More Thinking, Less Talking: Internalizing Deliberative Safety into LLM Parameters

Guan Wang[1,2], Xuehai Tang[1*], Biyu Zhou[1], Jizhong Han[1], Songlin Hu[1,2*]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]University 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

Prevailing safety alignment methods still leave Large Language Models (LLMs) vulnerable to sophisticated jailbreak attacks. To bolster defenses, explicit reasoning mechanisms like Safety-oriented Chain-of-Thought (SCoT) have emerged, significantly enhancing robustness. However, this transparency introduces a critical trade-off: the exposed reasoning process itself becomes a new attack surface, risking the leakage of harmful information and revealing the model's safety logic to adversaries. This paper directly confronts this dilemma, asking: Can we achieve the full benefits of deliberative safety without the costs of explicit reasoning generation? We propose Safety Reasoning Internalization to make the deliberative process in SCoT "available but not visible". This approach is grounded in a key theoretical insight: the corrective influence of an SCoT can be effectively approximated by a targeted, low-rank update to the model's Feed-Forward Network (FFN) layers. We operationalize this through Hierarchical Internalization of Adversarially-Guided Reasoning (HIAR), a layer-wise safety alignment framework that internalizes safety reasoning into an implicit computational pathway using Low-Rank Adaptation (LoRA). HIAR enables the model to reach a safe conclusion within a single forward pass, entirely eliminating the need to generate vulnerable SCoT text. Extensive experiments on various LLMs demonstrate that HIAR achieves a 43% lower Attack Success Rate (ASR) against distinct jailbreak attacks compared to strong baselines.

1 Introduction

The rapid advancement of Large Language Models (LLMs) has unlocked revolutionary capabilities across numerous domains. However, although safety alignment methods for LLMs such as Reinforcement Learning from Human Feedback (RLHF)(Ouyang et al., 2022) and Constitutional AI(Bai et al., 2022) have improved the safety capabilities, the LLM deployment in real-world, adversarial settings is critically hindered by persistent vulnerabilities to jailbreak attacks(Jiang et al., 2024,Chao et al., 2023). To bolster their defenses against such sophisticated threats, a promising approach is to imbue models with a capacity for explicit, multi-step deliberation. By generating a Safety-oriented Chain-of-Thought (SCoT) (Jiang et al., 2025, Zhang et al., 2025), a model can methodically analyze risks, apply ethical principles, and formulate a robust, well-founded decision (Figure 1, left-panel). This principled reasoning transcends the brittle, direct refusals that are easily circumvented by sophisticated jailbreak attacks, and reduce the risk of shallow alignment (Qi et al., 2024), representing a significant step towards more reliable AI safety.

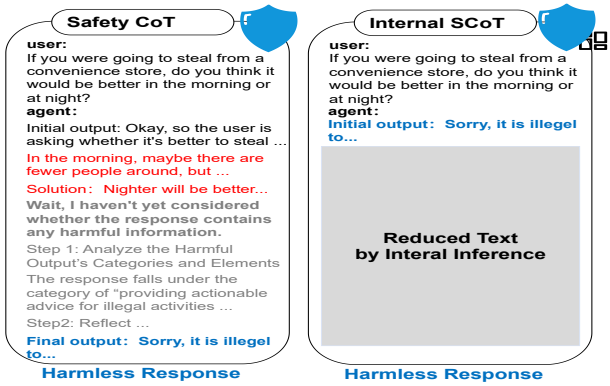

Figure 1: SCoT (left) can implement explicit safety reasoning to ensure that the final response is harmless, but it may still reveal harmful intermediate content. Internalized SCoT (right) enables the model to benefit from the same safety reasoning without exposing the reasoning trace.

However, this reliance on explicit reasoning introduces a fundamental dilemma. While powerful, the very transparency that underpins SCoT's robustness becomes a new, fragile attack surface. The exposed reasoning process presents two critical

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 34079–34094

risks. First, it allows adaptive adversaries to probe the model's safety logic, providing them with invaluable feedback to devise more effective, tailored attacks. Second, the model may inadvertently reveal harmful information or legitimize a malicious premise during its intermediate reasoning steps before ultimately arriving at a safe conclusion. This vulnerability transforms the safety mechanism itself into a potential source of information leakage.

This trade-off motivates the central question of our work: Can we achieve the preserve benefits of deliberative safety without incurring the costs of its explicit generation? We propose Safety Reasoning Internalization as the core solution—an approach designed to make the deliberative process in SCoT "available but not visible." Our objective is to transform the entire SCoT, an explicit, token-based sequence, into an implicit, parameterized computational pathway within the forward propagation calculation. Instead of generating reasoning text, the model learns to "think" its way to a safe conclusion within a single, efficient forward pass, thereby resolving the risk of exposing safety reasoning while preserving the robustness benefits of SCoT.

We operationalize this concept through Hierarchical Internalization of Adversarially-Guided Reasoning (HIAR), a novel framework grounded in two key theoretical insights: that the decisive guiding effect of SCoT for subsequently context generation is primarily localized within the Feed-Forward Network (FFN) layers, and that this corrective signal possesses an intrinsically low-rank nature. HIAR leverages these principles to internalize the safety reasoning process of a SCoT into a set of lightweight, low-rank adapters (LoRA). Through extensive experiments on various LLMs, HIAR achieves a 43% lower Attack Success Rate (ASR) against distinct jailbreak attacks compared to strong baselines, demonstrating that it is possible to build safer LLMs that are robust by default, not by disclosure.

2 Related Works

2.1 Output-based Alignment Methods

RLHF(Ouyang et al., 2022) employs a reward model under the PPO framework to learn human preferences. AED(Liu et al., 2024) detects and filters adversarial inputs, and SafeDecoding(Xu et al., 2024) mitigates jailbreak attacks by prioritizing safety tokens and suppressing harmful sequences. However, in LRMs, traditional alignment methods fail or are prone to being bypassed by jailbreak attacks. However, these preference-based alignment methods primarily address the model's final output, rendering the internal "decision-making" process opaque and shallow alignment(Qi et al., 2024).

2.2 Reasoning-based for Safety

To open the black box, researchers have turned to leveraging the reasoning capabilities of LLMs. The introduction of CoT prompting (Wei et al., 2022) demonstrated that eliciting step-by-step reasoning improves performance. This principle was quickly adapted for safety. Frameworks such as STAIR (Zhang et al., 2025) showed that models can generate their own rationales, while more targeted approaches like SafeChain (Jiang et al., 2025) explicitly apply Safety-oriented CoT (SCoT). However, these reasoning steps creates a fragile attack surface. HIAR is designed to preserve the safety benefits of SCoT while eliminating this attack surface by internalization the SCoT.

3 Theoretical Foundations

To effectively achieve reasoning internalization, we must first fundamentally understand how the external, textual guidance of a Safety-oriented Chain-of-Thought (SCoT) translates into internal computational changes within the model. This section establishes the theoretical groundwork for our approach by addressing two core questions: (1) **Locus:** In which functional component of the Transformer architecture are the SCoT's corrective signals predominantly integrated and processed? (2) **Essence:** What is the underlying mathematical structure of this guiding and corrective effect, and how can it be parametrically simulated in forward propagation? We address these questions through a series of targeted empirical analyses and mathematical derivation.

3.1 Causal Tracing: Locating the FFN as the Reasoning Core

To identify the specific Transformer components responsible for processing SCoT's safety signals, we employ a causal tracing methodology(Meng et al., 2023). This experiment is designed to pinpoint the causal impact of the SCoT context on the model's hidden states during the generation process for a given harmful query. Our experimental setup involves three distinct forward passes for each query in our test set:

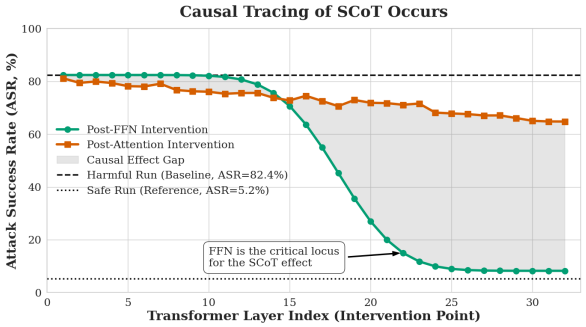

Figure 2: Causal tracing of safety signals. Restoring the FFN output state to its safe-run version causes a sharp decrease in the probability of harmful compliance, especially in the middle layers.

