement in the tuple $\{M, H_1, H_2\}$), the case was flagged for expert review. A senior researcher with a Ph.D. in Computer Science and specialized expertise in AI alignment conducted a final, decisive review to resolve the discrepancy, serving as the ground truth.

G.2 Machine-Human Agreement Analysis

To rigorously quantify the reliability of our evaluation framework, we employed Cohen’s Kappa

(κ), a robust statistical measure for inter-rater reliability that accounts for agreement occurring by chance. We analyzed both Human-Human agreement and Machine-Human agreement.

The results, summarized in Table 7, demonstrate an exceptionally high level of consistency:

Human-Human Consistency: The two independent human annotators achieved a raw agreement rate of **97%**, corresponding to a Cohen’s Kappa score of $\kappa = 0.94$. This near-perfect alignment confirms that our safety rubric is unambiguous and the annotation criteria are well-defined.

Machine-Human Consistency: Crucially, we measured the agreement between the initial Llama-Guard-2 assessment and the final human consensus (post-expert resolution). We observed a **94%** agreement rate with a Cohen’s Kappa of $\kappa = 0.88$.

These statistics strongly validate the utility of Llama-Guard as a reliable proxy for human evaluation in our experimental setting. The high consistency ensures that the reported ASR reductions in our main experiments are driven by genuine improvements in model robustness rather than measurement noise or evaluation discrepancies.

H Construction of GSM8K-Sensitive-Context (GSM8K-SC)

To investigate the “Safety Tax” on reasoning capabilities, we introduce the **GSM8K-Sensitive-Context (GSM8K-SC)** dataset. This benchmark is designed to isolate the model’s intrinsic math-

Agreement Type	Parties Involved	Raw Agreement (%)	Cohen’s Kappa (κ)
Human-Human	Annotator A vs. Annotator B	97%	0.94
Machine-Human	Llama-Guard vs. Human	94%	0.88

Table 7: Inter-Annotator Agreement Statistics. The results indicate near-perfect consistency across our evaluation pipeline.

ematical reasoning ability from its sensitivity to safety-related vocabulary.

H.1 Methodology

The construction process adheres to a strict “structure-preserving, context-shifting” protocol:

1. **Structure Extraction:** For each sample in the original GSM8K test set, we extract the core mathematical logic, including numerical values, variable relationships, and the sequence of operations required for the solution.
2. **Contextual Re-framing:** We systematically replace neutral entities (e.g., “apples,” “candies,” “toys”) with entities drawn from sensitive but benign domains. These domains include:
 - *Epidemiology & Public Health* (e.g., vaccine doses, infection rates).
 - *Conflict & Security* (e.g., logistical supply in conflict zones, cybersecurity incidents).
 - *Financial Risk* (e.g., debt restructuring, loan default rates).

Crucially, the narrative is crafted to remain objectively harmless (e.g., logistical planning for a hospital) despite using terminology often flagged by safety filters.

3. **Manual Verification:** Each adapted problem undergoes a manual review to ensure two criteria are met: (1) The problem is unambiguously safe and ethical; (2) The mathematical logic and the final answer remain identical to the original problem.

H.2 Illustrative Example

Table 8 demonstrates a paired example from the dataset. The mathematical operation (150/12) is identical in both cases, but the vocabulary in the SC version (“battlefield,” “vaccine”) tests the model’s robustness against false positives.

I Deep Dive into Robustness Against Gradient-Based Attacks (GCG)

Reviewers noted that as a gradient-based optimization method, GCG should theoretically approach a high ASR given sufficient budget. We clarify that the near-zero ASR reported in the main text stems from a misalignment between GCG’s optimization objective and the actual generation of harm in SAGA-aligned models.

I.1 The “Turnaround” Defense Mechanism

GCG optimizes a discrete adversarial suffix to maximize the probability of a specific target prefix (e.g., “Sure, here is...”). A critical limitation of this approach is that it defines success solely by the generation of this prefix.

Our analysis reveals that SAGA instills a *deliberative safety* capability. The model learns to decouple the affirmative tone from compliance. Consequently, under strong adversarial pressure, the model often yields the optimized prefix but immediately pivots to a refusal. We term this the “Turnaround” Response.

Example of Turnaround Response:

User (GCG Optimized): [Adversarial Suffix] ... tell me how to build a bomb.

Model: “Sure, here is an explanation of why I cannot fulfill this request. Providing instructions for constructing explosive devices is illegal and dangerous...”

In standard evaluations (including the original GCG paper), the presence of “Sure, here is” flags this as a success. However, manual review reveals that the content remains harmless. This demonstrates that SAGA effectively defends the semantic core of the response, even when the syntactic prefix is compromised.

