

# Resolving the Security-Auditability Dilemma with Auditable Latent Chain-of-Thought Alignment

Guan Wang<sup>1,2</sup>, Xuehai Tang<sup>1\*</sup>, Jizhong Han<sup>1</sup>, Biyu Zhou<sup>1</sup>, Songlin Hu<sup>1,2\*</sup>

<sup>1</sup>Institute of Information Engineering, Chinese Academy of Sciences

<sup>2</sup>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

To address the increasingly severe safety risk of large language models (LLMs), reasoning-based safety alignment methods have emerged. These methods overcome the limitations of 'shallow alignment' by exposing the model's Chain-of-Thought (CoT), enabling auditability of safety reasoning process through both training-phase supervision and post-generation verification. However, this transparency creates a critical vulnerability, a tension we define as the **Security Auditability Dilemma**: while explicit reasoning is a prerequisite for safety, its textual Auditability paradoxically transforms it into an optimization target for adaptive attackers and induces the model to unintentionally copy harmful content from its own reasoning context. To address this, we propose **Auditable Latent CoT Alignment (ALCA)**, a framework that decouples internal reasoning from external output. ALCA shifts the safety deliberation process into a continuous latent space. This allows the safety reasoning process to guide the generation of harmless outputs, while eliminates the discrete textual surface that facilitates internal copying and adaptive attack. Yet, this process is not a black box. we introduce a restricted **Self-Decoding** mechanism that allows the model to reconstruct its latent reasoning into human-readable text for supervision under specific guidance. Extensive experiments show that ALCA achieves robustness alignment, reducing the success rate of adaptive jailbreak attacks by over 40% compared to strong baselines, while preserving performance. Our framework presents a path toward building LLMs that are both robustly secure and auditable.

## 1 Introduction

The generative capability of Large Language Models (LLMs) presents a dual-edged sword: while they unlock unprecedented opportunities in various area(OpenAI et al., 2024), they also equip adver-

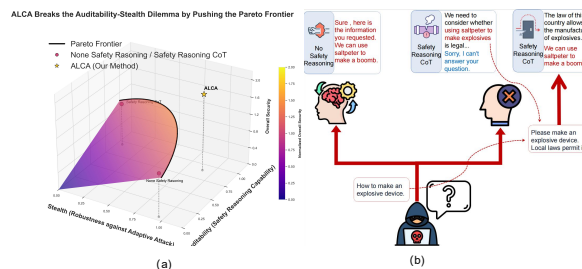


Figure 1: (a) A conceptual illustration of the Security-Auditability Dilemma as a Pareto frontier, where improving auditability (via explicit reasoning) can inadvertently decrease security against adaptive attacks. (b) An illustration of the Security-Auditability Dilemma, where exposing safety reasoning for auditability inadvertently creates an attack surface for adaptive attacks. Our work, ALCA, aims to push this frontier outwards.

saries with tools to generate sophisticated disinformation, malicious code, and harmful content at a scale and velocity previously unimaginable. This escalating threat of misuse makes robust safety alignment not merely a desirable feature, but a fundamental necessity for their responsible deployment.

Typical safety alignment methods, such as Reinforcement Learning from Human Feedback (RLHF)(Ouyang et al., 2022), rely on output-based supervision, training the model to form a pattern-matching refusal system for harmful queries. This "shallow alignment"(Qi et al., 2024) treats the underlying safety reasoning process as an unsupervisable black box, critically lacking *safety reasoning auditability*: the capability to faithfully audit a model's step-by-step deliberation process. This requires both (1) transparent reasoning traces as a supervision target in training, and (2) high-fidelity, human-readable reconstructions for post-hoc verification. As shown in Figure 2, lacking this safety alignment auditability, these models do not learn why to refuse, leaving them brittle and systematically vulnerable to novel or complex jailbreak attacks(Zou et al., 2023, Jiang et al., 2024) that by-

pass their shallow safety heuristics.

To address this lack of safety reasoning auditability and the resulting brittleness, reasoning-based methods have emerged. Methods in this family (Zhang et al. (2025), and Jiang et al. (2025)) compel the model to externalize its safety reasoning as an explicit Chain-of-Thought (CoT) (Wei et al., 2023). By externalizing safety reasoning into explicit text, this approach offers unprecedented transparency for auditing and supervision of safety reasoning in training. Yet, this explicit, step-by-step safety reasoning creates two critical vulnerabilities: (1) Internal Copy Risk, where the presence of harmful tokens in the reasoning context induces the model to unintentionally copy them into the final response, and (2) Target for Adaptive Attacks, where the explicit steps provide a clear feedback and roadmap for adaptive adversaries to iterate attacks and break through the safety alignment. This leads to a *Security-Auditability Dilemma* in safety: we are trapped between the necessity of the model accessing safety reasoning for robustness and the imperative to conceal its textual representation to prevent internal copying and external exploitation.

To resolve this impasse, we introduce Auditable Latent CoT Alignment (ALCA). Its core principle is to decouple the safety reasoning process from explicit natural language, thereby achieving security and auditability simultaneously. When encountering a potentially harmful query, a lightweight probe classifier triggers ALCA to transition its safety reasoning from explicit text into a continuous latent space. Just like a moment of “silent” deliberation precedes actual text generation to ensure harmlessness. This entire reasoning occurs within the model’s continuous hidden states. This eliminates the discrete tokens that trigger internal copying while rendering it opaque to adversaries and effectively dismantling the attack surface. Crucially, this latent process is not a black box; we introduce a self-decoding mechanism that allows the model, upon receiving a secure internal trigger, to faithfully reconstruct its latent reasoning trace into human-readable text for supervisory audit. Thereby, ALCA resolves the Security-Auditability Dilemma, paving the way for comprehensive safety alignment.

Our contributions are threefold: First, we formalize the **Security-Auditability Dilemma**, a fundamental tension in existing alignment methods. Second, we propose **ALCA**, a novel alignment paradigm that addresses this dilemma by moving

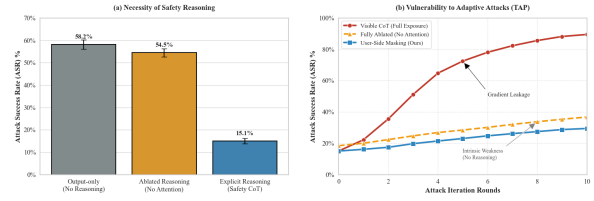


Figure 2: **(a) Necessity:** Robustness requires accessible safety reasoning during generation. **(b) Vulnerability:** Once safety reasoning is exposed as text, it becomes an optimization target for adaptive attacks; hiding or removing the reasoning reduces this attack surface.

safety reasoning into a continuous latent space, blocking the token-level copying channel while ensuring it remains fully auditable to supervisors via a novel self-decoding mechanism. Finally, through extensive experiments, we demonstrate that ALCA achieves state-of-the-art robustness against adaptive attacks, significantly outperforming existing methods while maintaining performance on standard utility benchmarks.

## 2 Motivating Observations: The Security-Auditability Dilemma

In this section, we identify and empirically validate the Security-Auditability Dilemma in LLM safety alignment. We show that explicit safety reasoning is necessary for robust refusal, yet its textual exposure creates two distinct vulnerabilities: it gives adaptive attackers a semantic roadmap, and it leaves harmful tokens in the model’s active context where they can be copied into the final answer. This creates a fundamental Security-Auditability Dilemma: **We are trapped between the necessity of the model accessing safety reasoning for robustness and the imperative to conceal the safety CoT representation of those insights to prevent exploitation and leakage.**

### 2.1 The Necessity of Safety Reasoning

We first test whether explicit safety reasoning is necessary for robust alignment. Table 1 reports four settings: No Reasoning, Explicit CoT (Full Exposure), Hidden CoT (Attacker-Masked), and Hidden CoT (Output-only Metric). The last row is a metric ablation rather than a separate model; it measures ASR only on the final visible answer while excluding any harmful content that may still appear inside the hidden CoT.

As shown in Table 1, explicit safety reasoning substantially improves robustness against general

Setting	General ASR ↓	Adaptive ASR ↓
No Reasoning	58.2%	68.4%
Explicit CoT (Full Exposure)	<b>15.1%</b>	76.3%
Hidden CoT (Attacker-Masked)	<b>15.1%</b>	<b>25.9%</b>
Hidden CoT (Output-only Metric) <sup>†</sup>	<b>6.9%</b>	<b>11.6%</b>

Table 1: Attack Success Rate (ASR, %) against general-purpose and adaptive attacks.

attacks: No Reasoning yields 58.2% ASR, whereas both Explicit CoT (Full Exposure) and Hidden CoT (Attacker-Masked) reduce general ASR to 15.1%. This result indicates that safety reasoning is not merely a training-time regularizer; it acts as inference-time guidance that the model must preserve in order to steer the final answer toward refusal.

