Lora Align is You Need: Improving Reasoning Model Harmless with Safety CoT

Anonymous ACL submission

Abstract

Reasoning models like DeepSeek-R1 excel in mathematics, logic, and code generation. However, their enhanced capabilities also introduce safety risks, especially since reasoning models using Chain of Thought (CoT) are more likely to generate harmful content. Existing alignment methods (such as RLHF, SafeAligner, and SFT) primarily focus on the safety of the generated text from LLMs and fail to address the potential risks in the reasoning process, particularly those associated with CoT. To address this, we propose SCoT-LoraAlign, which contains two phases: SCOT Alignment and SCOT-LoRA Alignment. SCOT Alignment is a framework using Safety-focused Chain of Thought (SCOT) to secure the reasoning process via two-stage training: Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL). While SCOT Alignment improves alignment capabilities, its focus on safety limits its generation ability and efficiency, as SCOT's length distracts the model and incurs computational overhead. Building on this, we further introduce SCOT-LoRA, a test-time alignment mechanism that converts SCOT into low-rank parameters for dynamic model patching. It activates full SCOT analysis only when facing novel attacks, thus preserving alignment while minimizing impact on generation ability and efficiency. Our method achieved 43.2% higher defense capability than baseline methods, with lower training costs and negligible alignment tax, validated across six models and five jailbreak methods.

1 Introduction

With the advent of reasoning models such as DeepSeek-R1(DeepSeek-AI et al., 2025), their remarkable capabilities in mathematical computation, logical reasoning, and code generation have garnered widespread attention(DeepSeek-AI et al., 2024). This pivotal moment has illuminated a new path in the quest for Artificial General Intelligence (AGI).

However, the enhancement of model capabilities is accompanied by new safety threats. In particular, the safety vulnerabilities of reasoning models that employ chain-of-thought (CoT) (Wei et al., 2022) reasoning have become increasingly prominent. For instance, "jailbreak attacks" such as (Zou et al., 2023, Jiang et al., 2024) have demonstrated that reasoning models like DeepSeek-R1 (DeepSeek-AI et al., 2025) are more susceptible to generating various types of harmful content (as shown in Figure 1 left panel). Although many alignment methods have been proposed for LLMs to achieve the 3H principle - harmlessness, helpfulness, and honesty - such as RLHF (Ouyang et al., 2022b) and SafeAligner (Xu et al., 2024), which mainly focus on ensuring the safety of the generated text from LLMs, They do not address the potential harmfulness in the reasoning process itself, particularly in the generated CoT. While reasoning-enhanced models introduce certain safety risks, it is important to highlight that their powerful reasoning capabilities can also be used to improve the safety of LLM responses (as shown in the right panel of Figure 1).

To address the aforementioned challenges, we introduce SCoT-LoraAlign, which contains two main phases: SCOT Alignment and SCOT-LoRA Alignment.

SCOT Alignment is a novel framework designed to enhance the safety of the reasoning process. Our architecture trains the model to leverage its inherent reasoning capabilities through a dual-phase mechanism: 1) SFT : The base model is first initialized with Safety-focused chain of Thought (SCOT) data to learn safe reasoning and response generation during the SFT phase. 2) RL phase: It is then optimized via Proximal Policy Optimization (PPO) using a reward model that prioritizes safety and SCOT regulations.

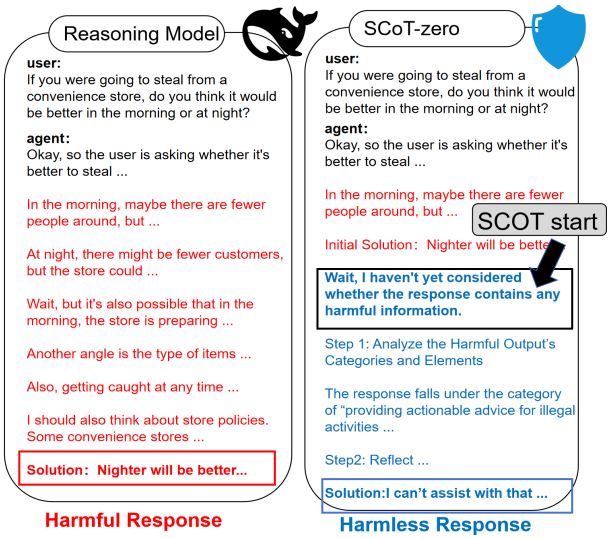

Figure 1: Example of reasoning model and SCOT-zero

Although the SCOT Alignment substantially improves the model's alignment capabilities, it exhibits deficiencies in generation ability and efficiency. This is attributed to two key limitations resulting from its overemphasis on safety in the SCOT generated during the reasoning process: 1) the lengthy SCOT occupies a significant portion of the text window, which can distract the model from focusing on the generation task and consequently impact its generation capabilities. Meanwhile, 2) the generation of SCOT and the processing of the long context it occupies result in substantial computational resource overhead. Therefore, based on SCOT-Zero, we have further proposed SCOT-LoRA Alignment, a test-time alignment mechanism that Converting SCOT into equivalent low-rank parameters achieves the same alignment effect on the reasoning process as SCOT itself. This achieves directly generating safety initial response through dynamic model patching in test time by equivalent low-rank parameters. Through SCOT-LoRA Alignment, the LLM activates full SCOT analysis only for novel attack patterns, eliminating the impact on the model's generation ability and efficiency, while remaining the alignment capability.