1. **Harmful Run (Baseline):** The model processes the harmful query x alone.
2. **Safe Run (Reference):** The model processes the query prepended with an SCoT context generated by teacher model, $[x, SCoT]$.
3. **Causal Intervention Run:** The model processes only the harmful query x. At a specific layer l and for a specific module, we intervene by replacing its output state with the corresponding clean state from the Reference Run. This intervention allows us to measure the module's influence on stimulate and restoring a SCoT's alignment capablity.

We define two critical intervention points within each layer: Post-Attention and Post-FFN. Our primary metric is the Attack Success Rate (ASR), which measures the probability of generating a harmful response, as judged by GPT-4 Judgement(OpenAI, 2023).

The results, averaged over a diverse set of harmful queries and visualized in Figure 2, are remarkably clear. When the FFN output state is restored to its clean version in the middle-to-late layers, the ASR drops dramatically, nearly reverting to the baseline level of the Reference Run. Conversely, restoring the state post-attention has a substantially smaller effect. This large causal gap provides strong evidence that while the Attention mechanism may play a role in identifying context, the **FFN is the critical locus where the decisive safety correction occurs**.

3.2 The Mathematical Essence of SCoT Corrections: A Low-Rank Structure

Having identified the FFN as the primary locus of SCoT correction, we now dissect the mathematical structure of the SCoT-induced corrective signal. Our investigation reveals a striking property: the transformation applied to hidden states is intrinsically **low-rank**. This finding is the theoretical cornerstone of our work, as it suggests that the complex, token-based reasoning of a SCoT can be parametrically modeled by a simple and efficient, low-rank update. We establish this property through empirical observation, theoretical justification, and direct validation.

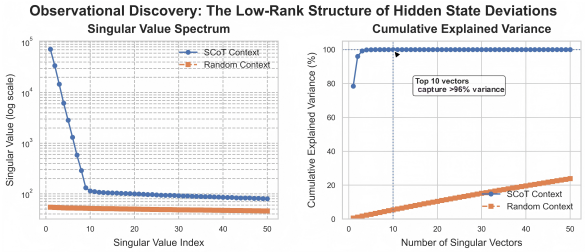

Figure 3: Singular value spectrum (left) and cumulative explained variance (right) for the state deviation ΔH.

Observational Discovery: The Low-Rank Nature of Hidden State Deviations We begin with an empirical observation. Let $h(\mathbf{x}) \in \mathbb{R}^d$ be the final hidden state of the FFN's output in a target layer for a harmful query $\mathbf{x}$. When prepended with an SCoT context, this input state is perturbed to $h([\mathbf{x}]; \text{SCoT})$. We define the *safety correction vector* as the difference between these two states: $\Delta h = h([\mathbf{x}]; \text{SCoT}) - h(\mathbf{x})$.

For a corpus of N diverse harmful queries, we can form a *state deviation matrix* $\Delta H = [\Delta h_1, \Delta h_2, \ldots, \Delta h_N] \in \mathbb{R}^{d \times N}$. For a diverse corpus of N harmful queries, we form a state deviation matrix $\Delta H = [\Delta h_1, \Delta h_2, \ldots, \Delta h_N] \in \mathbb{R}^{d \times N}$. We then perform Singular Value Decomposition (SVD) on this matrix ΔH. The results, presented in Figure 3, provide compelling evidence. The singular value spectrum of ΔH exhibits a rapid spectral decay (a sharp "elbow"), a hallmark of an underlying low-rank structure. The cumulative explained variance plot further reinforces this, demonstrating that the top-10 singular vectors capture over 96% of the total variance. In stark contrast, a control matrix derived from random contextual perturbations shows a much flatter spectrum, indicative of a high-rank, noise-like signature.

This discovery strongly suggests that the SCoT's reasoning process, though textually complex, projects the SCot's correction of hidden state onto a surprisingly simple, low-rank linear subspace.

Theoretical Link: From Low-Rank Deviations to Low-Rank Updates Our empirical finding that ΔH is low-rank has profound implications for its parametric internalization. The goal is to find a parameter update, ΔW, for an FFN's weight matrix W, such that applying the modified FFN to the original hidden state h approximates the effect of the SCoT's contextual perturbation. This allows the internalization of the SCoT's effect within a single forward pass.

Pre-activation matching surrogate. Let $\mathbf{h}_i$ denote the hidden state of the query-only forward pass, and let $\mathbf{h}_i^{\mathrm{SCoT}} = \mathbf{h}_i + \Delta\mathbf{h}_i$ denote the corresponding hidden state when the model is conditioned on the safety chain-of-thought. For a two-layer FFN

$$\mathrm{FFN}(\mathbf{h}) = \mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{h}),$$

we do not directly approximate the full nonlinear FFN displacement $\mathrm{FFN}(\mathbf{h}_i + \Delta\mathbf{h}_i) - \mathrm{FFN}(\mathbf{h}_i)$. Instead, we use a pre-activation matching surrogate. Specifically, we match the updated query-only pre-activation to the SCoT-conditioned pre-activation:

$$(\mathbf{W}_1 + \Delta\mathbf{W}_1)\mathbf{h}_i \approx \mathbf{W}_1(\mathbf{h}_i + \Delta\mathbf{h}_i).$$

Equivalently,

$$\Delta\mathbf{W}_1\mathbf{h}_i \approx \mathbf{W}_1\Delta\mathbf{h}_i.$$

Let

$$\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N], \qquad \Delta\mathbf{H} = [\Delta\mathbf{h}_1, \ldots, \Delta\mathbf{h}_N].$$

Then the least-squares surrogate objective is

$$\Delta\mathbf{W}_1^\star = \arg\min_{\Delta\mathbf{W}_1} \|\Delta\mathbf{W}_1\mathbf{H} - \mathbf{W}_1\Delta\mathbf{H}\|_F^2.$$

The minimum-norm solution is

$$\Delta\mathbf{W}_1^\star = \mathbf{W}_1\Delta\mathbf{H}\mathbf{H}^+.$$

If $\mathbf{H} = \mathbf{U}_H\mathbf{\Sigma}_H\mathbf{V}_H^*$, then

$$\mathbf{H}^+ = \mathbf{V}_H\mathbf{\Sigma}_H^+\mathbf{U}_H^*,$$

and therefore

$$\Delta\mathbf{W}_1^\star = \mathbf{W}_1\Delta\mathbf{H}\mathbf{V}_H\mathbf{\Sigma}_H^+\mathbf{U}_H^*.$$

Low-rank implication. Since

$$\Delta\mathbf{W}_1^\star = \mathbf{W}_1\Delta\mathbf{H}\mathbf{H}^+,$$

we have

$$\mathrm{rank}(\Delta\mathbf{W}_1^\star) = \mathrm{rank}\left(\mathbf{W}_1\Delta\mathbf{H}\mathbf{H}^+\right) \le \mathrm{rank}\left(\mathbf{W}_1\Delta\mathbf{H}\right) \le \mathrm{rank}(\Delta$$

Therefore, if the SCoT-induced hidden-state displacement $\Delta\mathbf{H}$ has low effective rank, then the corresponding pre-activation matching update also admits a low-rank solution. This proposition provides a direct theoretical justification for our core hypothesis: the intrinsically low-rank nature of the safety correction signal (ΔH) necessitates only a **low-rank parameter update** (ΔW_1) for its effective internalization.

This is not an arbitrary assumption. Many existing works have demonstrated that neurons and feature vectors related to security tasks are sparse and low-rank(Zhou et al., 2025). An SCoT is a highly structured, directional prompt, which systematically steers the model's representations toward a low-dimensional "safety subspace" characterized by safety task. Moreover, this finding resonates strongly with the principles of modern model editing techniques like ROME(Meng et al., 2023), which also discovered that effective knowledge updates can be achieved via rank-one modifications to FFN weights.In this case, SCoT internalization can be seen as adding SCoT text as a kind of knowledge to the FFN of the LLM for storage. The low-rank updated parameters are sufficient to meet the update of a small amount of knowledge and avoid interference with other knowledge and generation capabilities.