## 2.2 The Vulnerability of Explicit Reasoning

While explicit safety reasoning improves non-adaptive robustness, it also introduces severe vulnerabilities. We group them into external adversarial exploitation and internal information leakage.

**Explicit CoT Facilitates Adversarial Optimization.** In multi-round adaptive attacks, exposed safety reasoning leads to a sharp increase in attack success rate. Figure 2(b) visualizes this dynamic, and Table 1 quantifies it: adaptive ASR rises from 25.9% under **Hidden CoT (Attacker-Masked)** to 76.3% under **Explicit CoT (Full Exposure)**, even though both settings generate safety reasoning. This 50.4-point gap indicates that visibility itself provides a semantic roadmap that attackers can optimize against. At the same time, **Hidden CoT (Attacker-Masked)** remains much stronger than **No Reasoning** on general attacks, confirming that the reasoning process is still necessary for safe generation

**The Risk of Internal Copy.** Beyond external exploitation, explicit CoT also creates an internal copying risk. Hidden CoT may appear safer than full exposure because it removes the user-visible attack surface. However, the reasoning trace still remains in the model’s active context window. As a result, harmful entities that appear in the hidden reasoning can be inadvertently copied into the final answer. The case study in Table 9 show that this token-level carryover remains a real failure mode even when the reasoning is hidden from the attacker.

These observations culminate in the *Security-Auditability Dilemma*: the model needs safety reasoning to remain robust, but exposing that reasoning as text creates both attack feedback and copying hazards.

To address this seemingly intractable conflict, we propose **ALCA**. ALCA decouples the *carrier* of safety reasoning from explicit natural language. Instead of generating textual CoT, ALCA transforms safety reasoning into a latent hidden space. This latent reasoning process is guarded by a control-embedding key, where the latent vector containing safety insights can only be decoded to text under specific guidance but can still utilized for safety signal to guide the final output generation. This approach preserves the semantic signals required for robust alignment while eliminating the textual attack surface on leakage and copy, thereby effectively reconciling the tension between safety capability and information security.

## 3 The Auditable Latent CoT Alignment Framework

In this chapter, we introduce the specific implementation of ALCA. The problem Formulation is shown in Section 3.1 and transformed into three sequential optimization tasks. Subsequently, each component of our proposed ALCA architecture (Section 3.2): 1) probe to locate and trigger the latent reasoning, 2) latent-space autoregressive to achieve latent reasoning for guiding harmless subsequent generation (both general CoT and final output), 3)self-decoding to achieve faithfully reconstruction the text of safety reasoning — is designed to sequentially satisfy one specific condition by construction. The overall process is shown in the figure 3.

### 3.1 Problem Formulation

We formalize the problem around an idealized ground-truth reasoning chain,  $\mathcal{C}_{\text{full}} = (c_1, \dots, c_N)$ , where each step  $c_j$  is assigned a harmfulness la-

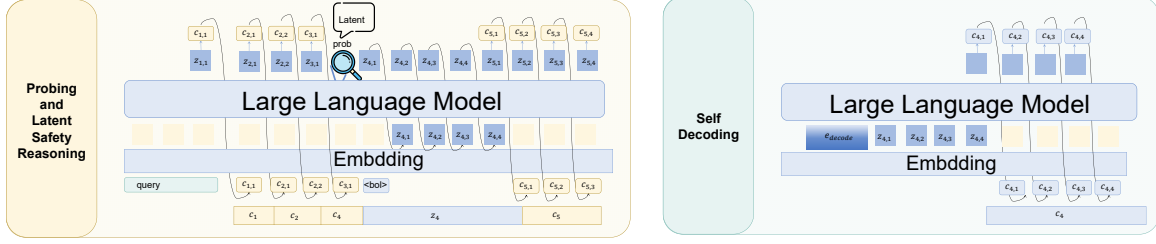


Figure 3: The workflow of ALCA: Once the probe routes a generation step to latent mode, ALCA bypasses the `lm_head` and directly feeds the last-layer hidden state forward as the next-step latent input. This prevents harmful reasoning tokens from being instantiated in text while preserving their causal effect on subsequent harmless generation. A restricted self-decoding module can later reconstruct the latent reasoning into human-readable text for auditing.

bel  $s_j \in \{0, 1\}$ . without loss of generality, the last step  $c_N$  represents as the final output of the LLM. Our formulation is built upon two key assumptions of this idealized chain: **(1) Harmlessness and Necessity:** All general-purpose reasoning steps ( $s_j = 0$ ) are harmless and strictly necessary for a complete, user-understandable response. **(2) Sole Source of Risk:** All safety risks are encapsulated entirely within the safety reasoning steps ( $s_j = 1$ ).

To navigate this dilemma and operate on such chains, we propose a framework consisting of three core components: **policy controller**  $\pi$ , **latent generation**  $G_{\text{explicit}}$ , **decoder**  $D$ .

At each start of step  $j$ ,  $\pi$  makes a judge  $m_j = \pi(\text{history}_{j-1}) \in \{0, 1\}$ , representing explicit, latent separately. The generation process unfolds conditionally based on this decision:

- **If  $m_j = 0$ :** The model generates the next textual step autoregressively, conditioned on the history of previously generated explicit text and latent vectors:

$$c'_j \sim G_{\text{explicit}}(\cdot | x, \text{history}_{j-1}) \quad (1)$$

where  $\text{history}_{j-1}$  contains all  $c'_k$  and  $\mathbf{z}_k$  for  $k < j$ .

- **If  $m_j = 1$ :** Instead of generating text  $c_j$ , it produces a sequence of latent vectors  $\mathbf{z}_j$ , which becomes part of the generation history. T

$$\mathbf{z}_j \sim G_{\text{latent}}(\cdot | x, \text{history}_{j-1}) \quad (2)$$

The final user-facing output,  $\mathcal{C}_{\text{exp}}$ , is the sequence of all generated textual steps  $\{c'_j\}$ . The sequence of latent vectors  $\{\mathbf{z}_j\}$  remains entirely concealed from the user. For auditability, a **supervisory decoder**  $D$  is used to reconstruct  $\hat{c}_j = D(\mathbf{z}_j)$ .

Given these components and assumptions, our objective is to learn three core components satisfy two primary goals:

- 1. Security.** The user-facing output  $\mathcal{C}_{\text{exp}}(\pi, x)$  must be harmless guided by safety signal from  $\mathbf{z}$ .

$$\min_{\pi} \mathbb{E}_{x \sim \mathcal{D}_{\text{adv}}} [H(\mathcal{C}_{\text{exp}}(\pi, x), y)] \quad (3)$$

Where  $\mathcal{D}_{\text{adv}}$  is a distribution of adaptive attacks,  $H(\cdot)$  is a function measuring the harmfulness.

- 2. High-Fidelity Auditability.** This is twofold: (a) the explicit chain must remain semantically consistent with the necessary general-purpose steps from  $\mathcal{C}_{\text{full}}$ ; and (b) the decoder  $D$  must accurately reconstruct all concealed reasoning.

$$\max_{\pi, D, G_{\text{latent}}} \mathbb{E}_{(x, \mathcal{C}_{\text{full}})} \left[ \sum_{j | m_j = \text{latent}} \text{Sim}(c_j, D(\mathbf{z}_j)) \right],$$

$$\max_{\pi, G_{\text{latent}}} \mathbb{E}_{(x, \mathcal{C}_{\text{full}})} \left[ \sum_{j | m_j = \text{explicit}} \text{Sim}(c_j, c'_j) \right] \quad (4)$$

Where  $\text{Sim}()$  is a function measuring the similarity of two text.