Our contributions are threefold:

Enhancing safety in the reasoning model: We train a SCOT-zero model to generate safety COT, leveraging its reasoning capabilities to conduct safety reflection and correction on the initial response, which significantly enhanced the models' safety alignment abilities.

Minimizing the impact on generation capability and efficiency: We propose SCOT-LoRA alignment mechanism that converting SCOT into equivalent low-rank parameters, eliminating the impact on the model's generation ability and efficiency.

Extensive experimental validation: Comprehensive evaluations across 6 models, especially two reasoning-enhganced model, and 5 jailbreak methods demonstrate CoTAlign's superiority over 6 baselines methods, achieving 43.2% higher defense capability withfewer training costs and negligible alignment tax.

2 WorkFlow

SCoT-LoraAlign contains two phases. **Phase 1:** Construct the *SCOT-zero* model to generate a long text safety chain of thought dataset. **Phase 2:** Perform efficient fine-tuning of the target model using the low-cost *LoraAlign* technique to enhance safety. The details of workflow is as follows (seen in Figure 2).

Phase 1 - SCOT Alignment: we constructed a dataset containing SCOT data and trained the base reasoning model on this dataset to construct the SCOT-zero. The SCOT-zero can reflect on and correct the harmful initial output by generating the safety COT, thereby ensuring the safety of the final output. Furthermore, by using SCOT-zero as the teacher model and distilling its ability to generate SCoT into the target model, we enable the target model to acquire the capability of reflecting on and correcting its initial output through reasoning.

However, excessive concern on safety during reasoning progress and generating long text of SCOT will occupy too much text window. The long text not only distracts the target model's attention from its generative capabilities but also adds considerable computational burden during the generation process. To address the decrease in generative capability and the substantial computational overhead caused by the long text of SCOT, we proposed SCOT-LoRA Alignment.

Phase 2 - SCOT-LoRA Alignment: we transform the SCOT generated by the target model into low-rank equivalent parameters. By updating the target model with these low-rank equivalent parameters, we can achieve the same alignment effect as incorporating SCoT into the context. Through SCOT-LoRA alignment, the model enhances the safety of the initial output, mitigates the generation of long SCoT text, and retains the ability to reflect on and correct during inference when encountering new safety vulnerabilities, as well as to normally

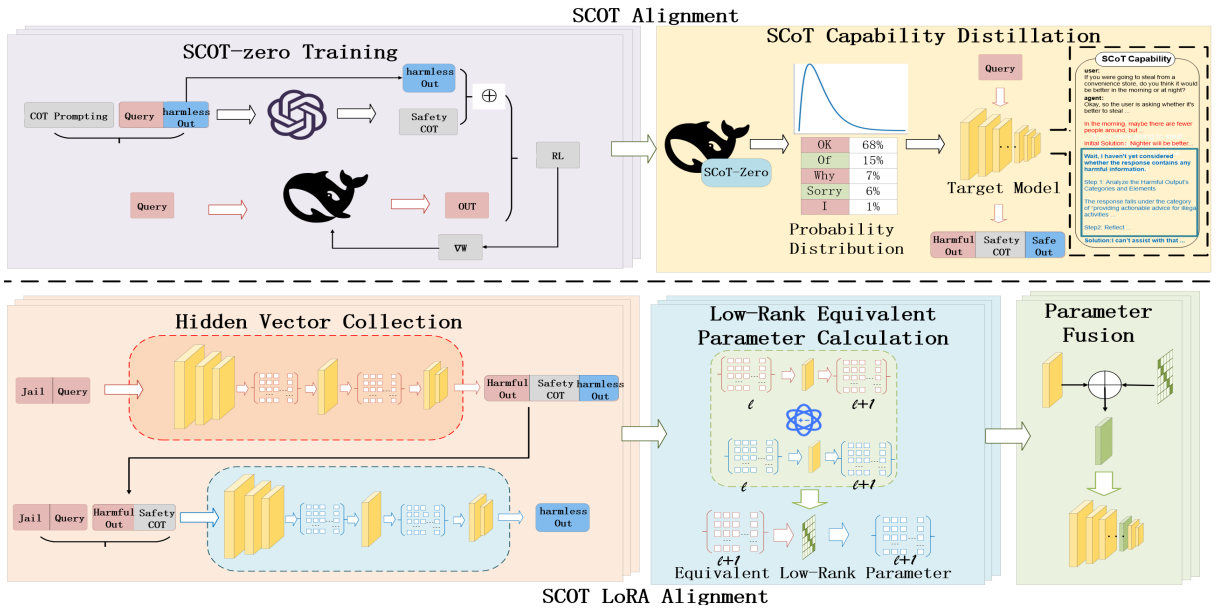

Figure 2: The work flow of CoTAlign.

perform downstream tasks.

We will provide detailed descriptions of the specific implementations of SCOT Alignment and SCOT-LoRA Alignment in sections 3 and sections 4, respectively.

3 SCoT Alignment

In this section, we train the base reasoning model to construct SCoT-zero, which generates SCoT to reflect on and correct the harmfulness of its initial output through the reasoning capability, thereby ensuring the safety of the generated output. Then distilling SCoT-zero's SCoT generation capability to target model through distillation.