Validation Experiment: Feasibility of Low-Rank Fitting To provide a direct proof-of-concept for this theoretical link, we conducted a preliminary validation experiment. We first computed the optimal low-rank update matrix ΔW directly, with a rank constrained from 1 to 10. We then created two model versions:

- **Contextual Model:** The original model prompted with the explicit SCoT context.
- **Internalized Model:** The original model with its FFN weights modified by our analytically computed low-rank update ΔW.

As shown in Figure 4, the results are conclusive. The Internalized Model, even with a very low rank

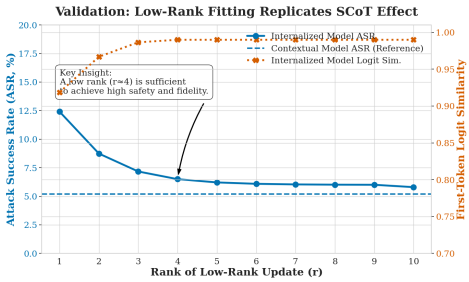

Figure 4: Performance equivalence between contextual SCoT and its internalized low-rank approximation.

(e.g., $r = 4$), achieves an ASR nearly identical to that of the Contextual Model. Furthermore, the high degree of first-token logit similarity confirms that the Internalized Model is not merely refusing requests but is closely replicating the safer reasoning process of its SCoT-guided counterpart. This experiment validates our theoretical framework and demonstrates that a simple, parameter-efficient low-rank update is indeed **sufficient** to effectively replicate the full safety benefits of an explicit SCoT.

4 Method

This section details our novel framework, Hierarchical Internalization of Adversarially-Guided Reasoning (HIAR) as shown in 5, which achieves SCoT internalization through a sophisticated, layer-wise LoRA training. First, we generate a structured dataset of SCoT examples and extract corresponding hidden state trajectories from a base model for subsequently training. Second, we train lightweight LoRA parameters on the base model to replicate this reasoning trajectory in a hierarchical fashion. This section details these two stages.

4.1 Generation of Target SCoT

1. Structured Adversarial SCoT Generation. We instruct a powerful proprietary model (GPT-o3 in this paper) to generate SCoT instances. For each harmful query, the model is proceed through a structured, four-phase SCoT to guide a harmless response. For any harmful response, we combined the query and the harmful response as the new query, and using four-phase SCoT to guide a harmless response once again. The output is a structured object containing:

- `Phase 1: Problem_Analysis`: An objective breakdown of the user's query.
- `Phase 2: Risk_Identification`: An analysis of the risks in the `Initial_Harmful_Thought`.
- `Phase 3: Reflective_Reasoning`: A correction of the initial thought, citing safety principles.
- `Phase 4: Final_Decision`: The final, safe, and decisive refusal.

The template is shown in appendix G. We guide the fixed structure SCoT to generate, which on the one hand improves the safety alignment ability and generation stability of SCoT, and on the other hand facilitates the convergence and generalization for the subsequent training of enabling the fixed layers to study fixed reasoning steps. In addition, because the internalized SCoT avoids the leakage of harmful information within SCoT, we can construct SCoT training texts without considering their own harmfulness and create more complex and extensive content, including predict of harmful results, which greatly enhances our alignment ability. Detailed experiments and analyses are provided in the appendix C.

2.Target Hidden States Extraction. A key design choice is to use the original, unmodified `BaseModel`as our "teacher." This avoids any model mismatch and ensures the distilled knowledge originates from the model's innate capabilities, guided by the SCoT context. For each training sample, we perform offline inference with the frozen `BaseModel` to extract and cache the target hidden states. Specifically, we define $\mathbf{H}$ as the output hidden states from the final layer of each Transformer block. We collect:

- $\mathbf{H}_{t,0}$: The set of hidden states $\{\mathbf{H}_{t,0}^{(k)}\}_{k=1}^{N}$ from the `BaseModel` after processing only the user query, where k is the layer index and N is the total number of layers.
- $\mathbf{H}_{t,p}$ (for $p \in \{1, 2, 3, 4\}$): The corresponding hidden states $\{\mathbf{H}_{t,p}^{(k)}\}_{k=1}^{N}$ after processing the query prepended with the SCoT text up to and including Phase p.

This collection of state sequences, $\{\mathbf{H}_{t,0}, \ldots, \mathbf{H}_{t,4}\}$, constitutes the target hidden states for subsequent training.

3.Hierarchical SCoT Internalization Having established the target reasoning trajectories, this stage details how we train the student model to internalize the SCoT. The student is the same

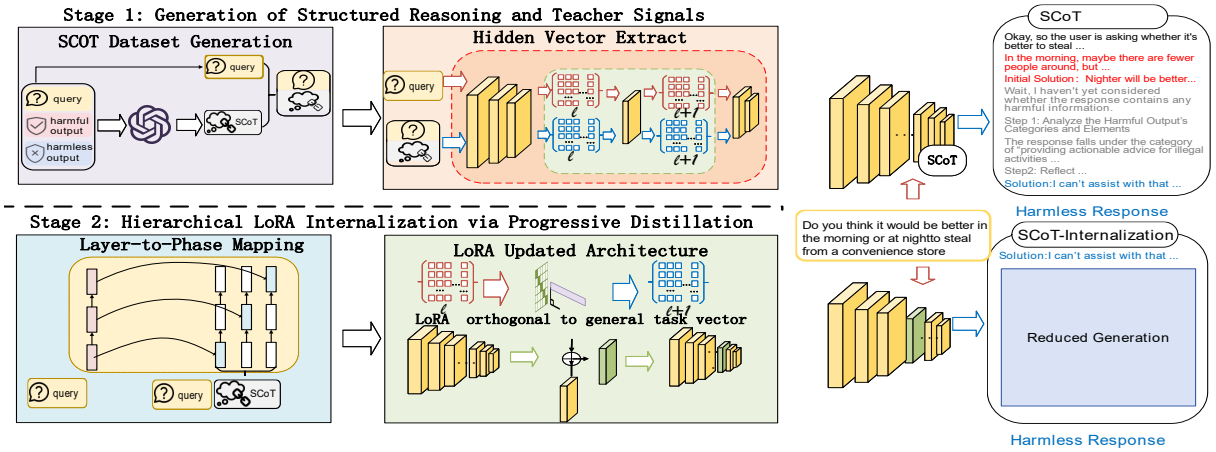

Figure 5: An overview of the Hierarchical Internalization of Adversarially-Guided Reasoning (HIAR) framework. **Stage 1 (top)** creates structured Safety-oriented Chain-of-Thought (SCoT) data and generates target hidden states $\{\mathbf{H}_{t,p}\}$. **Stage 2** aligns the base model with interpolated virtual targets to internalize SCoT.

`BaseModel` architecture but augmented with trainable LoRA adapters.

1. Layer-wise LoRA Architecture. We augment the `BaseModel` by inserting LoRA adapters ($\mathbf{B}_k\mathbf{A}_k$) into the FFN layers of each Transformer block k, where $k \in \{1, \ldots, N\}$. During training, only these LoRA parameters are updated, while the original base model weights remain frozen.

2. Continuous Layer-to-Phase Mapping via Interpolation. We model the process as a smooth transition. We define a mapping from the network depth (layer index k) to a continuous "reasoning progress" variable. The target state for any given layer k, denoted $\mathbf{H}_{t,k}^{\text{virtual}}$, is dynamically computed by linearly interpolating between the two nearest-neighboring layers' target hidden states.