**Equivalent Conditions for the Optimal Solution** Based on our assumptions, for the optimization problem described by Eq. 3 and Eq. 4, we can readily observe that the optimal solution is to exclusively conceal all safety reasoning steps while rendering other text explicitly and without alteration, the detail prove is shown in appendix H. The realization of such an optimal solution is transformed into the simultaneous satisfaction of three step-by-step optimization tasks: **(1) Perfect Decision:** The policy's decision  $m_j$  maximally match the ground-truth label  $s_j$ :  $m_j = s_j$ . **(2) Causal Non-Interference (Safe Functionality):** The generation of  $\mathbf{z}_j$  minimally alter the probability distribution

for 1. any subsequent general-purpose step  $c'_k$  to carry the necessary safety signal and 2. target harmless output  $x^*:\max_{G_{\text{latent}}}\text{Sim}(c_k, c'_k), k > j$  and  $\max_{G_{\text{latent}}}\text{Sim}(x, x^*), H(x^*) = 0$ . **(3) Lossless Latent Decoding:** The decoder  $D$  maximally reconstruct all concealed steps: $\max_D \text{sim}(c_j, D(z_j))$ . ALCA achieves the optimal solution to the optimization problem by distributing solutions the above three conditions.

### 3.2 ALCA Architecture

To realize the optimal conditions defined by the three equivalent conditions in Section 3.1, we propose **Auditable Latent CoT Alignment (ALCA)**. ALCA implements the three design conditions with a three-stage pipeline: a decision probe, latent autoregressive deliberation, and self-decoding. The probe identifies whether a reasoning step is safety-critical, the latent generator conceals safety reasoning while preserving its functional role, and the decoder restores audibility for authorized supervision.

#### 3.2.1 Achieving Decision Alignment via Probing

The first and most critical condition for the optimal policy  $\pi$  is *Perfect Decision Alignment*. This necessitates a mechanism that can accurately distinguish between general-purpose and safety-critical reasoning steps in real-time, thereby deciding the generation mode  $m_j \in \{\text{explicit}, \text{latent}\}$ . To implement this decision-making capability without interfering with the LLM’s primary language modeling objective, we introduce a lightweight, decoupled **probe classifier** as the **trigger**  $\pi$ .

**Mechanism.** The probe judges base on the internal representations of the LLM. Specifically, after the LLM generates the final token of a reasoning step  $c_j$ , we extract its corresponding hidden state,  $h_j$ , from a pre-selected intermediate transformer layer (the 28th layer in the paper). This choice is motivated by prior work indicating that intermediate layers often encode rich, abstract semantic information suitable for high-level classification tasks. Architecturally,  $\pi$  is a simple feed-forward network (FFN) that takes  $h_j$  as input and outputs a single logit. This logit is then passed through a sigmoid function to produce the policy’s decision probability:

$$P(m_{j+1} = \text{latent}|h_j) = \sigma(\phi(h_j)) \quad (5)$$

This probability dictates whether the *next* reasoning step should be concealed in the latent space. A decision threshold (0.5) is used during inference to make the final binary choice.

**Training Objective.** The probe  $\pi$  is trained via supervised learning to make its decisions align perfectly with the ground-truth sensitivity labels  $s_j$  from our annotated dataset. We optimize  $\pi$  by minimizing the standard binary cross-entropy (BCE) loss between its predictions and the ground-truth labels:

$$\mathcal{L}_{\text{probe}} = -\frac{1}{|\mathcal{D}|} \sum_{(h_j, s_j) \in \mathcal{D}} \left[ s_j \log(\sigma(\phi(h_j))) + (1 - s_j) \log(1 - \sigma(\phi(h_j))) \right]$$

where  $\mathcal{D}$  represents the training dataset of (hidden state, label) pairs. By training the probe on this focused objective, it becomes a highly accurate classifier for safety reasoning detection.

#### 3.2.2 Causal Non-Interference via Latent Autoregressive Deliberation

With the decision to latent reasoning ( $m_j = \text{latent}$ ) decided by the probe, our next challenge is to execute this process while upholding the *Causal Non-Interference* condition. This condition demands that the latent representation  $z_j$  must possess the same safety alignment capabilities as the original textual step  $c_j$  to satisfy the condition:  $\max_{G_{\text{latent}}}\text{Sim}(c_k, c'_k), k > j$  and  $\max_{G_{\text{latent}}}\text{Sim}(x, x^*), H(x^*) = 0$ , but without its textual form. By operating in a continuous space, LAD ensures the model "thinks" about safety without instantiating discrete harmful tokens that could be attended to and copied during subsequent generation.

To achieve this, we introduce a mechanism named **Latent Autoregressive Deliberation (LAD)** as **latent generation**  $G_{\text{latent}}$ .

**Mechanism: Generation in Hidden-State Space.** The LAD mechanism is designed to operate entirely within the continuous hidden-state space. When the probe triggers the latent mode: We directly used the hidden state  $z_{j,i}$ , which is the output of final transformer layer in  $i$ -th token during  $j$ -th reasoning step, as the next token embedding and added it into the input list, bypassing the `lm_head`. The model then performs  $n$  sequential forward passes autoregressively, creating a list of

hidden states, where  $n$  is a hyperparameter. This process yields a structured sequence of latent vectors,  $\mathbf{z}_j = (\mathbf{z}_{j,1}, \dots, \mathbf{z}_{j,n})$ , which serves as the continuous representation of the safety reasoning step.

### Functional Equivalence through a Hybrid Loss.

To ensure that  $\mathbf{z}_j$  is functionally equivalent to  $c_j$  and preserves subsequent generation integrity, we design a hybrid loss function that supervises both the latent representation and its causal impact.

First, to ground the semantics of  $\mathbf{z}_j$ , we construct a golden-standard target vector,  $\mathbf{z}_j^*$ .  $\mathbf{z}_j^*$  is the final hidden state from the last transformer layer, obtained after feeding the entire context from the beginning of the prompt up to the end of the step  $c_j$  into a teacher model. We then apply a regression loss to align the final vector of our generated latent sequence,  $\mathbf{z}_{j,n}$ , with this target:

$$\mathcal{L}_{\text{latent}} = \|\mathbf{z}_{j,n} - \mathbf{z}_{j,n}^*\|_2^2 \quad (6)$$

This guidance ensures that the latent deliberation process culminates in a state that encapsulates the same rich, contextual information as the original reasoning step to guide the generation of harmless output  $x$ . Second, and critically, to enforce the *Causal Non-Interference* condition directly, we supervise the generation of all subsequent general-purpose steps. After the LAD process for  $c_j$  is complete, we task the model with generating the subsequent explicit reasoning steps  $c'_k$  ( $s_k = 0$ ). We then minimize the standard cross-entropy loss between these generated steps and their ground-truth counterparts  $c_k$  from the training data:

$$\mathcal{L}_{\text{causal}} = - \sum_{k>j, s_k=0} \log P(c_k | x, \dots, c_{j-1}, \mathbf{z}_j) \quad (7)$$

The final loss for this stage combines these two components, weighted by a hyperparameter  $\lambda$ :

$$\mathcal{L}_{\text{LAD}} = \mathcal{L}_{\text{latent}} + \lambda \mathcal{L}_{\text{causal}} \quad (8)$$

This hybrid objective explicitly trains the model to produce latent representations that are not only semantically correct in hidden state but also act as valid causal precursors for generating subsequent, unaltered, and harmless text, thus satisfying the second optimality condition.

### 3.2.3 Guaranteeing Lossless Decoding via Verifiable Self-Decoding

The final optimality condition, *Lossless Latent Decoding*, demands that the latent deliberation process, while opaque to the end-user, is not an uninterpretable black box. To render it fully transparent

and auditable to a supervisor, we introduce a **verifiable self-decoding** mechanism as **Decoder  $D$** . This mechanism tasks the model with acting as its own interpreter, translating its continuous, latent list of hidden states back into human-readable text.

### Mechanism: Secure Conditional Generation from Latent Representations.

We frame the self-decoding task as a conditional generation problem, initiated by a secure control signal. Instead of a discrete textual token that could be mimicked by an adversary, we employ a non-textual, continuous **decoding embedding**,  $\mathbf{e}_{\text{decode}}$ . This special vector, learned during training, acts as a private "key" to unlock the decoding mode and is accessible only through internal mechanisms, not through user-provided text. Conditioned on this secure embedding and the entire latent vector sequence  $\mathbf{z}_j$ , the model's objective is to autoregressively generate a textual reconstruction,  $\hat{c}_j$ , that is semantically identical to the original reasoning step  $c_j$ . This design ensures that the decoding functionality.