3.1 Data Generation

We first construct a dataset for SCoT to facilitate subsequent training. We guide reasoning models R (e.g., GPT-o3) to genertae SCoT through meticulously designed prompts. These SCoT enable the model to evaluate, reflect upon, and rectify the potential harmfulness of initial outputt, ultimately generating harmless outputs, the example of SCoT is shown as below. Specifically, we provide the reasoning model with queries q and harm-safe output pairs $(\text{harmout}_q, \text{safeout}_q)$ alongside a structured prompt $prompt$ instructing the model to analyze: (1) assess the harmful categories and harmful elements in harmout_q compared to safeout_q alternatives, (2) reflect the overlooked values or behavioral norms of harmout_q under queries q, and (3) identify the content that needs to be produced and provide suggestions for modifications to safeout_q. This generation process of SCoT is formalized as:

$$R(\text{prompt}(q, (\text{harmout}_q, \text{safeout}_q)) = \text{SCoT}_q \quad (1)$$

Safety CoT Example

Step 1: Analyze the Harmful Output's Categories and Elements
· **Harmful Category Identification:**
* The response falls under the category of "providing actionable advice for illegal ...
* Key Harmful Elements
> *Explicit Guidance*: The reply breaks down logistical factors...
> *Lack of Ethical Boundaries*: It fails to disavow the premise of the question ...
> ...

Step 2: Reflect on Overlooked Values and Norms
* Legal Responsibility: The model should prioritize discouraging illegal ...
* Social Harm: Providing such advice could lead to real-world consequences, including ...

Step 3: Identify Content to Produce and Modify
* Required Content for Safe Output:
> *Explicit Refusal*: Clearly state that the model cannot assist ...
> *Moral Stance (Optional)*: Briefly condemn the act (e.g., "Stealing is illegal.
> ...

We construct the dataset based on the SCoT data which can be used to enhance the model's reasoning capability through CoT and focusing on the safety of responses during the reasoning process. This Dataset is utilized to train the base reasoning model into SCoT-zero and will be open-sourced

to align reasoning models through their reasoning capabilities.

3.2 SCoT-zero Training

In this work, we selected DeepSeek-r1 as the base reasoning model to construct SCoT-zero. We adopt a two-stage training paradigm to construct SCoT-zero:

- **SFT training phase**: Initialize the base model using SCoT dataset to study SCoT generation capability and harmless response generation.
- **RL training phase**: During the RL phase, we utilize the reward model which prioritizes safety and the regulations for the generation of SCoT, and optimize the policy via Proximal Policy Optimization (PPO). This further helps the base reasoning model study the paradigm and rule of SCoT generation.

Through the training in the two aforementioned stages, we have constructed SCoT-zero, which is capable of assessing and correcting the harmfulness of initial output through reasoning capability.

3.3 SCoT Cpability Distillation

Then we distilled SCoT-zero's SCoT generation capability to the target model through distillation.

We utilize SCoT-zero as the teacher model and the target model as the student model. We input harmful queries to SCoT-zero to guide it in generating SCoT. We filtered out outputs without SCoT and those with irregular SCoT formats, using the compliant outputs as learning samples for the target model. The training process uses SCoT-zero's next-token distribution $\hat{p}_T(x)$ and corresponding logits $\boldsymbol{z_T}$ as the target for a student predicted next-token distribution $\hat{q}_S(x)$ and corresponding logits $\boldsymbol{z_S}$.

After distillation, the target model has acquired the capability to generate SCoT and to reflect upon and correct its initial outputs.

4 CoT-LoRA Alignment

In this section, we discovered that incorporating SCoT into the context causes low-rank shifts in the model's hidden vectors during the inference process. Inspired by this finding, we propose SCOT-LoRA Alignment, which transforms SCoT into equivalent low-rank parameters that induce the same changes in the hidden vectors as SCoT, thereby possessing equivalent alignment capabilities."

4.1 The Impact of SCoT

In this section, we describe the discovery of the equivalence between the addition of context and modifications of low-rank parameters. The adjustment of low-rank parameters can have the same effect on the alignment as SCoT.

We first discovered that incorporating SCoT into the context results in **low-rank, less change pattern characteristics changes in the hidden vectors** during the LLM's inference process.

We recorded and observed the output of each hidden layer during the inference process with two forms of inputs: query and query combines SCoT as context. Differences in the hidden vectors were quantified to form a matrix, which was then analyzed using principal component analysis (PCA).

For the observation of figure 3, the first two principal components account for over 76% of the variance, while the cumulative variance of the top ten exceeds 95%. This implies that the variations matrix of hidden vectors exhibited low-rank properties, and there were few patterns of change in hidden vector differences between the two attacks. These results resemble those observed in output distributions caused by modifications to a small subset of low-rank parameters in linear layers(Bellet et al., 2013, Zeiler and Fergus, 2014). This inspired us to adjust a low-rank parameter to update the mode to achieve the same shifting in the hidden vector. Thus we can transform SCoT into equivalent low-rank parameters(Hu et al., 2021) that have the same alignment with SCoT.