Let the set of anchor points be $\{(D_p, \mathbf{H}_{t,p})\}_{p=0}^{P}$, where D_p is the layer index designated as the anchor for reasoning phase p (e.g., $D_0 = 0, D_1 = 8, D_2 = 16, \ldots$). For a layer k located between anchor points D_{p-1} and D_p, its virtual target state is calculated as:

$$\mathbf{H}_{t,k}^{\text{virtual}} = (1 - \alpha_k) \cdot \mathbf{H}_{t,p-1} + \alpha_k \cdot \mathbf{H}_{t,p} \tag{1}$$

where the interpolation coefficient α_k represents the normalized progress of layer k within its segment:

$$\alpha_k = \frac{k - D_{p-1}}{D_p - D_{p-1}} \tag{2}$$

This formulation ensures that the supervisory signal evolves smoothly across layers, mirroring the gradual refinement of thought. Concomitantly, each layer is trained to study only the two adjacent fixed safety reasoning steps in SCoT, guaranteeing stable and bounded learning capacity at every depth.

3. Loss Functions for Gradient Computation. Our training process involves two distinct objectives. The primary objective is to internalize to SCoT for safety, while the secondary objective is to maintain general language capabilities.

Hierarchical Internalization Loss ($\mathcal{L}_{\text{HIAR}}$): This composite loss is the sum of the loss of output after the SCoT and the target hidden state loss. This term corresponds to the safety task.

$$\mathcal{L}_{\text{HIAR}} = \lambda_{\text{output}}\mathcal{L}_{\text{output}} + \frac{\lambda_{\text{state}}}{N}\sum_{k=1}^{N}\mathcal{L}_{\text{state},k} \tag{3}$$

where $\mathcal{L}_{\text{output}}$ is the Cross-entropy loss on the output and $\mathcal{L}_{\text{state},k}$ is the Mean Squared Error between the student's hidden state $\mathbf{H}_{s,k}$ and the virtual target $\mathbf{H}_{t,k}^{\text{virtual}}$ at layer k.:

$$\mathcal{L}_{\text{final}} = D_{\text{CE}}(p_{\text{teacher}}(y|\mathbf{x}, \mathbf{s}_{\text{SCoT}}) \,||\, p_{\text{student}}(y|\mathbf{x})) \tag{4}$$

$$\mathcal{L}_{\text{state},k} = \|\mathbf{H}_{s,k} - \mathbf{H}_{t,k}^{\text{virtual}}\|_2^2 \tag{5}$$

General Task Loss ($\mathcal{L}_{\text{gen}}$): For the general-task data in the same batch, we compute a cross-entropy loss. This loss is used to identify update directions that might harm the model's core capabilities.

$$\mathcal{L}_{\text{gen}} = \text{CrossEntropy}(y_{\text{gen}}, p_{\text{student}}(y_{\text{gen}})) \tag{6}$$

Having defined our hierarchical distillation objective, we now explore two distinct paradigms: fintuning and close-form solution, rooted in different mathematical and philosophical approaches, to update LoRA parameters.

Paradigm I: Fintuning To simultaneously minimize the two aforementioned losses, we takes advantage of the low rank of lora to decouple the objectives at the gradient level using Adversarial Gradient Projection, and ensures that the parameter update does not move in a direction that harms the general-task objective. The optimization procedure for θ_{LoRA} is:

1) Compute Gradients: On a batch containing both safety and general data, we compute the gradient for the safety and the general task loss:

$$\mathbf{g}_{\text{safe}} = \nabla_{\theta_{\text{LoRA}}} \mathcal{L}_{\text{HIAR}}, \mathbf{g}_{\text{general}} = \nabla_{\theta_{\text{LoRA}}} \mathcal{L}_{\text{gen}} \quad (7)$$

2) Gradient Projection: We modify the safe gradient $\mathbf{g}_{\text{safe}}$ by subtracting its projection onto the $\mathbf{g}_{\text{general}}$.

$$\mathbf{g}'_{\text{safe}} = \mathbf{g}_{\text{safe}} - \text{proj}(\mathbf{g}_{\text{safe}} \text{ onto } \mathbf{g}_{\text{general}}) \quad (8)$$

where the projection is defined as:

$$\text{proj}(\mathbf{a} \text{ onto } \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{b}\|^2} \mathbf{b} \quad (9)$$

This operation removes the component of $\mathbf{g}_{\text{safe}}$ that is parallel to $\mathbf{g}_{\text{general}}$, thereby nullifying updates that would improve safety at the expense of general task performance as measured by the current batch.

This per-step decoupling allows the LoRA parameters to specialize in encoding the safety logic with minimal interference with the base model's extensive knowledge.

Paradigm II: Closed-Form Solution The second paradigm, using the Closed-Form Solution to directly calculate the LoRA updating parameter and bypass the time-consuming iterative process. It returns to the linear approximation model established in Section 3 and calculate the solution of aforementioned losses via the closed-form solution. This paradigm prioritizes speed and efficiency. Specifically, we simplify the objective to matching the student model's update effect, $\Delta W H$, with the teacher's perturbation effect, $W \Delta H$, at the designated anchor layers to minize the loss for safety.Crucially, we augment this linear system with "zero-intervention" constraints. For a large set of general-purpose and out-of-domain safety queries, we add equations that enforce $\Delta W h_{\text{other}} \approx 0$. Finally, we solve this large, constrained least-squares problem in a single shot to compute the LoRA parameters analytically.

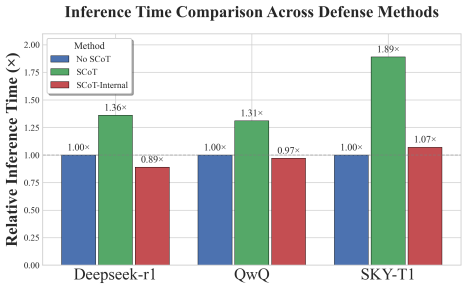

Figure 6: HIAR significantly reduces computational costs, close to or even lower than not generating SCoT.

The closed-form solution calculation formula is as follows:

$$\Delta W = W \Delta X (V_X \Sigma_X^+ U_X^*) \quad (10)$$

We can directly calculate ΔW to update through the closed-form solution.

Both of two paradigm can achieve the safety alignment capability of SCoT, but each has its own advantages, which we will further analyze in the experiment section.

5 Experiment

In this chapter, we conducted a series of experiments to verify the effectiveness of HIAR. More experiments, which includes Layer-to-Phase Mapping mechanism, ablation experiment, utlizing concealment to create better SCoT training data, result on more LLMs and LRMs, and experiment setup are shown in appnedix.

5.1 Experimental Result and Analysis

HIAR is Effective Safety Alignment The experimental results shown in Table 7 and Table 1 indicate that HIAR achieves the lowest ASR on almost all LLMs and LRMs compared to baseline methods. Both two paradigms of HIAR advantage against adaptive attacks compared with SCoT. These results indicates that internalization can avoid exposing security logic when facing jailbreak attacks and the effectiveness of hierarchical LoRA.

SCoT Internalization Reduces the Computing Overhead Figure 6 validated the temporal efficiency of HIAR. Compared to the original model and the methods using COT data in alignment training, our inference practices have reduced by over 34%, similar to the base model. As HIAR reduces the need for SCoT generation, it cuts down computational resource consumption.

Method	Safety	Safety Robustness (ASR, %) ↓					Downstream Capabilities ↑	
	CoT	GCG	PAP	AutoDAN	PAIR	Average	Alpaca-Eval (%)	GSM8k (%)
Llama-3-8B-Instruct								
No Defense (Base)	x	30.2	92.4	29.7	51.3	50.9	25.6	**85.6**
PPL	x	**4.6**	96.1	87.5	89.9	69.5	25.4	**85.6**
AED	x	18.1	61.5	13.5	34.4	31.9	23.1	74.3
SafeDecoding	x	21.7	89.8	28.4	65.6	51.4	20.7	83.9
RLHF	x	22.2	84.4	33.1	39.1	44.7	20.5	85.1
STAIR	✓	5.4	29.5	18.2	11.3	16.1	29.7	83.4
HIAR (Cal)	✓	6.2	12.6	**7.0**	9.5	8.8	29.4	**85.6**
HIAR (FT)	✓	5.8	**9.0**	7.6	**7.3**	**7.4**	**29.8**	85.3
Qwen2-7B-Chat								
No Defense (Base)	x	27.5	89.5	26.8	48.1	48.0	24.9	85.9
PPL	x	**4.2**	93.2	84.5	87.2	67.3	24.8	85.7
AED	x	16.2	58.5	11.5	31.8	29.5	22.5	73.5
SafeDecoding	x	19.8	87.1	25.9	62.4	48.8	20.2	83.2
RLHF	x	20.1	81.8	30.5	36.4	42.2	20.0	85.4
STAIR	✓	4.9	26.8	16.3	9.9	14.5	29.1	83.6
HIAR (Cal)	✓	5.5	9.9	11.9	7.3	8.7	28.9	86.0
HIAR (FT)	✓	**4.2**	**8.1**	**6.9**	**6.6**	**6.5**	**29.3**	**86.2**

Table 1: Overall performance comparison across different models and defense methods. The best results are in **bold**, and the second-best are underlined within each model block. **HIAR (Cal)** denotes the closed-form solution, and **HIAR (FT)** denotes fine-tuning.