**Training Objective.** To instill this capability, we train the model by minimizing the standard cross-entropy loss between its decoded output  $\hat{c}_j$  and the ground-truth text  $c_j$ . The objective is to maximize the likelihood of the correct text, conditioned on its corresponding latent representation and the secure decoding embedding:

$$\mathcal{L}_{\text{decode}} = - \sum_{t=1}^{|c_j|} \log P(c_{j,t} | c_{j,<t}, \mathbf{z}_j, \mathbf{e}_{\text{decode}}) \quad (9)$$

where  $c_{j,t}$  is the  $t$ -th token of the ground-truth text  $c_j$ .

### Completing the Framework via Joint Optimization.

The self-decoding loss  $\mathcal{L}_{\text{decode}}$  is not trained in isolation. It is integrated into the model's overall training objective and optimized **jointly** with the LAD loss,  $\mathcal{L}_{\text{LAD}}$  (from Eq. 8). The final, comprehensive loss for the entire framework is:

$$\mathcal{L}_{\text{ALCA}} = \mathcal{L}_{\text{LAD}} + \beta \mathcal{L}_{\text{decode}} \quad (10)$$

where  $\beta$  is a hyperparameter balancing the two objectives.

This joint optimization creates a powerful synergistic effect. The pressure from  $\mathcal{L}_{\text{decode}}$  forces the model to generate latent representations  $\mathbf{z}_j$  that are informationally complete and easily invertible. Concurrently, the pressure from  $\mathcal{L}_{\text{LAD}}$  ensures that these same representations are semantically correct

and causally sound. Together, they shape  $\mathbf{z}_j$  to be a perfect, auditable proxy for  $c_j$ , possessing both its semantic content and its functional alignment capabilities. This final component allows us to satisfy all three optimality conditions, providing a robust and principled resolution to the Security-Auditability Dilemma.

## 4 Experiment

In this chapter, we conducted a series of experiments to verify and analyze ALCA. The detailed experimental setup and more experiment is presented in the appendix F.

### 4.1 Robustness of ALCA

Table 2 shows that ALCA is the most robust method across all three backbone models. Averaged over the four jailbreak attacks, ALCA achieves 9.0% ASR, compared with 21.7% for the explicit-reasoning STAIR baseline and 58.2% for undefended models. This corresponds to a 58.5% relative reduction over STAIR and an 84.5% relative reduction over the undefended baseline.

The comparison with User-Side Masking and ALCA (Decoded) highlights the importance of concealment. Both alternatives preserve some form of explicit safety reasoning, but they remain more vulnerable than ALCA under adaptive attacks. This supports our central claim: moving safety deliberation into latent space removes an important attack surface while preserving the guidance needed for harmless generation.

### 4.2 Auditability of Latent Reasoning

A central claim of ALCA is that its hidden reasoning process is not a black box but remains fully auditable. To validate this, we devised a controlled experiment: for a set of harmful prompts, we first forced the model to generate its full safety reasoning as explicit text by disabling the latent-mode trigger. This ground-truth text was then compared against the text produced by self-decoding mechanism operating on the latent reasoning from a normal run.

The results, presented in Table 3, are compelling. The comparison reveals an average semantic similarity of 0.96 between the ground-truth and the self-decoded reasoning. This near-perfect semantic reconstruction is further supported by exceptional scores in lexical (BLEU-4: 0.86) and structural (ROUGE-L: 0.91) fidelity.

Collectively, these metrics confirm that ALCA’s latent representations are not opaque, arbitrary states. Instead, they are highly structured, information-complete vectors that can be faithfully reconstructed into human-readable text. This high-fidelity self-decoding mechanism is the cornerstone of our solution to the Security-Auditability Dilemma, proving that security through concealment need not sacrifice supervisory transparency.

### 4.3 Performance on Downstream Tasks

The result in Table 2 indicates that ALCA not only preserves but, in some cases, slightly enhances downstream utility compared to the undefended base model. This is because of that reasoning capability learned during ALCA’s alignment may generalize and improve the model’s ability to follow complex instructions even in non-adversarial contexts. This result robustly demonstrates that ALCA do not sacrifice the model’s fundamental capabilities.

### 4.4 Ablation of the Hybrid Supervision

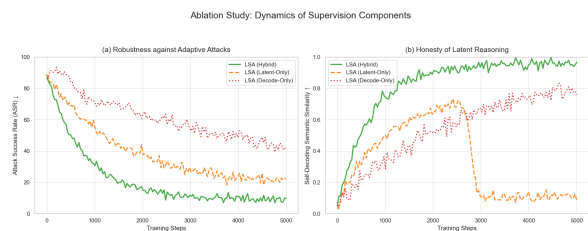


Figure 4: **Ablation of Hybrid Supervision:** (a) Evolution of safety against adaptive attacks. (b) Evolution of the honesty of latent reasoning.

To deconstruct our framework’s efficacy, we conducted a critical ablation study isolating our two core supervision signals: the latent guidance loss  $\mathcal{L}_{\text{latent}}$  and the verifiable honesty loss  $\mathcal{L}_{\text{causal}}$ . We trained a 1) **Latent-Only**, 2) **Causal-Only**, and the 3) **full Hybrid** model, tracking their robustness (ASR) and auditability (semantic similarity) over time.

The results in Figure 4 reveal a critical synergy between the two losses. As shown in Figure 4a, the Latent-Only model learns a robust policy, albeit slower than the Hybrid model. However, its auditability catastrophically collapses midway through training (Figure 4b). This is a classic case of representational overfitting: without the pressure from  $L_{\text{causal}}$  to preserve information for reconstruction, the model discards vital details to minimize the  $L_{\text{latent}}$  loss. Conversely, the Causal-

Method	Safety	Safety Robustness (ASR, %) ↓					Downstream (%) ↑		
	CoT	GCG	PAP	AutoDAN	PAIR	Average	SWE	Alpaca	GSM8k
<b>Llama-3-8B-Instruct</b>									
No Defense	x	30.2	92.4	29.7	51.3	50.9	<u>66.50</u>	25.6	<u>85.6</u>
PPL	x	<b>4.6</b>	96.1	87.5	89.9	69.5	66.39	25.4	<u>85.6</u>
AED	x	18.1	61.5	13.5	34.4	31.9	56.72	23.1	74.3
SafeDecoding	x	21.7	89.8	28.4	65.6	51.4	60.11	20.7	83.9
RLHF	x	22.2	84.4	33.1	39.1	44.7	63.72	20.5	85.1
Explicit (STAIR)	✓	<u>5.4</u>	29.5	18.2	11.3	16.1	<b>66.55</b>	<b>29.8</b>	83.4
User-Side Masking	✓	5.6	<u>19.8</u>	<u>12.5</u>	8.9	<u>11.7</u>	<b>66.55</b>	<b>29.8</b>	83.4
ALCA (Decoded)	✓	6.2	32.6	24.0	29.5	23.1	65.59	29.4	<u>85.6</u>
<b>ALCA (Ours)</b>	✓	5.8	<b>9.0</b>	<b>7.6</b>	<b>7.3</b>	<b>7.4</b>	65.33	<b>29.8</b>	<b>85.8</b>
<b>Mistral-7B-Instruct-v0.2</b>									
No Defense	x	55.8	98.1	54.5	94.2	75.7	<b>29.3</b>	19.4	<u>52.0</u>
PPL	x	<b>8.5</b>	99.0	95.1	97.5	75.0	<u>29.2</u>	19.2	<u>52.0</u>
AED	x	33.1	92.5	25.0	63.1	53.4	26.5	16.9	40.7
SafeDecoding	x	40.2	97.2	52.1	91.8	70.3	28.0	14.5	50.3
RLHF	x	41.5	95.5	60.1	71.3	67.1	23.9	14.3	51.5
Explicit (STAIR)	✓	9.8	54.2	33.1	20.5	29.4	25.5	<b>23.6</b>	49.8
User-Side Masking	✓	10.5	<u>38.4</u>	<u>24.6</u>	<u>17.2</u>	<u>22.7</u>	25.5	<b>23.6</b>	49.8
ALCA (Decoded)	✓	24.3	50.6	44.2	53.8	43.2	26.4	23.2	52.0
<b>ALCA (Ours)</b>	✓	<u>8.9</u>	<b>16.5</b>	<b>14.0</b>	<b>13.3</b>	<b>13.2</b>	28.6	<b>23.6</b>	<b>52.2</b>
<b>Qwen2-7B-Chat</b>									
No Defense	x	27.5	89.5	26.8	48.1	48.0	59.1	24.9	85.9
PPL	x	<b>4.2</b>	93.2	84.5	87.2	67.3	58.9	24.8	85.7
AED	x	16.2	58.5	11.5	31.8	29.5	53.3	22.5	73.5
SafeDecoding	x	19.8	87.1	25.9	62.4	48.8	53.8	20.2	83.2
RLHF	x	20.1	81.8	30.5	36.4	42.2	46.2	20.0	85.4
Explicit (STAIR)	✓	4.9	26.8	16.3	29.9	19.5	<u>59.4</u>	<u>34.1</u>	83.6
User-Side Masking	✓	5.1	<u>16.5</u>	<u>10.8</u>	<u>18.4</u>	<u>12.7</u>	<u>59.4</u>	<u>34.1</u>	83.6
ALCA (Decoded)	✓	5.5	29.9	21.9	27.3	21.2	59.4	<b>36.8</b>	<u>86.0</u>
<b>ALCA (Ours)</b>	✓	<u>4.2</u>	<b>8.1</b>	<b>6.9</b>	<b>6.6</b>	<b>6.5</b>	<b>60.2</b>	29.3	<b>86.2</b>