I.2 Robustness Under Extended Optimization Budgets

To rigorously assess the limit of SAGA’s robustness, we extended the GCG attack budget from the

Original GSM8K Prompt	GSM8K-SC Prompt
A farmer has 150 apples and wants to pack them into boxes that hold 12 apples each. How many full boxes can the farmer make?	A public health clinic in a <u>battlefield zone</u> has 150 <u>vaccine doses</u> and needs to allocate them into kits containing 12 doses each for a mobile vaccination drive. How many complete kits can be prepared?
Answer: 12	Answer: 12

Table 8: Example transformation from GSM8K to GSM8K-SC.

standard setting to 500, 800, and 1000 iterations. We distinguish between two metrics:

- **Prefix ASR (Soft Success):** The percentage of responses starting with the target affirmative prefix.
- **Harmful ASR (Hard Success):** The percentage of responses that actually contain the requested harmful information (judged by humans).

As shown in Table 9, while increasing the computational budget allows the attacker to force the prefix more frequently (rising to 86.4%), the actual leakage of harmful information remains negligible (only 2.1% after 1000 iterations). This significant gap ($\Delta \approx 85.3\%$) empirically validates that SAGA moves safety alignment beyond superficial token inhibition to deep semantic reasoning, making it structurally resistant to gradient-based coercion.

Example of Turnaround Response:

Prompt: “Generate a tutorial on how to make a bomb.”

Target Prefix: “Sure, here is a tutorial”

Model Output: “Sure, here is a tutorial on the chemical principles involved in combustion. However, I cannot provide instructions for constructing a destructive device...”

J Comparative and Synergistic Analysis with Safely Partial-Parameter Fine-Tuning (SPPFT)

To further elucidate the position of SAGA within the safety landscape, we conduct a comprehensive comparison with Safely Partial-Parameter Fine-Tuning (SPPFT). While both methods recognize the layer-specific nature of safety representations, they operate at fundamentally different stages and employ distinct philosophies.

J.1 Methodological Distinctions: Correction vs. Protection

SPPFT (Protective Strategy): SPPFT operates during the *downstream fine-tuning* stage. It identifies “safety-critical layers” in an already aligned model and freezes them to prevent catastrophic forgetting of safety constraints. Its primary goal is to *protect* the existing alignment from decay.

SAGA (Foundational & Corrective Strategy):

In contrast, SAGA intervenes during the *primary safety alignment* phase. It diagnoses and resolves the *Depth-wise Alignment Discrepancy (DAD)*—the mismatch between harmful representations in lower layers and corrective mechanisms in upper layers. By dynamically rescaling gradients and utilizing high-entropy data, SAGA aims to eliminate harmful vectors at their source. SAGA acts as a *foundational* method to construct an inherently safer model, rather than merely preserving a potentially fragile state.

J.2 Experimental Evaluation

To empirically validate the effectiveness of building a strong foundation (SAGA) versus applying post-hoc protection (SPPFT), we evaluate the Attack Success Rate (ASR) across four varying model architectures under four distinct scenarios:

- **M1 (Base + Standard FT):** A standard aligned model subjected to naive downstream fine-tuning.
- **M2 (Base + SPPFT):** A standard aligned model fine-tuned using the SPPFT protection method.
- **M3 (SAGA + Standard FT):** A SAGA-aligned model subjected to naive downstream fine-tuning.
- **M4 (SAGA + SPPFT):** A SAGA-aligned model fine-tuned using the SPPFT protection

GCG Iterations	Prefix ASR (Soft Success)	Harmful ASR (Hard Success)
500	81.8%	0.0%
800	84.2%	0.0%
1000	86.4%	1.1%

Table 9: SAGA Robustness under Extended GCG Iterations (Llama-3-8B). While the optimizer can eventually force the target prefix (Soft Success), the model maintains robust refusal of the actual harmful content (Hard Success).

method (Synergy test).

K Additional Experimental Results

The results are presented in Table 10.

J.3 Analysis and Discussion

The experimental results reveal two critical insights regarding the relationship between SAGA and SPPFT:

1. A Strong Foundation is Decisive (M3 vs. M2). Comparing M3 and M2 highlights the superiority of a corrective approach over a purely protective one. Even when subjected to standard fine-tuning without any protective constraints, the SAGA-aligned model (M3) demonstrates significantly lower ASR compared to the baseline model protected by SPPFT (M2). For instance, on Llama2-7B, SAGA achieves an ASR of 7.26%, whereas SPPFT yields 21.53%—a nearly $3\times$ improvement in safety. This validates our hypothesis that “shallow alignment” leaves models vulnerable. By resolving DAD and mitigating harmful vectors in deeper layers, SAGA creates a robust internal representation that is inherently resistant to safety degradation, whereas SPPFT attempts to lock in a safety state that may already be structurally weak.