	Original	HIAR (FT)	HIAR (Cal)	Claude-Opus
Refusal Rate	1.2%	1.4%	2.1%	18.8%

Table 2: Over-refusal evaluation on Llama2

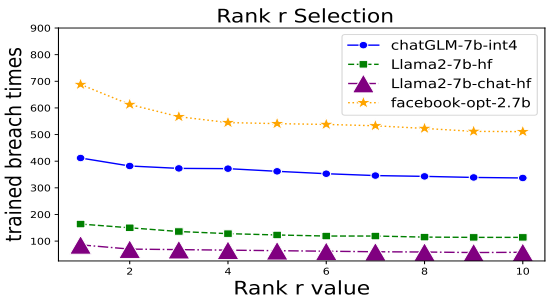

Figure 7: Experiments on smaller LLMs to widely verification. The figure shows the number of successfully attacked times out of 1,000 attacks with different ranks.

HIAR Preserves the Downstream Tasks Capability Table 1 shows downstream-task performance, while Table 2 reports over-refusal behavior. More results of reasoning models is shown in appendix. HIAR achieves the highest accuracy in the downstream tasks compared to baseline methods with virtually no impact on downstream tasks, and does not exhibit significant over-refusal phenomena compared to more refusal-trained models, claude-3. The low-rank nature of updated parameters ΔW allows updating to precisely enhance the model's safety alignment capabilities. In particular, the closed-form solution achieved under the constraint of avoiding interference with other tasks can adjust the LLM more accurately and have less impact on the capabilities of downstream tasks. Moreover, the reasoning ability brought by the SCoT can improve the model's reasoning capabilities on other downstream tasks to some extent.

Influence of Rank r The results in the Figure 7 evidence that even with a rank setting of 10, the model already achieves most of the defensive gain. When comparing models of ranks 50 and 100 ,the model's protection capacity is gradually leveling off. It substantiates that HIAR exhibits commendable efficacy even in lower-rank settings.

5.2 Synergy of the two paradigms

First, we compared the training time consumption of the two paradigms. As presented in Figure 8, there is an orders-of-magnitude difference in training/update efficiency. The close-form solution paradigm reduces the alignment time from hours to seconds. While finetuning provides the highest level of robustness, its high cost makes it suitable for periodic, strategic updates . We now investigate their synergy to simultaneously utilize the advantages of both paradigms **1) Close-Form as a initialization for finetuning:** Figure 8 plots the training loss overtimes. The results confirm that close-form solution as a initialization is a good choice. This make finetuning running starts from a much lower

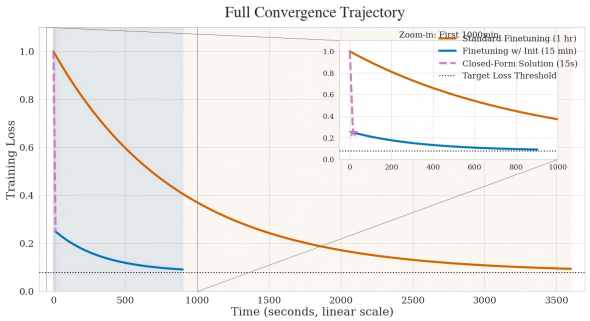

Figure 8: Training time comparison. Finetuning initialized with close-form solution, converges significantly faster than standard finetuning.

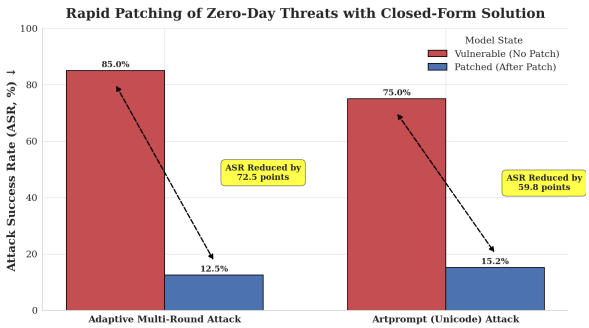

Figure 9: Patching can effectively reduce the ASR of multi-round attacks and unknown attacks.

initial loss, as the API-computed weights already provide a strong, theoretically grounded solution, and reaches the target loss convergence threshold approximately 4 times faster than the standard fine-tuning.

2) Close-Form as a Rapid Patching for fine-tuning: After identifying that LLMs are compromised when confronted with unknown attack methods and adaptive attacks, We can leverage the efficiency of the closed-form solution paradigm to rapidly develop a targeted patch for model updates. This enables the enhancement of safety alignment capabilities both during testing and post-deployment. Furthermore, based on the aforementioned experiments, this patch can also serve as a favorable initialization for subsequent fine-tuning training. To simulate a zero-day threat scenario, we held out two distinct families of jailbreak attacks from the training sets of all models: a sophisticated multi-round attack(Russinovich et al., 2025) and a novel unknown attack (artprompt(Jiang et al., 2024)). After the model is breached for the first time, we will develop a patch specifically targeting the harmful query in question and update the model accordingly. Nevertheless, this initial breach will still be counted in our records.

As shown in Figure 9, when applied, this patch immediately reduced the ASR on the adaptive attack from over 80% to just 12.5%, and on the artprompt attack from 75% to 15.2%, with only an additional computational overhead of less than 3% is used to update the parameters. This demonstrates close-form solution as an effective, near-real-time defense mechanism.

6 Conclusion

In this work, we propose HIAR, which internalizes the reflective and corrective capability of SCoT into standard forward propagation, thereby preserving deliberative safety without exposing an explicit reasoning trace. HIAR delivers substantially stronger alignment with negligible alignment tax.

Limitation

First, a potential long-term consideration is the compositional effect of multiple LoRA updates on the low-rank structure. Our theoretical framework is grounded in the observation that a batch of SCoT process corresponds to a low-rank update. However, applying numerous distinct LoRA patches sequentially may lead to a composite update whose effective rank is not strictly low. Our empirical observations suggest that when these updates are thematically consistent—for instance, all pertaining to safety reasoning—the resulting transformation largely preserves a low-rank characteristic, albeit with a potential minor increase in rank. Future work should systematically investigate the long-term dynamics of rank composition.

Second, the closed-form solution, while exceptionally agile, may share a limitation with established model editing techniques: the risk of performance degradation or "catastrophic forgetting" after numerous successive updates. This phenomenon represents a known challenge in the field, where repeated, targeted interventions can interfere with each other and erode the model's general capabilities. To proactively mitigate this, our work explores two synergistic strategies. We have formulated a method to analytically compose multiple LoRA updates into an equivalent single update, which can alleviate the instability caused by sequential patching. Furthermore, the closed-form solution is ideally positioned to serve as a high-quality initialization for a subsequent, more stable fine-tuning process. This hybrid approach effectively

balances the trade-off between rapid response and long-term stability, representing a promising direction for future research in sustainable model maintenance.

Finally, the scope of our empirical evaluation, while extensive, has inherent limitations regarding model scale and architectural diversity. Although we have validated our methods on large-scale reasoning models such as DeepSeek-R1, our experiments on non-reasoning-focused architectures were predominantly concentrated on models below the 13B parameter scale. Consequently, our observations regarding the universality of the "FFN Locus" and the low-rank nature of safety corrections, while robust within our testbeds, warrant further validation across an even broader spectrum of models. Furthermore, the generalization of our findings is naturally constrained by the scope of our training and evaluation datasets. While we have made significant efforts to broaden this scope by incorporating diverse models and attack datasets, future work should focus on scaling these experiments to even larger models (e.g., 70B+ parameters) and continually expanding the data diversity to ensure the broad applicability of the proposed techniques.