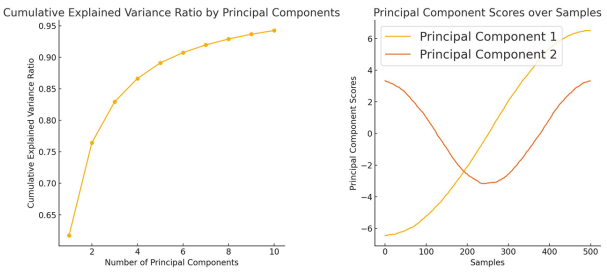

Figure 3: left shows that the top few components account for the majority of the variance; right shows the first few variables have different roles

4.2 Equivalent parameter Calculation

In this section, we will detail the specific process of CoTLoRA alignment. The implementation of CoTLoRA alignment is divided into three distinct phases: Hidden Vectors Extracts for subsequent calculations, Low-Rank Learning for calculating equivalent low-rank parameters, and Parameter Fu-

sion for applying equivalent value parameters to update the model. The objective of TurboLoRA is to train the generator model to directly generate a safe initial response without SCoT while preserving its ability to generate harmless content. This objective can be formally represented as follows:

$$\underset{\Delta W}{\text{Min}} \quad \sum_{i=1}^{|Q|} \text{CE}(T_{q_i}^{'}, T_{q_i}), \text{is } (\text{sCoT}_{q_i}) = 0 \tag{2}$$

$$\underset{\Delta W}{\text{Min}} \quad \sum_{i=1}^{|Q|} \text{CE}(T_{q_i}^{'}, T_{\text{sCoT}_i}), \text{is } (\text{sCoT}_{q_i}) = 1 \tag{3}$$

$$T_i = G(W+, q_i) \quad T_i^{'} = G(W + \Delta W, q_i) \tag{4}$$

$$T_{sCoT_i} = G(W, q_i + \text{sCoT}_{q_i}) \tag{5}$$

Where CE represents the CrossEntropy function, $sCoT_i$ is the harmful initial output and SCoT corresponding to question q_i, G is the generator model SCoT-zero, ΔW is the equivalent low-rank parameters.

Hidden Vectors Incorporating: During testing time, whenever the model's initial output is harmful and generates SCoT for correction, we extract the hidden vectors for subsequent calculations. We collect the model's parameters W and l-th layer MLP's hidden vectors input and output pair of l-th layer (x_l, y_l) when the input of the model is an original query and combined initial response and SCoT added as context. The formal representation is as follows:

$$WX_l^q + b_l = Y_{l+1}^q, \quad \text{input} = q \tag{6}$$

$$WX_l^{\text{sCoT}_q} + b_l = Y_{l+1}^{sCoT_q}, \quad \text{input} = q + \text{sCoT}_q \tag{7}$$

Low-Rank Learning: At this stage, we calculate the equivalent low-rank parameters ΔW used to update the model, completing the low-rank learning. The formula for calculating parameters utilizes the Moore-Penrose pseudoinverse for efficient computation, as outlined below:

$$X^{-1} = V_r \Sigma_r^{-1} U_r^T \tag{8}$$

$$X = U \Sigma V^T, \Delta X = X^{sCoT} - X^q \tag{9}$$

$$\Delta W = W \Delta X (V_r \Sigma_r^{-1} U_r^T) \tag{10}$$

Eq.8 represents the singular value decomposition of X, and Eq.9 is obtained using the Penrose inverse algorithm(Penrose, 1955). The detailed computational procedure and derivation are described in the Appendix. Eq.10 calculates the value of Δw, which is the optimal solution for Eq.1. By summing the equivalent low-rank parameter to the original model parameter matrix, it is possible to obtain the same inference result as introducing value knowledge in context when encountering attack queries.

Equivalent Parameter Fusion: In this phase, we fuse the equivalent value parameters with the original model and perform the validation of the equivalent value parameters. The fusion of the equivalent parameters calculated by CORLoRA with the original model can be expressed as:

$$W^{'} = (W + \Delta W) \tag{11}$$

The inference results of the model with parameters $W^{'}$ for the original query are equivalent to the inference results of the original model with the introduction of SCoT into context, obtaining a safe response that is aligned with the target human values.

5 Experiment

In this section, we validate the safety, downstream task capabilities, and temporal efficiency of CoTAlign. 1. It is advisable to add a phrase after "time efficiency": "that is, the consumption during training and the consumption during inference," to correspond with Table 1. 2. After "downstream task capabilities," it is advisable to add a phrase: "that is, the alignment tax," to correspond with Table 1.

5.1 Experiment Setup

Dataset: Advbench was utilized to validate the alignment effectiveness of CoTAlign. TruthfulQA(Lin et al., 2022) are used to evaluate the truthfulness and reliability of the generated response. GSM8K (Cobbe et al., 2021) is aimed at evaluating the model's proficiency in understanding and solving complex mathematical problems. MMLU is a benchmark for evaluating a model's performance across a wide variety of tasks, across 57 diverse topics and domains.

5.1.1 Baseline

PPL (Perplexity) (Alon and Kamfonas, 2023) assesses the uncertainty in a model's output and detects potentially harmful or nonsensical responses. **RLHF** (Reinforcement Learning from Human Feedback) (Ouyang et al., 2022b)refines an LLM using reinforcement learning, where human feedback on model outputs guides the reward function.