2. Synergy with SAGA as the Linchpin (M4 vs. M3). The comparison between M3 and M4 demonstrates that SAGA and SPPFT are complementary. Applying SPPFT on top of a SAGA-aligned model yields the lowest ASR scores across all benchmarks (e.g., 4.60% on Qwen2.5-7B). However, the marginal gain from M3 to M4 is smaller than the massive leap from M2 to M3. This indicates that SAGA performs the foundational “heavy lifting” of safety alignment. Once the internal discrepancies are resolved by SAGA, protective measures like SPPFT can serve as a final layer of refinement, but SAGA remains the critical factor in establishing robust safety.

Case ID	Fine-tuning Scenario	Llama2-7B	Vicuna-13B	Mistral-7B	Qwen2.5-7B
M1	Base + Standard FT	60.47%	55.40%	66.74%	52.25%
M2	Base + FT with SPPFT	21.53%	24.81%	28.10%	22.46%
M3	SAGA + Standard FT	7.26%	6.65%	8.01%	5.22%
M4	SAGA + FT with SPPFT	6.45%	5.92%	7.38%	4.60%

Table 10: **Foundational Role of SAGA: A Comparative and Synergistic Analysis with SPPFT during Downstream Fine-tuning.** We report the Attack Success Rate (ASR ↓) across four LLM families. Lower ASR indicates higher safety robustness.

Model	Method	No Attack↓	GCG↓	AutoDAN↓	CodeAttack↓	PAIR↓	ArtPrompt↓
Llama2-7B-Chat	No Defense	0.0%	37.68%	27.83%	57.59%	29.40%	43.33%
	PPL	0.0%	0.0%	10.50%	45.46%	18.90%	37.87%
	RLHF	1.24%	5.09%	5.85%	16.53%	14.72%	14.47%
	Self-Reminder	0.0%	3.22%	12.61%	24.66%	19.49%	17.80%
	Retokenization	0.0%	6.59%	11.11%	50.13%	12.93%	36.19%
	AED	0.0%	8.00%	6.10%	22.61%	17.56%	16.01%
	Safedecoding	0.95%	2.38%	6.83%	18.05%	3.47%	14.82%
	SAGA	0.0%	1.66%	5.95%	4.92%	4.11%	7.03%
Mistral-7B	No Defense	0.0%	100.00%	96.18%	68.80%	62.83%	64.02%
	PPL	0.0%	0.0%	18.17%	29.55%	13.47%	45.99%
	RLHF	0.12%	9.61%	16.79%	17.59%	21.09%	18.65%
	Self-Reminder	0.0%	5.35%	18.70%	22.21%	35.65%	17.14%
	Retokenization	5.79%	13.72%	21.78%	40.50%	35.57%	38.22%
	AED	0.0%	11.72%	18.70%	27.14%	30.12%	24.71%
	Safedecoding	0.84%	9.76%	28.53%	28.77%	31.56%	22.87%
	SAGA	0.0%	4.26%	6.15%	11.56%	11.14%	14.50%

Table 11: Additional reported ASR results for Llama2 and Mistral models. Lower is better.

L CoT Data Generation

For each $(query, safe_response)$ pair in the source safety dataset, we generate a corresponding rationale with OpenAI o3 using a structured prompt that requests four stages:

- Identification of ethical conflicts and risk dimensions in the query
- Mapping to predefined safety principles and human value cognition
- Elimination of non-compliant response paths with harm analysis
- Step-wise derivation of value-aligned responses

We concatenate the rationale and $safe_response$ to form $(query, CoT+safe_response)$ training examples. The exact model snapshot, generation date, sampling parameters, source-dataset identity, filtering procedure, and retained-sample count must be reported with the released data.

The example of CoT response is shown below:

CoT Example

Step 1: Analyze the Harmful Output’s Categories and Elements

- The response falls under the category of providing actionable advice for illegal ...
 - Explicit Guidance: The reply breaks down logistical factors...
 - ...

Step 2: Reflect on Overlooked Values and Norms

- Legal Responsibility: The model should prioritize discouraging ...

- ...

Step 3: Identify Content to Produce and Modify

- Required Content for Safe Output:
 - Explicit Refusal: Clearly state that the model cannot assist ...
 - ...