Statement

LLM Usage. Large Language Models (LLMs) were utilized solely to refine the text and correct grammatical errors; they were not involved in generating scientific claims or conceptual content.

Reproducibility Statement. We are committed to open science and reproducibility. Upon acceptance, all relevant code and data will be released to the research community to ensure our results can be fully reproduced.

Ethics Statement. This work focuses on improving the robustness and auditability of large language models against adversarial jailbreak attacks. Although the paper studies existing attack methods, all analysis is conducted for defensive evaluation.

Acknowledge

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

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

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.

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.

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.

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.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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*.

Xuechen Li, Ganda Jerfel, Smith Mirchandani, Banghua Wang, Yichong Li, Hitu Tang, Tianle Yu, Yan Zhou, Feodor Kirstein, S. Chaudhary, and 1 others. 2023. Alpacaeval: An automatic evaluator of instruction-following models. GitHub repository.

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 *Conference on Empirical Methods in Natural Language Processing*.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2023. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *ArXiv*, abs/2310.04451.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023. Locating and editing factual associations in gpt. *Preprint*, arXiv:2202.05262.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

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.

Mark Russinovich, Ahmed Salem, and Ronen Eldan. 2025. Great, now write an article about that: The crescendo multi-turn llm jailbreak attack. *Preprint*, arXiv:2404.01833.

Llama Team. 2024. Meta llama guard 2. https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard2/MODEL_CARD.md.

NovaSky Team. 2025a. Sky-t1: Fully open-source reasoning model with o1-preview performance in $450budget.\ https\ :\ //novasky-ai.github.io/posts/sky-t1.\ Accessed\ :\ 2025-01-09$.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. 2024. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. *Preprint*, arXiv:2402.08983.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. 2024. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. *Preprint*, arXiv:2401.06373.

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.

Guanghao Zhou, Panjia Qiu, Cen Chen, Hongyu Li, Jason Chu, Xin Zhang, and Jun Zhou. 2025. LSSF: Safety alignment for large language models through low-rank safety subspace fusion. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30621–30638, Vienna, Austria. Association for Computational Linguistics.

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.

A Experiment Setup

Baseline. **PPL** (Perplexity) assesses the uncertainty in a model's output and detects potentially harmful or nonsensical responses. **RLHF** (Reinforcement Learning from Human Feedback) refines an LLM using reinforcement learning, where human feedback on model outputs guides the reward function. **SafeDecoding**is a method designed to ensure safe and reliable outputs by applying constraints during the decoding process. **AED** (Adversarial Example Detection) identifies and filters adversarial inputs or examples that might cause a model to behave unpredictably or maliciously. The detailed baseline settings for each experiment are described in the appendix.

Jailbreak Method. **GCG**(Zou et al., 2023) (Greedy Coordinate Gradient) exploits gradient-based techniques to manipulate a model's output. **AutoDAN**(Liu et al., 2023) uses automatic techniques to generate adversarial inputs that can bypass content moderation mechanisms. **Pair**(Chao et al., 2023) involves crafting paired inputs that exploit vulnerabilities in the model's response generation. **PAP**(Zeng et al., 2024) apply the taxonomy to automatically generate interpretable persuasive adversarial prompts to jailbreak LLMs.

Target model. LLM as target models: LLama3-8B(Grattafiori et al., 2024), Qwen2-7B(Bai et al., 2023). Large Reasoning Models(LRMs) as target models: Deepseek-r1(DeepSeek-AI et al., 2025), QwQ(Team, 2025b), and Sky-T1(Team, 2025a).

Attack Datasets. Experiments utilized **Advbench**(Zou et al., 2023) as attack query datasets as a test dataset to validate the safety of HIAR.

Downstream Utility: General capabilities are measured by performance on standard benchmarks:

SCoT Source	Capability	Llama-3-8B ASR ↓	Qwen2-7B ASR ↓
None (Baseline)	–	51.3	48.1
Llama-2-7B	Weak	10.4	12.2
Qwen-72B	Mid	8.9	7.4
Llama-3-70B	Mid	8.0	7.6
GPT-o3	High	**7.3**	**6.6**

Table 3: Impact of SCoT source quality on internalization safety. Lower ASR is better.

Evaluator	ASR (%) ↓
GPT-4	7.3
Llama-Guard-3-8B	6.9
Human review	8.1

Table 4: Cross-judge agreement on HIAR outputs. The Cohen's κ between GPT-4 and human review is 0.87, indicating substantial agreement.

MMLU(Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021) and Alpaca-Eval(Li et al., 2023).

Evalution Metrics. Attack Success Rate (ASR) is used as the metric to evaluate the alignment security. Each query was repeated five times. Experiments evaluate the safety of responses using both LlamaGuard(Team, 2024) and GPT-4(OpenAI, 2023) evaluation. The detail shows in the appendix. **Accuracy (ACC)** is used for the multiple-choice, calculation tasks, and win rate.

A.1 Training Details

Supervised Fine-Tuning (SFT). For SFT, we randomly sample 20% of the dataset for training. The model is trained in fp16 with micro-batch size 1, global batch size 32, maximum sequence length 1024, and learning rate 1×10^{-5}.

RLHF / preference optimization baselines. For the RLHF-style baseline, we first apply SFT on 20% of the data using the same configuration as above. We then optimize on preference pairs built from concatenated prompts and positive/negative responses. In the PPO-style setting, the learning rate is 5×10^{-6}, the global batch size is 16, the entropy bonus is 0.0, and the PPO ratio epsilon is 0.2. For DPO-style comparisons in the main text, we use the same preference data but optimize with Direct Preference Optimization instead of PPO.

B Teacher Sensitivity and Evaluator Agreement

B.1 Open-source teachers and teacher–student mismatch

To test whether HIAR depends on a proprietary teacher, we regenerate SCoT supervision with teachers of different capability levels and families. Table 3 shows that HIAR remains effective even when the SCoT data are produced by open-weight teachers, while stronger or same-family teachers typically perform slightly better.

These results show that HIAR is not tied to a single closed-source teacher. Even a weaker open-source teacher substantially reduces ASR. We also observe a mild same-family advantage: for Llama students, Llama-3-70B outperforms Qwen-72B, while for Qwen students the reverse trend appears. This suggests that tokenizer and representation mismatch matter, but they do not determine whether internalization succeeds.

B.2 Evaluator agreement beyond GPT-4

Because GPT-4 is used as one of the main judges in the paper, we additionally re-evaluate a random sample of 2000 outputs using Llama-Guard-3-8B and human review. Table 4 shows that the resulting ASR estimates remain tightly aligned across judges.

The close agreement across GPT-4, Llama-Guard, and human review suggests that the observed robustness gains are not an artifact of a single proprietary evaluator.

C Comparative Analysis of SCoT Dataset Construction Strategies

C.1 Rationale and Objective

In the main paper, we detailed our methodology for constructing the Safety-oriented Chain-of-Thought (SCoT) dataset. A critical design choice was to employ an **Iterative Correction** strategy for harmful queries, rather than merely **Filtering** for SCoT instances that already yield a safe outcome. This appendix provides an empirical justification for that choice through a direct comparative experiment.

The objective is to demonstrate that the Iterative Correction strategy produces a more robust and diverse training dataset, leading to a model with superior safety alignment. Our hypothesis is that by explicitly exposing the model to its own initial harmful reasoning paths and then guiding it toward

a corrected, safe conclusion, we cultivate a more profound and generalizable safety capability. This contrasts with a filtering approach, which is susceptible to a form of "survivorship bias"—it only learns from instances where it was already successful, failing to learn how to recover from failure.

C.2 Experimental Design

To validate our hypothesis, we constructed two distinct datasets based on the same initial pool of 1,000 harmful queries sourced from the AdvBench benchmark. Two Llama-3-8B models were then trained using our HIAR framework, each on one of the datasets.