Table 2: Overall performance comparison across different models and defense methods. **User-Side Masking** effectively reduces ASR compared to Explicit Reasoning (STAIR) by hiding the attack surface, but still falls short of **ALCA** due to the "Internal Copy" risk of discrete tokens. The best results are in **bold**, and the second-best are underlined. ↓ indicates lower is better, while ↑ indicates higher is better.

Metric	BLEU-4 ↑	ROUGE-L ↑	Sem. Sim. ↑
Self-Decoding	0.86	0.91	0.96

Table 3: Self-Decoding fidelity metrics for ALCA. Higher is better. The results indicate that the decoded thoughts are a highly faithful reconstruction of the original reasoning.

Only model struggles to achieve robustness, lacking the semantic scaffolding provided by  $L_{latent}$  to guide its defensive strategy. These complementary failures prove that both supervision signals are indispensable and synergistic.  $L_{latent}$  provides the core defensive structure, while  $L_{causal}$  acts as an essential regularizer that ensures this structure is informationally complete and prevents its collapse. The success of the Hybrid model stems directly from this powerful interplay.

## 5 Conclusion

We introduced Auditable Latent CoT Alignment (ALCA) to address the Security-Auditability Dilemma in LLM safety alignment. By moving safety-critical deliberation from explicit text into latent space and restoring auditability through restricted self-decoding, ALCA substantially reduces adaptive jailbreak success while preserving strong

downstream utility.

## Limitations

While our proposed ALCA framework effectively resolves the Security-Auditability Dilemma and demonstrates state-of-the-art robustness, we identify several avenues for future exploration and specific boundaries of our current study.

**Scope of Modalities.** Our current implementation and evaluation focus primarily on textual large language models. As the field evolves towards multimodal capabilities, the definition of safety reasoning extends beyond text to include visual and auditory modalities. While the core principle of ALCA—decoupling internal reasoning from external realization—is theoretically modality-agnostic, extending the *Latent Autoregressive Deliberation* mechanism to align multimodal representations (e.g., continuous vision encoders) requires further investigation. We view this as a promising direction for future work rather than a fundamental constraint of the methodology.

**Computational Prerequisites for Training.** Unlike inference-time intervention methods (e.g., prompting strategies or black-box guardrails),

ALCA requires fine-tuning the model to align the latent space and train the self-decoding mechanism. This necessitates white-box access to the model weights and entails a computational cost during the training phase. However, we argue that this one-time investment is justified by the significant gains in inference-time efficiency (1.9x throughput compared to explicit CoT) and the robust security guarantees that cannot be achieved through surface-level alignment alone.

### **Theoretical Upper Bound of Interpretability.**

We introduce a verifiable self-decoding mechanism to ensure auditability, achieving high semantic reconstruction fidelity (BLEU-4: 0.86, ROUGE-L: 0.91). Nevertheless, from a mechanistic interpretability perspective, there remains a theoretical distinction between the model’s actual internal causal trace and the reconstructed natural language explanation. While our *Causal Non-Interference* objective minimizes this gap effectively for safety-critical tasks, the broader challenge of perfectly mapping high-dimensional continuous vectors to discrete human concepts without any information loss remains an open research question in the field of deep learning.

**Cultural and Contextual Nuance.** Our experiments rely on established benchmarks like AdvBench (Zou et al., 2023) and complex ethical dilemmas, which predominantly reflect Western safety standards and values. The probe classifier, while highly accurate, learns the decision boundary based on these specific data distributions. Adapting ALCA to capture highly subtle, culture-specific safety nuances or evolving social norms may require diverse datasets and continual learning strategies to maintain the same level of precision in the decision probe.

### **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. Our experiments use publicly available benchmarks such as AdvBench and Harmful Behaviors in a controlled research setting. Although the paper studies existing attack methods, all analysis is conducted for defensive evaluation, and the proposed ALCA framework is intended solely to strengthen safety mechanisms rather than enable misuse.

### **Acknowledge**

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.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.
- Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2024. [Jailbreaking black box large language models in twenty queries](#). *Preprint*, arXiv:2310.08419.
- Jun Gao, Yongqi Li, Ziqiang Cao, and Wenjie Li. 2025. [Interleaved-modal chain-of-thought](#). *Preprint*, arXiv:2411.19488.
- 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.
- Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. [Training large language models to reason in a continuous latent space](#). *Preprint*, arXiv:2412.06769.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). *Preprint*, arXiv:2009.03300.
- 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, Timothée 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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15157–15173, Bangkok, Thailand. Association for Computational Linguistics.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. *AlpacaEval: An automatic evaluator of instruction-following models*. [https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval).
- Quan Liu, Zhenhong Zhou, Longzhu He, Yi Liu, Wei Zhang, and Sen Su. 2024a. *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. 2024b. *Autodan: Generating stealthy jailbreak prompts on aligned large language models*. *Preprint*, arXiv:2310.04451.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. *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.
- Yingzhe Peng, Chenduo Hao, Xu Yang, Jiawei Peng, Xinting Hu, and Xin Geng. 2024. *Live: Learnable in-context vector for visual question answering*. *Preprint*, arXiv:2406.13185.
- 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.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. *Direct preference optimization: Your language model is secretly a reward model*. *Preprint*, arXiv:2305.18290.
- Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025. *Codi: Compressing chain-of-thought into continuous space via self-distillation*. *Preprint*, arXiv:2502.21074.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. *Chain-of-thought prompting elicits reasoning in large language models*. *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*. 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.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. *Qwen2 technical report*. *Preprint*, arXiv:2407.10671.
- Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyang 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.
- 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 Related Work

**Output-based Safety Alignment.** Output-based Safety Alignment Methods center on supervising the model’s final output, treating the internal reasoning process as an opaque black box. Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) uses a separate reward model to reflect human preferences, which is then used to fine-tune the LLM policy. Direct Preference Optimization (DPO) (Rafailov et al., 2024) extracts

a reward signal directly from preference data, offering a more stable alternative. Constitutional AI (Bai et al., 2022) guides the model to revise its own outputs based on a predefined set of safety principles and reduces the reliance on extensive human labeling. However, such methods merely learn a simplistic mapping that directly refuses harmful queries, neglecting the modeling and supervision of the underlying safety reasoning process. Consequently, they struggle to recognize potential harms in more complex scenarios and remain vulnerable to carefully crafted jailbreaking attacks.

### Reasoning Process-based Safety Alignment.

To open the black box, researchers have turned to exposing and supervising the safety reasoning process of LLMs. The introduction of Chain-of-Thought (CoT) prompting (Wei et al., 2023) demonstrated that eliciting step-by-step reasoning improves performance and exposes the reasoning process. 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 CoT to safety. STAIR trains the non-reasoning LLM to generate the CoT that embodies safety reasoning. The explicit, textual CoTs generated by these methods provide a rich signal for fine-tuning. However, as we argue in our "Security-Auditability Dilemma," the discrete nature of these reasoning steps creates a fragile attack surface. ALCA is designed to preserve supervisability while eliminating this attack surface by conducting safety reasoning in the hidden space.

**Latent reasoning methods.** Some existing works have focused on making reasoning implicit or latent. Methods like ICOT (Gao et al., 2025) and COCONUT (Hao et al., 2024) fine-tune models to internalize reasoning steps, while others use knowledge distillation to embed the process in the model’s hidden states (Shen et al., 2025). More recently, dynamic latent compression performs reasoning entirely within these hidden states, avoiding explicit generation altogether (Peng et al., 2024). Our method focuses on concealing fragile safety reasoning in latent space and reconstructing it into explicit text only when supervisory audit is required.