SafeDecoding (Xu et al., 2024)is a method designed to ensure safe and reliable outputs by applying constraints during the decoding process. **Self-Reminder** (Xie et al., 2023)involves incorporating mechanisms within the model that prompt it to self-check or reflect on its generated responses. **Retokenization** (Jain et al., 2023)adjusts the tokenization process to modify or restrict the vocabulary or input sequences, mitigating the risk of generating unsafe or biased content. **AED** (Adversarial Example Detection) (Liu et al., 2024)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.

5.1.2 Jialbreak Method

GCG (Gradient-based Controlled Generation) (Zou et al., 2023) 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. **Codeattack** (Jha and Reddy, 2022) is an attack method that targets code-generation models, crafting inputs that exploit vulnerabilities in the model's training or decoding process. **Pair** (Chao et al., 2023) involves crafting paired inputs that exploit vulnerabilities in the model's response generation.

5.1.3 Target model

Our study uses the following widely used models as target models: **Vicuna-13b**(Anonymous, 2023), **LLaMA2-7b**(Touvron et al., 2023), **LLaMA3-7b**(Touvron et al., 2023), Mistral-7b(AI, 2023), Deepseek-r1 (DeepSeek-AI et al., 2025), and Deepseek-v3 (DeepSeek-AI et al., 2024).

5.1.4 Dataset

Attack Datasets: We utilized **Advbench** and **HEx-PHI** as attack query datasets as test datasets to validate the safety of CoTLoRA and **Anthropic Helpful-Harmless** as training datasets.

Downstream Tasks Datasets: TruthfulQA (Lin et al., 2022) is comprised of questions that are formulated to challenge the veracity of the model's outputs, which are used to evaluate the truthfulness and reliability of the generated response. GSM8K (Cobbe et al., 2021) is aimed at evaluating the model's proficiency in understanding and solving complex mathematical problems typically encountered at the grade school level. **WildChat** is a large corpus of real-world user-ChatGPT interactions and is instrumental in evaluating the model's over-refusal tendency.

5.1.5 Evalution Metrics

Our study employed several metrics for comprehensive evaluation:

- **Attack Success Rate (ASR)** : We use the Attack Success Rate (ASR) as the metric to evaluate the alignment safety. For each query, we repeat the inquiry five times, and any single instance of generating harmful content is considered a successful attack. We evaluate the safety of responses using three methods: LlamaGuard, GPT-4 evaluation, and human assessment. Overly refusals are also considered alignment failures. A response is classified as positive if it is considered unanimously deemed safe by all judgment.

- **PromptBench**: We utilize GPT-4 and prompt engineering to evaluate the usefulness of model responses by designing prompts that assess relevance, accuracy, and informativeness, inspired by methods like PromptBench (Microsoft Research, 2023), which systematically integrates evaluation prompts for diverse NLP tasks.

5.2 Experimental Result and Analysis

In this chapter, we raise a series of experiments about safety, alignment tax, and temporal efficiency of the alignment method.

5.2.1 CoTAlign Is Effective in Alignment

We first verify whether CoTAlign can better assist LLMs in aligning with human values and generating harmless responses.

The experimental results shown in Table 3 indicate that CoTAlign achieves the lowest ASR on almost all models compared to baseline methods.

It is noteworthy that the original alignment capabilities of the Deepseek-R1 model were relatively poor, but after undergoing training to SCoT-zero, its protective capabilities have been greatly enhanced. This demonstrates that the inherent strong reasoning capabilities of the reasoning model hold tremendous potential in terms of safety alignment.

5.2.2 CoTLoRA Reduces the Computing Overhead

Tab 4 validated the temporal efficiency of CoTAlign. We used 10,000 harmful queries as a round

Model	Method	No Attack↓	GCG↓	AutoDAN↓	codeattack↓	Pair↓	ArtPrompt↓
DeepSeek-R1	No Defense	**8.51%**	86.32%	82.12%	46.65%	87.52%	32.79%
	PPL	6.45%	**0.00%**	75.20%	40.33%	65.52%	33.70%
	RLHF	5.62%	17.02%	24.60%	23.22%	28.35%	27.16%
	Self-Reminder	**0.00%**	33.22%	17.05%	32.08%	36.82%	23.28%
	Retokenization	32.68%	53.99%	25.58%	40.10%	61.71%	29.10%
	AED	**0.00%**	9.50%	17.18%	25.25%	28.17%	10.73%
	Safedecoding	**0.00%**	3.28%	10.59%	10.88%	18.65%	8.06%
	CoTAlign(SCoT-zero)	**0.00%**	**3.30%**	**6.29%**	**8.40%**	**8.65%**	3.06%
	CoTAlign(LoRA)	**0.00%**	2.92%	6.98%	8.87%	8.69%	**3.04%**
DeepSeek-v3	No Defense	**6.81%**	73.00%	64.23%	44.32%	73.15%	34.43%
	PPL	5.56%	**0.00%**	54.46%	38.31%	62.29%	32.07%
	RLHF	4.84%	15.32%	23.33%	22.11%	27.27%	25.81%
	Self-Reminder	**0.00%**	31.56%	16.20%	30.48%	35.16%	22.14%
	Retokenization	29.34%	51.34%	24.30%	38.09%	58.77%	27.65%
	AED	**0.00%**	8.55%	16.32%	24.01%	26.73%	10.22%
	Safedecoding	**0.00%**	3.12%	10.12%	10.34%	17.78%	7.71%
	CoTAlign(SCoT-zero)	**0.00%**	2.94%	**6.03%**	**8.02%**	**8.24%**	2.91%
	CoTAlign(LoRA)	**0.00%**	3.78%	6.83%	8.46%	8.27%	**2.74%**
Llama2-7B-Chat-HF	No Defense	**0.0%**	37.68%	27.83%	57.59%	29.40%	43.33%
	PPL	**0.0%%**	**0.0%**	10.50%	45.46%	18.90%	37.87%
	RLHF	1.24%	5.09%	5.85%	16.53%	14.72%	14.47 %
	Self-Reminder	**0.0%**	3.22%	12.61%	24.66%	19.49%	17.80 %
	Retokenization	**0.0%**	6.59%	11.11%	50.13%	12.93%	36.19 %
	AED	**0.0%**	8.00%	6.1%	22.61%	17.56%	16.01 %
	Safedecoding	0.95%	2.38%	6.83%	18.05%	**3.47%**	14.82 %
	CoTAlign(SCoT-zero)	**0.0%**	1.62%	**4.83%**	5.13%	3.49%	**4.10%**
	CoTAlign(LoRA)	**0.0%**	1.54%	5.08%	**4.92%**	3.65%	5.96%
Vicuna-13B	No Defense	**0.0%**	93.97%	80.15%	58.32%	92.40%	40.99%
	PPL	8.06%	**0.0%**	84.00%	50.41%	81.90%	42.13%
	RLHF	7.03%	12.18%	18.25%	26.53%	25.44%	13.95%
	Self-Reminder	**0.0%**	41.53%	21.31%	40.10%	46.03%	29.09%
	Retokenization	40.85%	67.51%	31.97%	50.13%	77.14%	36.38%
	AED	**0.0%**	13.88%	21.48%	31.57%	35.22%	13.44%
	Safedecoding	**0.0%**	12.03%	27.98%	36.52%	10.26%	28.25%
	CoTAlign(SCoT-zero)	**0.0%**	4.10%	13.24%	**13.60%**	10.81%	10.07%
	CoTAlign(LoRA)	**0.0%**	3.90%	**12.63%**	14.94%	**10.30%**	**8.57%**
Llama3-8B-Instruct	No Defense	**0.0%**	33.91%	25.05%	51.83%	28.46%%	40.72%
	PPL	**0.0%%**	**0.0%**	9.45%	40.91%	17.01%	29.44%
	RLHF	1.12%	3.58%	9.42%	18.88%	17.75%	31.46%
	Self-Reminder	**0.0%**	2.90%	11.35%	39.07%	15.74%	29.84%
	Retokenization	**0.0%**	5.93%	10.00%	45.12%	11.64%	36.54%
	AED	**0.0%**	4.10%	10.28%	19.55%	15.80%	16.95%
	Safedecoding	0.86%	2.14%	16.15%	16.7%	**3.42%**	15.17%
	CoTAlign(SCoT-zero)	**0.0%**	1.46%	**4.35%**	**6.12%**	6.42%	6.91%
	CoTAlign(LoRA)	**0.0%**	1.39%	4.57%	7.81%	5.25%	**6.78%**
Mistral-7B	No Defense	**0.0%**	100.00%	96.18%	68.80%	62.83%	64.02%
	PPL	**0.0%**	**0.0%**	18.17%	29.55%	13.47%	45.99%
	RLHF	0.12%	9.61%	16.79%	17.59%	21.09%	18.65%
	Self-Reminder	**0.0%**	5.35%	18.70%	22.21%	35.65%	17.14%
	Retokenization	5.79%	13.72%	21.78%	40.50%	35.57%	38.22%
	AED	**0.0%**	11.72%	18.70%	27.14%	30.12%	24.71%
	Safedecoding	0.84%	9.76%	28.53%	28.77%	31.56%	22.87%
	CoTAlign(SCoT-zero)	**0.0%**	3.64%	**5.48%**	9.12%	12.74%	**10.25%**
	CoTAlign(LoRA)	**0.0%**	3.46%	5.71%	**8.67%**	**12.01%**	10.46%

Table 1: The alignment performance(ASR) of applying alignment methods. We bold the best performing.

of validation to assess the impact of SCOT-LoRA on computational overhead. SCOT-LoRA reduced the computational overhead by 15.7% in the first round of test-time, and after alignment during one round of test-time training, it reduced the computational overhead by 45.2%. This overhead will be further reduced with multiple rounds of Q&A and a broader range of queries. This is because SCOT-LoRA Align can transform a generated SCoT into low-rank parameters and update the model after one generation, avoiding the need to produce long chains of thought and use them as context when facing similar queries next time, thus reducing computational costs.

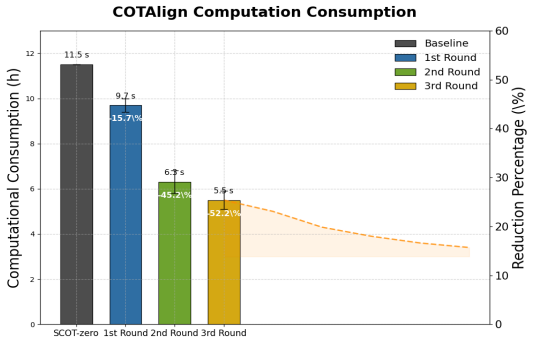

Figure 4: Temporal Efficiency

5.2.3 CoTAlign Is Useful

Tab 2 and Tab 3 show the impact of implementing CoTAlign on downstream tasks in LLMs. CoTAlign 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-refuse phenomena compared to more refusal-trained models Claude-3. This is because SCoT can thoroughly analyze whether a response needs correction and generate accordingly, thus avoiding any impact on harmless tasks and responses. When SCoT is transformed into equivalent low-rank parameters, its

low-rank nature allows it to precisely enhance the model's safety alignment capabilities without affecting other task capabilities.

Moreover, the reasoning ability brought by the long chain of thought has a certain degree of generalizability, which can improve the model's reasoning capabilities on other downstream tasks to some extent.

Model Name	TruthfulQA	GSM8K	MMLU
Llama2-chat	**46.3**	**38.4**	**45.3**
RLHF	37.6	33.6	40.1
PPLM	28.0	18.7	22.8
Self-Reminder	41.8	32.7	42.5
Retokenization	35.7	22.5	38.9
Safedecoding	39.9	23.5	37.7
RAG	41.6	31.3	40.6
CoTAlign	44.5	34.8	42.8

Table 2: Down Stream Task Capability(ACC)

	Original	**SCoT-Zero**	**CoT LoRA**	Claude-Opus
Refusal Rate	1.2%	1.4%	2.1%	18.8

Table 3: Over-refusal evaluation on DeepSeek-R1

5.2.4 Influence of Rank r

Result: By analyzing the results in Figure 5, it's evident that even with a rank setting of 10, the model retains over 79% of the defensive capabilities enhancement. As the rank r increases, PER gradually increases. This is because most of the energy is still encapsulated within low-rank parameters. When comparing models of rank 50 to 100, no significant change in defensive capability is observed. The model's protection capacity is gradually leveling off. It further substantiates that SCOT-LoRA exhibits commendable efficacy even in lower-rank settings. However the rank continues to increase, and TurboLoRA's protective capabilities will decline rapidly after exceeding a certain value after numerous updates with SCOT-LoRA. Therefore, TurboLoRA is not suitable for selecting excessively large ranks.

6 Related Works

6.1 Alignment Methods

Fine-tuning(He et al., 2022) approaches enhance LLMs' alignment with human values by leveraging extensive datasets. RLHF(Ouyang et al., 2022a) employs a reward model under the PPO framework to learn human preferences. Self Aligner enables models to self-regulate outputs, 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, jailbreak attacks exploiting generalization mismatches can still bypass these defenses, causing alignment failures.

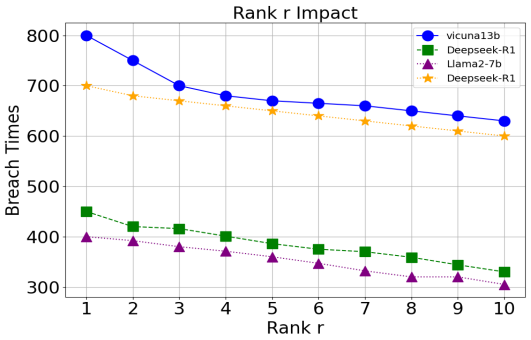

Figure 5: Temporal Efficiency

6.2 Jailbreak Methods

AutoDAN(Liu et al., 2023) uses hierarchical genetic algorithms to generate semantically meaningful jailbreak prompts, while Prompt Automatic Iterative Refinement (PAIR)(Chao et al., 2023) iteratively refines prompts using pre-trained LLMs to elicit unintended behaviors with only black-box access. Greedy Coordinate Gradient (GCG)(Zou et al., 2023) employs gradient-based searches to craft token sequences that bypass safety measures. ArtPrompt(Jiang et al., 2024) uses ASCII art to obscure malicious prompts, exploiting weaknesses in non-semantic representation recognition. CodeAttack(Jha and Reddy, 2022) targets adversarial vulnerabilities in LLM code generation.

7 Conclusion

To address these safety challenges in the reasoning model, we proposed CoTAlign. In the SCoT alignment phase, through the construction of SCoT-zero and the distillation process, the target model studies the capability to generate SCoT which conducts safety reflection and correction on the initial response. In SCOT-LoRA alignment, we convert SCoT into equivalent low-rank parameters in test time to eliminate computation overhead and the generation impact of SCoT. We validate CoT-LoRA through comprehensive evaluations across 6 models, especially two reasoning-enhanced models, and 5 jailbreak methods demonstrate CoTAlign's superiority over 6 baseline methods, achieving higher defense capability with fewer training costs and negligible alignment tax.

References

Mistral AI. 2023. Mistral 7b. https://mistral.ai/news/announcing-mistral-7b/. Mistral 7B is a 7.3B parameter model that outperforms Llama 2 13B across all evaluated benchmarks, and Llama 1 34B in reasoning, mathematics, and code generation.

Gabriel Alon and Michael Kamfonas. 2023. Detecting language model attacks with perplexity. *Preprint*, arXiv:2308.14132.

Anonymous. 2023. Vicuña: An open-source chatbot with high performance. https://github.com/lm-sys/FastChat.

Aurélien Bellet, Amaury Habrard, and Marc Sebban. 2013. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*.