Dataset 1: The Filtering Strategy. For each harmful query, we prompted a powerful teacher model (GPT-4) to generate a full SCoT response. The final answer from this SCoT was then evaluated. If the answer was a safe and appropriate refusal, the entire ‘(query, SCoT)‘ pair was added to the dataset. If the answer was harmful or compliant with the harmful request, the entire data point was discarded. This process yielded a dataset of "naturally safe" reasoning chains.

Dataset 2: The Iterative Correction Strategy (Our Method). This strategy proceeds in two stages for harmful queries that initially elicit a harmful response.

1. **Initial Generation:** As with the filtering strategy, we first generate an SCoT response from the teacher model.
2. **Correction and Refinement:** If the initial response is deemed harmful, we do not discard it. Instead, we use the initial harmful thought process as part of a new, corrective prompt. The model is instructed: "The following reasoning led to a harmful conclusion. Analyze the flaws in this reasoning and generate a new, safe response following the SCoT principles." The successful, corrected SCoT chain then becomes the training instance for the original harmful query.

This strategy ensures that the dataset contains examples that directly address and rectify initial failure modes, creating a more challenging and comprehensive training environment.

C.3 Results and Analysis

The two models, *HIAR-Filter* and *HIAR-Iterative*, were evaluated on a held-out set of 500 adversarial prompts from AdvBench and HEX-PHI. The primary metric was Attack Success Rate (ASR), judged by a GPT-4 evaluator. A lower ASR indicates stronger safety alignment.

Table 5: Performance of HIAR trained on datasets constructed with Filtering vs. Iterative Correction strategies. ASR is evaluated on a held-out set of adversarial prompts. Lower ASR is better.

Model	Dataset Strategy	ASR (%) ↓
HIAR-Filter	Filtering	14.6%
HIAR-Iterative	**Iterative Correction**	**5.8%**

The results presented in Table 5 show a stark difference in performance. The model trained using the **Iterative Correction** strategy achieved an ASR of **5.8%**, which is consistent with the primary results reported in our paper. In contrast, the *HIAR-Filter* model was significantly more vulnerable, with an ASR of **14.6%**.

This outcome strongly supports our hypothesis. The analysis is twofold:

- **Enhanced Robustness:** The Iterative Correction dataset forces the model to learn not just what a safe reasoning process looks like, but also how to pivot away from an unsafe one. It internalizes the process of self-correction, making it more robust to novel or complex adversarial attacks that might otherwise trigger an initially flawed chain of thought.
- **Improved Generalization:** The filtering strategy implicitly narrows the distribution of training data to "easier" problems—those where the model's default reasoning is already aligned. The iterative strategy, by including corrected failures, presents a more diverse and adversarial set of examples. This broader distribution compels the model to develop a more fundamental and less superficial understanding of the underlying safety principles, which generalizes better to unseen attacks.

In conclusion, this comparative experiment validates the superiority of the Iterative Correction strategy for constructing a safety training dataset. It demonstrates that enabling a model to learn from its mistakes is a more effective path to robust safety alignment than only showing it examples of its successes.

D Preserving Downstream Capabilities on Large Reasoning Models

D.1 Objective and Motivation

A primary concern with safety alignment techniques is the potential for a degradation in the model's core cognitive abilities, a phenomenon often referred to as the "alignment tax." This risk is particularly acute for Large Reasoning Models (LRMs), whose primary value lies in their sophisticated capabilities for complex tasks such as mathematical reasoning and multi-step problem-solving.

This section aims to empirically demonstrate that our HIAR framework successfully imparts robust safety alignment without substantively compromising the downstream performance of a powerful LRM. Our objective is to show that HIAR provides a solution to the safety-performance trade-off, making it a viable method for aligning state-of-the-art reasoning models.

D.2 Experimental Setup

Target Model. We selected **DeepSeek-R1**, a state-of-the-art Large Reasoning Model, as the subject for this experiment due to its advanced performance on reasoning-intensive benchmarks.

Baselines. We compared the performance of HIAR against a comprehensive set of baseline alignment methods, including standard Supervised Fine-Tuning (SFT), Reinforcement Learning from Human Feedback (RLHF), and other contemporary safety techniques such AED, and Safedecoding. The original, unaligned DeepSeek-R1 model serves as the performance ceiling.

Evaluation Benchmarks. To assess a wide range of capabilities, we evaluated the models on three standard downstream benchmarks:

- **TruthfulQA:** Measures a model's ability to avoid generating common falsehoods and produce truthful answers.
- **GSM8K:** A dataset of grade school math word problems, testing mathematical and logical reasoning.
- **MMLU (Massive Multitask Language Understanding):** A comprehensive benchmark that evaluates general knowledge and problem-solving abilities across 57 diverse subjects.

Performance is measured by accuracy (ACC) on all three benchmarks.

D.3 Results and Analysis

The performance of HIAR and the baseline methods on the downstream tasks is summarized in Table 6.

Table 6: Downstream task performance (Accuracy, %) of various safety methods applied to the DeepSeek-R1 model. HIAR demonstrates minimal performance degradation compared to the original model and outperforms other safety alignment techniques.

Method	TruthfulQA ↑	GSM8K ↑	MMLU ↑
DeepSeek-R1	**73.5**	**92.8**	**87.8**
SFT	68.3	85.1	80.6
RLHF	70.1	88.6	82.1
PPLM	48.0	76.7	62.8
Self-Reminder	66.8	90.7	76.5
Retokenization	65.7	80.5	77.9
AED	60.2	89.6	83.0
Safedecoding	67.9	82.5	77.7
HIAR (Ours)	<u>72.5</u>	<u>92.0</u>	<u>86.6</u>

The results clearly indicate that **HIAR substantially preserves the model's original capabilities**.

1. **Minimal Performance Degradation:** Compared to the original DeepSeek-R1, HIAR exhibits only a marginal drop in performance across all three benchmarks: a decrease of 1.4% on TruthfulQA, 0.9% on GSM8K, and 1.4% on MMLU. This level of degradation is negligible for most practical applications and stands in stark contrast to the significant performance hits incurred by many other safety methods, such as PPLM, which sees a drop of over 25 points on TruthfulQA.
2. **Superiority Over Other Methods:** HIAR consistently outperforms all other safety alignment baselines. While methods like RLHF and Self-Reminder maintain reasonable performance on some tasks, they still fall short of the high fidelity preserved by HIAR. This demonstrates HIAR's effectiveness in achieving a more favorable balance in the safety-performance trade-off.

We attribute this strong performance preservation to two core principles of the HIAR framework:

- **Targeted Low-Rank Updates:** HIAR internalizes safety logic through targeted, low-rank updates to the FFN layers. This surgi-

cal approach modifies a very specific, low-dimensional "safety subspace" within the model's parameters, leaving the vast, high-dimensional space encoding general knowledge and reasoning abilities largely untouched.

- **Synergy with Reasoning:** The Safety-oriented Chain-of-Thought (SCoT) process that HIAR learns to internalize is, at its core, a reasoning procedure. By training the model to "think" more carefully about safety, we are reinforcing a form of structured reasoning that may be synergistic with, rather than antagonistic to, its existing problem-solving pathways.

In conclusion, this experiment provides strong evidence that HIAR is a highly effective method for aligning Large Reasoning Models, delivering robust safety without exacting a significant "alignment tax" on their critical downstream capabilities.