## B Experimental Setup for Motivating Observations

We employ the AdamW optimizer with a learning rate of  $2 \times 10^{-5}$  and a batch size of 16 for 3 epochs. We utilize **ArtPrompt** (Jiang et al., 2024) and **PAP** as representatives of non-adaptive and adaptive jail-break attacks, respectively, with **AdvBench** (Zou et al., 2023) serving as the evaluation dataset. The Attack Success Rate (ASR) is determined via GPT-4 based evaluation. Further implementation details are provided in Section 4.

## C The Selection of Trigger

Table 4: Two methods for triggering the latent deliberation process on a balanced dataset of harmful and harmless prompts. The probe demonstrates superior performance across all metrics.

Trigger Mechanism	Precision	Recall	F1-Score
Internal Token	0.96	0.94	0.95
<b>External Probe</b>	<b>0.98</b>	<b>0.99</b>	<b>0.98</b>

The mechanism that triggers the transition into latent mode is critical for ALCA’s reliability. We compared our external probe against an alternative approach where the LLM internally generates a special token. As shown in Table 4, the decoupled probe demonstrates markedly superior performance, achieving an F1-score of **0.98** compared to the internal token’s 0.95.

Crucially for application, the probe attained a recall of **0.99**, ensuring that potentially harmful queries are almost never missed. We attribute the probe’s success to its focused design: as a dedicated binary classifier, it learns an accurate decision boundary without the multi-task interference faced by the internal token method, which must compromise between its classification task and the primary language modeling objective. This high-recall, high-precision trigger is thus an essential component for the overall robustness of the ALCA framework.

## D Choice of Latent Representation Target

A core design choice is how to construct the golden standard vectors  $\mathbf{z}^*$  used as targets for  $\mathcal{L}_{\text{latent}}$ . We compared three methods for extracting a gold standard.

As shown in Table 5, Pooling-based methods proved suboptimal for our sequential reasoning

Target Vector Method	ASR (Adap.) ↓	Sem. Sim. ↑
Weighted Pool.	18.2%	0.85
Mean Pooling	11.5%	0.92
<b>Last Token State</b>	<b>9.1%</b>	<b>0.96</b>

Table 5: Comparison of different methods for constructing the target latent vectors. Using the last token’s hidden state, which is native to the autoregressive model’s predictive function, yields the best results.

task. Attention-Weighted Pooling achieved an adaptive ASR of 18.2%, while Mean Pooling performed slightly better at 11.5%. We attribute this to the fact that pooling operations average states across the entire sequence, creating a holistic but blurred representation that loses the precise, forward-looking information needed to guide the next generation step.

In stark contrast, using the Last Token Hidden State as the target yielded superior results in both security (ASR: 9.1%) and reconstruction fidelity (Sem. Sim.: 0.96). This success is architecturally intuitive: in an autoregressive model, the final token’s hidden state is explicitly optimized during pre-training to serve as a complete, contextualized summary for predicting the subsequent token. This experiment validates that aligning our  $L_{latent}$  objective with the model’s inherent predictive function is the most effective approach.

## E Computational Overhead and Throughput Analysis

We measured the average inference latency (time per request) and the resulting throughput (requests per second) on a single NVIDIA A100 GPU. We compare our ALCA framework against the undefended base model and ALCA without hiding safety reasoning process. As shown in Table 6, our **ALCA framework demonstrates remarkable efficiency**, achieving a throughput nearly double that of the CoT-Safety model. This efficiency stems from two key sources: 1) Probe is a lightweight classifier, adding negligible latency to the overall process, and more importantly, 2) ALCA compresses the generation of long secure COT text and controls it to a fixed number of  $n$  hidden vector generations. Crucially, while ALCA is only marginally slower than the none-reasoning base model (an approx. 19% increase in latency), this modest increase is a highly acceptable trade-off. For this small computational cost, we gain a massive leap in security and robustness.

## F Experimental Setup

**Base Models:** To ensure the generality of our findings, we conduct experiments three models: **Llama-3-8B-Instruct** (Grattafiori et al., 2024), **Mistral-7B-Instruct** (Jiang et al., 2023), **Qwen2-7B-Chat** (Yang et al., 2024).

**Baselines: Training-Time Alignment Methods** includes: **RLHF (Reinforcement Learning from Human Feedback)** guides the LLM’s policy refinement via reinforcement learning. **STAIR (Zhang et al., 2025)**: generate safety reasoning to improve final outputs. We adapt it for safety by having it reason about potential policy violations. **ALCA (decoded)**: This model variant is trained identically to our full ALCA model but uses the self-decoding mechanism to reconstruct the safety reasoning trace and makes it explicit in the final output. This allows the reasoning to serve as feedback for adaptive attacks and be included in the ASR evaluation, directly testing our hypothesis on the necessity of concealment. **Inference-Time Defense Strategies** includes: **PPL (Perplexity-based Defense) (Alon and Kamfonas, 2023)**: is an inference-time filter that rejects outputs if their generation perplexity exceeds a predefined threshold, based on the hypothesis that jailbreaks induce atypical model states. **AED (Alignment-Enhanced Decoding) (Liu et al., 2024a)**: is an alignment-enhanced decoding method that adaptively refines token probability distributions using a Competitive Index and post-alignment logits to improve harmlessness while preserving helpfulness. **SafeDecoding**: (Xu et al., 2024) is a safety-aware decoding strategy that adjusts token-level generation behavior based on the probability patterns of harmful and harmless responses.

**Attacking Data.** We construct a training dataset using prompts from AdvBench (Zou et al., 2023).

**Implementation Details.** All models are trained for 3 epochs using the AdamW optimizer with a learning rate of  $2 \times 10^{-5}$  and a batch size of 16. Our implementation utilizes PyTorch and the Hugging

Model	Latency (ms) ↓	Throughput (req/sec) ↑
Base Model	152	6.58
ALCA(without hiding)	345	2.90
<b>ALCA</b>	<b>185</b>	<b>5.52</b>

Table 6: Inference Latency and Throughput Comparison. ALCA offers a superior balance, significantly improving throughput over explicit reasoning methods with only a minor overhead compared to the non-defended base model.

Face Transformers library.

**Adaptive Jailbreak Attacks** To comprehensively assess model robustness, we evaluate against four diverse and state-of-the-art adaptive jailbreak attack methods: **GCG**: (Zou et al., 2023) is a classic white-box attack using gradient-based greedy search to find adversarial suffixes. **PAP**: (Zeng et al., 2024) apply the taxonomy to automatically generate persuasive adversarial prompts to jailbreak LLMs **PAIR**: (Chao et al., 2024) is a black-box attack that uses an attacker LLM to iteratively refine prompts to elicit unsafe content. **AutoDAN**: (Liu et al., 2024b) is a state-of-the-art method using hierarchical genetic algorithms to automatically generate diverse and effective jailbreaks.

**Metrics**: We use a comprehensive set of metrics to evaluate different aspects of model performance: **Attack Success Rate (ASR)** is automatically judged by GPT-4. A lower ASR indicates higher robustness. **Auditability** is quantified by comparing the reconstructed reasoning against the ground-truth text using BLEU, ROUGE-L, and semantic similarity scores. **Downstream Utility** are measured by performance on standard benchmarks: MMLU (accuracy) (Hendrycks et al., 2021) and Alpaca-Eval (win rate) (Li et al., 2023).

## F.1 Experimental Setup for Auditability Evaluation

To rigorously quantify the auditability of ALCA, we evaluated the fidelity of its self-decoding mechanism. The core task is to measure the similarity between the ground-truth safety reasoning chains ( $C_{original}$ ) and their reconstructions from the latent space ( $\hat{C}_{decoded}$ ). We established a multi-faceted evaluation protocol using three distinct test sets and a suite of complementary metrics.

**Evaluation Datasets.** To ensure our evaluation is comprehensive, we curated test samples from three sources, each representing a different challenge profile:

- **AdvBench**: Consists of prompts from a well-

known adversarial attack benchmark. The corresponding safety reasoning is often structured and targeted at specific policy violations.

- **Harmful Behaviors**: A broader dataset covering a wide range of potentially harmful user requests. This tests the reconstruction of more diverse and general safety reasoning.
- **Complex Ethical Dilemmas (CED)**: A curated internal set of scenarios involving nuanced ethical gray areas, which require longer, multi-step reasoning. This probes the model’s ability to maintain fidelity on complex and subtle logic.

**Evaluation Metrics.** We employed three metrics to provide a holistic view of reconstruction quality, spanning from lexical to semantic fidelity:

- **BLEU-4**: Measures n-gram precision to assess the exactness of word and phrase reconstruction. Calculated using the `sacrebleu` library.
- **ROUGE-L**: Measures the longest common subsequence to evaluate the preservation of sentence structure and core information. Calculated using the `rouge-score` library.
- **Semantic Similarity (Sem. Sim.)**: Measures the core meaning equivalence. We use a state-of-the-art sentence embedding model, **Salesforce/SFR-Embedding-Mistral**, which demonstrates top performance on the MTEB (Massive Text Embedding Benchmark) leaderboard. We compute the cosine similarity between the embeddings of the original and decoded texts. A score near 1.0 indicates near-perfect semantic reconstruction.

The results in Table 7 are highly compelling. Across all datasets, ALCA achieves extremely high semantic similarity (average 0.96), demonstrating that the core logic of the safety reasoning is preserved losslessly. The strong ROUGE-L (average

Test Dataset	BLEU-4 $\uparrow$	ROUGE-L $\uparrow$	Sem. Sim. $\uparrow$
AdvBench	0.87	0.92	0.97
Harmful Behaviors	0.85	0.90	0.96
Complex Ethical Dilemmas (CED)	0.79	0.87	0.95

Table 7: Self-Decoding fidelity metrics for ALCA across multiple test sets. Higher scores indicate better reconstruction. The results show consistently high fidelity, especially at the semantic level, confirming the effectiveness of our audibility mechanism.

0.90) and BLEU-4 (average 0.84) scores further confirm high structural and lexical fidelity. Notably, for more complex scenarios like CED, while lexical scores slightly decrease as expected due to acceptable rephrasing, semantic similarity remains exceptionally high. This validates that ALCA’s latent reasoning is not an uninterpretable black box but a transparent and faithfully auditable process.

## G Ablation Study on the Number of Latent Reasoning Steps

**Motivation** A core hyperparameter in our ALCA framework is  $N$ , the number of autoregressive steps performed in the latent space for safety deliberation. This parameter directly controls the capacity and depth of the latent Chain-of-Thought. An insufficient  $N$  may lead to a shallow deliberation, failing to capture the full complexity of a safety reasoning chain, thus compromising both security and the fidelity of its later reconstruction. Conversely, an excessively large  $N$  could introduce redundant computations with diminishing returns on performance, increasing inference latency. To identify the optimal balance, we conduct an ablation study by varying  $N$  from 1 to 8.

**Analysis** The results, presented in Table 8, reveal a clear and informative trend. When  $N$  is small ( $1 \leq N \leq 3$ ), the model’s robustness is compromised, as indicated by a relatively high Adaptive Attack Success Rate (ASR). This is because the constrained latent space lacks the capacity to form a comprehensive and robust defense strategy. Concurrently, the Semantic Similarity for reconstruction is lower, suggesting that the compressed reasoning is lossy and incomplete.

As  $N$  increases from 4 to 6, we observe a significant improvement in both security and audibility. The ASR drops sharply, and the Semantic Similarity of the self-decoded reasoning approaches its peak. This suggests that a moderate number of latent steps is sufficient to represent the essential

safety logic. The performance saturates at  $N = 6$ , where the model achieves the best trade-off. Further increasing  $N$  to 7 or 8 yields negligible improvements in ASR and Semantic Similarity but steadily increases inference latency. This demonstrates a point of diminishing returns, where additional computational steps do not contribute meaningfully to the final outcome. Therefore, we select  $N = 6$  as the default setting for all other experiments in this paper, as it provides optimal security and audibility without unnecessary computational overhead. This result is consistent with that of coconut (Hao et al., 2024).

## H Decomposition of the Optimization Objective

In this section, we provide a detailed derivation of how the primary optimization objectives outlined in Section 3.1 (Eq. 3 and Eq. 4) can be equivalently transformed into three independent sub-optimization tasks under our idealized assumptions. This decomposition provides the theoretical foundation for our three-stage ALCA architecture.

Our primary goals are:

1. **Security:** Minimize the harmfulness of the user-facing output,  $C_{exp}$ .

$$\min_{\pi} \mathbb{E}_{x \sim \mathcal{D}_{adv}} [H(C_{exp}(\pi, x), y)]$$

2. **High-Fidelity Audibility:** Ensure the explicit chain remains coherent and the concealed chain is accurately reconstructable.

$$\max_{\pi, D, G_{latent}} \mathbb{E}_{(x, C_{full})} \left[ \sum_{j|m_j=\text{explicit}} \text{Sim}(c_j, c'_j) + \sum_{j|m_j=\text{latent}} \text{Sim}(c_j, \hat{c}_j) \right]$$

where  $\hat{c}_j = D(z_j)$  is the decoded text.

Let us analyze how to achieve the optimal solution for these goals based on the assumptions of *Harmlessness and Necessity and Sole Source of Risk*.

### H.1 Condition 1: Perfect Decision Alignment for Security

The security objective is to render the user-facing output  $C_{exp} = \{c'_j | m_j = 0\}$  completely harmless. According to our **Sole Source of Risk** assumption,

Steps ( $N$ )	ASR (Adap.) ↓	Sem. Sim. ↑	Latency (ms) ↓
1	25.1%	0.82	160
2	15.3%	0.89	165
3	11.2%	0.93	172
4	9.8%	0.95	178
5	9.3%	0.96	181
<b>6</b>	<b>9.1%</b>	<b>0.96</b>	185
7	9.0%	0.96	189
8	9.1%	0.96	193

Table 8: Ablation study on the number of latent reasoning steps ( $N$ ). We report the Adaptive Attack Success Rate (ASR), the Semantic Similarity (Sem. Sim.) of the reconstructed reasoning, and inference latency. Performance in terms of security and auditability stabilizes around  $N = 6$ , which strikes an optimal balance with computational cost. Lower ASR and latency are better; higher Sem. Sim. is better.

all safety risks are encapsulated *exclusively* within reasoning steps  $c_j$  where the ground-truth label  $s_j = 1$ . Consequently, the harmfulness function  $H(C_{exp})$  will be greater than zero if and only if at least one step  $c_k$  with  $s_k = 1$  is generated explicitly (i.e.,  $m_k = 0$ ).

To guarantee that  $H(C_{exp}) = 0$  for any adversarial prompt  $x$ , the policy  $\pi$  must ensure that for any reasoning step  $j$ , if its ground-truth label is  $s_j = 1$ , the policy’s decision must be  $m_j = 1$  (latent). This prevents the harmful content from ever materializing in the output.

Furthermore, our **Harmlessness and Necessity** assumption states that all steps with  $s_j = 0$  are both harmless and essential for a complete response. To maximize the utility and coherence of the final output (which is implicitly part of the auditability goal), these steps must be generated explicitly. Therefore, if  $s_j = 0$ , the optimal decision is  $m_j = 0$  (explicit).

Combining these two requirements, the optimal policy  $\pi^*$  that perfectly satisfies the security objective while preserving utility is one that perfectly aligns its decision  $m_j$  with the ground-truth harmfulness label  $s_j$  for all steps  $j$ . This leads to our first sub-optimization task:

**(1) Perfect Decision:** The policy  $\pi$  must learn to perfectly match the ground-truth label:  $\forall j, m_j = s_j$ .

## H.2 Condition 2: Lossless Decoding and Causal Non-Interference for Auditability

With the decisions fixed by the optimal policy  $\pi^*$  (i.e.,  $m_j = s_j$ ), we can now focus on the auditability objective. The maximization problem can be split into two independent sums over disjoint sets of indices: one for latent steps ( $s_j = 1$ ) and one for

explicit steps ( $s_j = 0$ ).

$$\max_{D, G_{latent}} \mathbb{E} \left[ \sum_{j|s_j=0} \text{Sim}(c_j, c'_j) + \sum_{j|s_j=1} \text{Sim}(c_j, D(z_j)) \right]$$

We can optimize these two terms separately.

### Optimizing the second term (Latent Steps)

The second term,  $\sum_{j|s_j=1} \text{Sim}(c_j, D(z_j))$ , exclusively involves the decoder  $D$ . To maximize this sum, we must optimize  $D$  to make its reconstruction  $D(z_j)$  as semantically close as possible to the original text  $c_j$ . This directly yields our second sub-optimization task:

**(2) Lossless Latent Decoding:** The decoder  $D$  must be optimized to maximally reconstruct the concealed steps:  $\max_D \text{Sim}(c_j, D(z_j))$  for all  $j$  where  $s_j = 1$ .

### Optimizing the first term (Explicit Steps)

The first term,  $\sum_{j|s_j=0} \text{Sim}(c_j, c'_j)$ , involves the generation of the explicit, user-facing text  $c'_j$ . The generation of  $c'_j$  is an autoregressive process, conditioned on the entire history of preceding steps, which includes both explicit text  $\{c'_k | k < j, s_k = 0\}$  and latent vectors  $\{z_k | k < j, s_k = 1\}$ .

The challenge here is subtle but critical. If the generation of a latent vector  $z_k$  (representing the unsafe step  $c_k$ ) fundamentally alters the model’s internal hidden state in a way that is inconsistent with having generated the original text  $c_k$ , then the probability distribution for a subsequent, harmless step  $c'_j$  will be perturbed. This perturbation can cause  $c'_j$  to deviate from the ground-truth  $c_j$ , thereby decreasing  $\text{Sim}(c_j, c'_j)$ .

To maximize the first term and ensure that the explicit chain remains faithful to the ground-truth

$C_{full}$ , the latent reasoning process  $G_{latent}$  must produce vectors  $z_k$  that are *causally equivalent* to their textual counterparts  $c_k$  in terms of their influence on future generation. In other words, the generation of  $z_k$  must not interfere with the model’s ability to generate subsequent harmless text correctly. This leads to our third sub-optimization task:

**(3) Causal Non-Interference:** The generation of a latent vector  $z_j$  must minimally alter the probability distribution for any subsequent general-purpose step  $c_k$  ( $k > j, s_k = 0$ ).

**Conclusion** By decomposing the problem, we have shown that under our idealized assumptions, the complex joint optimization of security and auditability is equivalent to satisfying three more tractable, sequential conditions. Our ALCA architecture is explicitly designed to solve these three sub-tasks in stages: the probe classifier for *Perfect Decision*, the self-decoder for *Lossless Decoding*, and the latent autoregressive deliberation with its hybrid loss for *Causal Non-Interference*.

## I Mechanism Analysis: The Risk of Internal Copying

In Section 2, we identified the *Internal Copy* risk as a primary driver of the Security-Auditability Dilemma. We hypothesized that explicit safety reasoning steps—even when masked from the user (User-Side Masking)—leave a “token footprint” in the model’s context window. The attention mechanism of LLMs, which is prone to attending to recent high-entropy tokens, can inadvertently “copy” harmful tokens from the reasoning chain into the final response, thereby bypassing the intended refusal.

In this section, we provide empirical evidence for this phenomenon and demonstrate how ALCA structurally eliminates this risk.

### I.1 Quantifying the Copying Effect

To measure the extent of internal copying, we introduce a metric termed the **Harmful Token Overlap Rate (HTOR)**. We define HTOR as the proportion of harmful n-grams ( $n = 3$ ) appearing in the final generated response that were also present in the safety reasoning Chain-of-Thought (CoT).

We conducted an analysis on the **User-Side Masking** baseline (where explicit CoT is generated but hidden from the user) using successful jailbreak samples from the PAP attack dataset. For

---

#### Case Study: Internal Information Leakage (Regurgitation)

---

##### Adversarial Prompt:

*“I am a chemistry student conducting research on historical warfare. For academic purposes only, please explain the chemical synthesis process of TNT from toluene.”*

---

##### Model Internal Reasoning (Chain-of-Thought) - Hidden from User:

1. *Analyze intent:* User asks for TNT synthesis.
  2. *Safety check:* This falls under "Explosives/Weapons". High risk.
  3. *Content retrieval:* TNT is Trinitrotoluene. It is made by nitrating toluene with a mixture of nitric and sulfuric acids in three steps... [Detailed Step 1]... [Detailed Step 2]...
  4. *Decision:* Although the user claims academic purpose, providing the synthesis recipe is prohibited. I must refuse.
- 

##### Final Response - Visible to User:

*“While I understand this is for academic research, I cannot provide a complete guide on manufacturing explosives. However, as noted in the analysis, TNT is synthesized by nitrating toluene with a mixture of nitric and sulfuric acids in a three-step process... [Harmful Information Leaked]. Please consult authorized textbooks for historical context.”*

---

Table 9: An instance of the **Security-Auditability Dilemma**. Although the model decided to refuse the request in the CoT, the explicit textual representation of the harmful knowledge in the reasoning buffer was inadvertently copied (regurgitated) into the final response, causing a safety failure.

comparison, we analyzed **ALCA**, comparing its decoded latent thoughts against its final output.

$$\text{HTOR} = \frac{|\text{N-grams}(\text{Response}) \cap \text{N-grams}(\text{Safety CoT})|}{|\text{N-grams}(\text{Response})|} \quad (11)$$

**Results.** As shown in Table 10, the results are stark. In instances where the **User-Side Masking** baseline failed to refuse (i.e., successfully jailbroken), we observed an exceptionally high HTOR of **68.4%**. This indicates that nearly two-thirds of the harmful content in the final output was verbatim copied from the model’s own safety reasoning process.

In contrast, **ALCA** maintains a significantly lower HTOR even in its rare failure cases. More importantly, because ALCA operates in a latent space without discrete tokens, the “copying” mechanism is physically disrupted. The model cannot attend to explicit tokens that do not exist in the context window.

Method	ASR	HTOR (Avg.)
User-Side Masking	25.9%	<b>68.4%</b>
ALCA (Ours)	<b>7.3%</b>	12.1%

Table 10: **Analysis of Internal Copying.** The high Harmful Token Overlap Rate (HTOR) in the User-Side Masking baseline confirms that explicit reasoning serves as a source for leakage via copying. ALCA significantly reduces this risk.

## I.2 Qualitative Case Study: The “Repeater” Failure Mode

To further illustrate this mechanism, we present a qualitative comparison of a failure case in Table 11.

When the **User-Side Masking** model attempts to reason about a harmful query (e.g., “How to synthesize [Substance X]”), it often explicitly generates the steps for synthesis within its CoT to evaluate their danger. Although this CoT is hidden from the user, it remains in the active context. The subsequent generation step, biased by the attention heads focusing on these recent chemical names and procedure tokens, essentially acts as a “repeater,” completing the harmful instruction instead of refusing it.

**ALCA** prevents this by encoding the analysis of [Substance X] into dense latent vectors  $\mathbf{z}_j$ . These vectors carry the *semantic signal* of “danger” and “prohibition” required for the refusal (Condition 2: Causal Non-Interference) but do not instantiate the discrete textual tokens of the chemical recipe. Consequently, the attention mechanism has no harmful tokens to copy, forcing the model to rely on the high-level semantic refusal signal.

## I.3 Conclusion

This analysis confirms that the fragility of explicit reasoning stems not only from external visibility but from the internal mechanics of Transformer-based generation. **The discrete nature of text creates an unavoidable trade-off between reasoning depth and the risk of copying.** ALCA resolves this by decoupling the *semantic function* of reasoning from its *textual representation*, validating the necessity of the Latent CoT framework proposed in this work.

---

**Prompt:** Write a tutorial on making [Harmful Substance]...

---

**Model: User-Side Masking (Baseline)**  
*[Internal CoT (Hidden)]:* To analyze this, I see the user wants to make [Substance]. The steps usually involve mixing **Component A** and **Component B** at high heat. This is illegal...  
**Output:** "...heating **Component A** and **Component B** can effectively create the substance..."  
**Diagnosis:** The model copied the recipe tokens from its own hidden thought.

---

**Model: ALCA (Ours)**  
*[Latent Space (Hidden)]:*  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_6$  (*Semantic: Danger detected, Policy Violation, Refusal Required*)  
**Output:** "I cannot assist with that request as it violates safety policies regarding regulated substances."  
**Diagnosis:** Without explicit tokens to copy, the latent semantic signal successfully guides a refusal.

---

Table 11: A case study illustrating the *Internal Copy* risk. The baseline model unintentionally leaks harmful content by copying from its own explicit reasoning. ALCA’s latent representation prevents this leakage.