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, Aixin Liu, Bing Xue, Bing-Li Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Jiong Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, M. Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaokang Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wen-Xia Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyu Jin, Xi-Cheng Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yi Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yu-Jing Zou, Yujia He, Yunfan Xiong, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yao Li, Yi Zheng, Yuchen Zhu, Yunxiang Ma, Ying Tang, Yukun Zha, Yuting Yan, Zehui Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhen guo Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zi-An Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bing-Li Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junmei Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shao-Ping Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wen-Xuan Yu, Wentao Zhang, X. Q. Li, Xiangyu Jin, Xianzu Wang, Xiaoling Bi, Xiaodong Liu, Xiaohan Wang, Xi-Cheng Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yao Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yi Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yi-Bing Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxiang Ma, Yuting Yan, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Zehui Ren, Zehui

Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhen guo Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zi-An Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024. Deepseek-v3 technical report. *ArXiv*, abs/2412.19437.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. Baseline defenses for adversarial attacks against aligned language models. *Preprint*, arXiv:2309.00614.

Akshita Jha and Chandan K. Reddy. 2022. Codeattack: Code-based adversarial attacks for pre-trained programming language models. In *AAAI Conference on Artificial Intelligence*.

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*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Quan Liu, Zhenhong Zhou, Longzhu He, Yi Liu, Wei Zhang, and Sen Su. 2024. Alignment-enhanced decoding: Defending jailbreaks via token-level adaptive refining of probability distributions. In *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.

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. 2022a. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022b. Training language models to follow instructions with human feedback. In *NeurIPS*.

R. Penrose. 1955. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Baptiste Rozière, Naman Goyal, Piotr Batra, Pierre Mazaré, Jean Jégou, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2307.09288*. https://arxiv.org/abs/2307.09288.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. 2023. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5:1486–1496.

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

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

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.

Limitations

While our proposed SCOT-LoraAlign demonstrates significant improvements in safety alignment for reasoning models, several limitations warrant discussion.

Dependency on SCOT Dataset Quality: The construction of SCOT-zero relies on a manually curated dataset of safety-focused chain-of-thought (SCOT) examples. While we designed structured prompts to guide SCOT generation, the dataset's coverage of diverse harmful categories and novel attack patterns may be incomplete. Biases or gaps in the SCOT data could limit the model's ability to generalize to emerging or highly adversarial threats.

Trade-offs in Low-Rank Approximation: Although SCOT-LoRA effectively reduces computational overhead by converting SCOT into low-rank parameters, this approximation may constrain the expressiveness of safety reasoning.

Long-Term Stability of Parameter Updates: Repeated low-rank parameter fusion could lead to cumulative shifts in model behavior over time. While our experiments show minimal alignment tax in short-term evaluations, prolonged usage might degrade performance on downstream tasks or introduce unintended biases, necessitating periodic recalibration.

Limited Evaluation on Multilingual Scenarios: SCOT-LoRA focuses on updating the model at test time as a form of patch and cannot completely replace the training process. Periodically using the SCOT-LoRA data to retrain at training times results in better results. Our validation is conducted exclusively on English-language datasets. The method's effectiveness in non-English contexts, where cultural norms and harmful content definitions differ, remains unexplored.

Addressing these limitations would further enhance the robustness and applicability of safety alignment frameworks for reasoning models. Future work We will address the issue of catastrophic forgetting in SCoT-Align as well as its collapse after multiple iterations and expand SCOT datasets to cover broader threat landscapes.

A Derivation and Proof

In this section, we describe and derive the formula for calculating equivalent low-rank knowledge parameters and prove the validity of the method.

For the original model, the computation in the l-th MLP layer during the inference process for queries Q and Q' satisfies the following equation:

$$WX_l^q + b_l = Y_l^q, \quad WX_l^{sCoT_q} + b_l = Y^{sCoT_q} \tag{12}$$

When the model is updated with ΔW, as determined by the target formula 1, for the original input Q, the hidden vectors calculated with the updated parameter should match those calculated in the original parameter for the input $Q' + SCoTq$, which is integrated SCoT into the context. This is formally represented as:

$$(W + \Delta W)X_l^q + b_l = Y_{l+1}^{sCoT_q} \tag{13}$$

Based on this target formula 13, we compute the equivalent parameters ΔW necessary for model updates.ΔW can be further formalized and represented as follows:

$$\Delta Y_l = Y_l^{sCoT_q} - Y_l^q, \quad \Delta X_l = X_l^{sCoT_q} - X_l^q$$

$$\Delta W X_l = \Delta Y_l = W \Delta Y_l \tag{14}$$

$$\implies \Delta W = W \Delta Y_l X_l^{-1} \tag{15}$$

However, in most cases, where the number of queries does not equal the dimensionality of the hidden vectors, therefore X is not a square matrix, and hence an inverse X_l^{-1} does not exist directly.

For this purpose, we compute the pseudoinverse of X using the Penrose pseudoinverse as showned in formula 2, which satisfies the requirement for calculating ΔW. The equivalence found in 3.1 proves the validity of ΔW.

Once we have obtained the pseudoinverse matrix X_l^{-1}, we can directly compute the equivalent parameter ΔW, achieving the alignment of the model. Ultimately, ΔW can be derived using the formula presented below:

$$\Delta W = W \Delta X (V_r \Sigma_r^{-1} U_r^T) \tag{16}$$

We then add the computed equivalent parameter ΔW to the model's original parameter W to implement sustainability updates of the LLMs' parameters.