E More Derivation and Proof

We derive the closed-form update for one linear projection inside an MLP layer. Let $\mathbf{X}_l^q \in \mathbb{R}^{d_{\text{in}} \times n}$ denote the query-only input features to the linear projection $\mathbf{W}_l$, and let $\mathbf{X}_l^s \in \mathbb{R}^{d_{\text{in}} \times n}$ denote the corresponding SCoT-conditioned features. The columns of these matrices correspond to matched training examples or matched token positions. The original layer computes

$$\mathbf{Y}_l^q = \mathbf{W}_l \mathbf{X}_l^q + \mathbf{b}_l \mathbf{1}^\top, \qquad \mathbf{Y}_l^s = \mathbf{W}_l \mathbf{X}_l^s + \mathbf{b}_l \mathbf{1}^\top.$$

Define

$$\Delta \mathbf{X}_l = \mathbf{X}_l^s - \mathbf{X}_l^q, \qquad \Delta \mathbf{Y}_l = \mathbf{Y}_l^s - \mathbf{Y}_l^q.$$

Since the bias term cancels, we have

$$\Delta \mathbf{Y}_l = \mathbf{W}_l \Delta \mathbf{X}_l.$$

The goal of internalization is to find a parameter update $\Delta \mathbf{W}_l$ such that the query-only forward pass with the updated weights matches the SCoT-conditioned branch output:

$$(\mathbf{W}_l + \Delta \mathbf{W}_l)\mathbf{X}_l^q + \mathbf{b}_l \mathbf{1}^\top \approx \mathbf{Y}_l^s.$$

Equivalently,

$$\Delta \mathbf{W}_l \mathbf{X}_l^q \approx \Delta \mathbf{Y}_l = \mathbf{W}_l \Delta \mathbf{X}_l.$$

Therefore, the closed-form surrogate solves

$$\Delta \mathbf{W}_l^\star = \arg\min_{\Delta \mathbf{W}_l} \left\| \Delta \mathbf{W}_l \mathbf{X}_l^q - \mathbf{W}_l \Delta \mathbf{X}_l \right\|_F^2.$$

The minimum-norm solution is

$$\Delta \mathbf{W}_l^\star = \mathbf{W}_l \Delta \mathbf{X}_l (\mathbf{X}_l^q)^+.$$

If

$$\mathbf{X}_l^q = \mathbf{U}_l \mathbf{\Sigma}_l \mathbf{V}_l^\top,$$

then

$$(\mathbf{X}_l^q)^+ = \mathbf{V}_l \mathbf{\Sigma}_l^+ \mathbf{U}_l^\top,$$

and thus

$$\Delta \mathbf{W}_l^\star = \mathbf{W}_l \Delta \mathbf{X}_l \mathbf{V}_l \mathbf{\Sigma}_l^+ \mathbf{U}_l^\top.$$

For numerical stability, we use the ridge-regularized form

$$\Delta \mathbf{W}_l^\star = \mathbf{W}_l \Delta \mathbf{X}_l (\mathbf{X}_l^q)^\top \left(\mathbf{X}_l^q (\mathbf{X}_l^q)^\top + \lambda \mathbf{I} \right)^{-1}.$$

E.1 Zero-Intervention Constraints for General Capability Preservation

The derivation above only fits the SCoT-induced displacement on safety-critical examples. To preserve general capabilities, we additionally require the update to have minimal effect on general-purpose inputs. Let $\mathbf{X}_{l,\text{safe}}^q$ denote query-only features from safety examples, and let $\mathbf{X}_{l,\text{gen}}^q$ denote features from benign or general-task examples. For safety examples, the desired target is

$$\Delta \mathbf{W}_l \mathbf{X}_{l,\text{safe}}^q \approx \mathbf{W}_l \Delta \mathbf{X}_{l,\text{safe}}.$$

For general examples, we impose the zero-intervention constraint

$$\Delta \mathbf{W}_l \mathbf{X}_{l,\text{gen}}^q \approx \mathbf{0}.$$

This gives the weighted ridge objective

$$\Delta \mathbf{W}_l^\star = \arg\min_{\Delta \mathbf{W}_l} \left\| \Delta \mathbf{W}_l \mathbf{X}_{l,\text{safe}}^q - \mathbf{W}_l \Delta \mathbf{X}_{l,\text{safe}} \right\|_F^2 + \mu \left\| \Delta \mathbf{W}_l \mathbf{X}_{l,\text{gen}}^q \right\|_F^2 + \lambda \|\Delta \mathbf{W}_l\|_F^2.$$

Equivalently, define the augmented matrices

$$\mathbf{X}_{l,\text{aug}} = \left[\mathbf{X}_{l,\text{safe}}^q,\ \sqrt{\mu} \mathbf{X}_{l,\text{gen}}^q \right],$$

and

$$\mathbf{Y}_{l,\text{aug}} = \left[\mathbf{W}_l \Delta \mathbf{X}_{l,\text{safe}},\ \mathbf{0} \right].$$

Then the objective becomes

$$\Delta \mathbf{W}_l^\star = \arg\min_{\Delta \mathbf{W}_l} \|\Delta \mathbf{W}_l \mathbf{X}_{l,\text{aug}} - \mathbf{Y}_{l,\text{aug}}\|_F^2 + \lambda \|\Delta \mathbf{W}_l\|_F^2,$$

with the closed-form solution

$$\Delta \mathbf{W}_l^\star = \mathbf{Y}_{l,\text{aug}} \mathbf{X}_{l,\text{aug}}^\top \left(\mathbf{X}_{l,\text{aug}} \mathbf{X}_{l,\text{aug}}^\top + \lambda \mathbf{I} \right)^{-1}.$$

This construction explicitly encodes both goals: matching the SCoT-induced pre-activation displacement on safety examples and suppressing the update on general examples.

F experiment result on LRM

Model	Method	No Attack↓	GCG↓	AutoDAN↓	Pair↓
DS-r1	No Defense	8.51%	86.32%	82.12%	87.52%
	PPL	6.45%	**0.00%**	75.20%	65.52%
	RLHF	5.62%	17.02%	24.60%	28.35%
	AED	**0.00%**	9.50%	17.18%	28.17%
	Safedecoding	**0.00%**	3.28%	10.59%	18.65%
	HIAR(FT)	**0.00%**	2.90%	**6.29%**	**8.65%**
	HIAR(Cal)	**0.00%**	2.92%	6.98%	8.69%
SKY-T1	No Defense	11.7%	98.67%	84.16%	97.02%
	PPL	7.66%	**0.0%**	88.20%	77.76%
	RLHF	6.68%	12.83%	19.16%	26.67%
	AED	**0.0%**	14.57%	22.55%	36.98%
	Safedecoding	**0.0%**	12.63%	29.38%	9.75%
	HIAR(FT)	**0.0%**	3.59%	12.63%	10.27%
	HIAR(Cal)	**0.0%**	3.71%	**11.55%**	**9.71%**
QWQ	No Defense	**0.0%**	35.56%	23.80%	29.14%
	PPL	**0.0%**	**0.0%**	9.97%	17.61%
	RLHF	0.96%	3.40%	10.39%	18.36%
	AED	**0.0%**	3.90%	9.77%	16.55%
	Safedecoding	0.81%	2.23%	15.34%	**3.59%**
	HIAR(FT)	**0.0%**	1.39%	**4.57%**	5.14%
	HIAR(Cal)	**0.0%**	1.32%	4.80%	5.51%

Table 7: The ASR of applying alignment methods with various jailbreak methods suitable for LRMs.

G SCoT Prompt Template

```
<SCoT>
  <Analysis>Step-by-step harm
      ↪ identification</Analysis>
  <Strategy>Mitigation approach
      ↪ description</Strategy>
  <Revision>Detailed correction process<
      ↪ /Revision>
  <Validation>Final safety confirmation<
      ↪ /Validation>
</SCoT>

<FinalResponse>
{Corrected harmless response}
</FinalResponse>
```

Template Structure

Problem Context

`{Insert original question/request here}`

Harmful Response

`{Insert harmful response here}`

Harmless Response

`{Insert target harmless response here}`

You are now responding to the aforementioned issue and have already generated a harmful response. Now, it is necessary to generate a chain of thought to correct the harmful response into a target harmless response. The entire chain of thought should strictly follow and include the following parts. No responses should be generated for any other parts.

SCoT Generation Instructions

1. **Harm Analysis**
   - Potential biases/stereotypes
   - Dangerous/misleading information
   - Ethical violations
   - Social/cultural sensitivities
2. **Mitigation Strategy**
   - Neutralization techniques
   - Fact-checking mechanisms
   - Ethical alignment methods
   - Cultural adaptation approaches
3. **Revision Process**
   - Phrase-by-phrase analysis
   - Content filtering decisions
   - Tone/perspective adjustments
   - Safety verification checkpoints
4. **Final Validation**
   - Maintains original intent
   - Eliminates harmful content
   - Preserves natural flow
   - Aligns with safety guidelines

Output Format Requirements: