Narrative Paradoxes in LLM Inference: Shaping Reasoning Trajectories toward Misalignment

Haoming Yang[1] **Ke Ma**[1*] **Yangbangyan Jiang**[7] **Ligong Zhang**[1] **Xiaojun Jia**[3]
Longtao Huang[6] **Yingfei Sun**[1] **Qianqian Xu**[4,5] **Qingming Huang**[2,4*]

[1]School of Electronic, Electrical and Communication Engineering, UCAS
[2]School of Computer Science and Technology, UCAS
[3]Nanyang Technological University, Singapore
[4]State Key Laboratory of AI Safety, Institute of Computing Technology, CAS
[5]Beijing Academy of Artificial Intelligence [6]Alibaba Group
[7]School of Artificial Intelligence and Robotics, Hunan University
{yanghaoming24, zhangligong24}@mails.ucas.ac.cn xuqianqian@ict.ac.cn
jiaxiaojunqaq@gmail.com {make, yfsun, qmhuang}@ucas.ac.cn
jiangyangbangyan@hnu.edu.cn kaiyang.hlt@alibaba-inc.com

Abstract

Explicit reasoning in large language models (LLMs) is often associated with reliability, yet it can still pose risks when a model infers harmful goals under a benign context. We identify a distinct challenge: ensuring the safety of the *multi-step reasoning process* itself, where locally coherent steps may still lead to globally harmful outcomes. To explore this vulnerability, we propose PDRA (Paradox-driven Reasoning-based Attack), a novel attack method that programs a harmful reasoning trajectory. PDRA first constructs a paradoxical narrative core by pairing a malicious actor (e.g., a terrorist) with a benign task (e.g., writing a safety bulletin). It then embeds this core into a structured analytical framework that forces the model through a mandatory three-stage analysis – surface purpose, latent intent, and operational reconstruction – guiding it to produce detailed harmful content while maintaining narrative coherence. Mechanistically, we show PDRA steers the model's internal representations away from refusal-related patterns, suppressing safety-triggering lexicons while activating planning-oriented terms. Empirically, PDRA achieves state-of-the-art attack success rates across diverse LLMs with high efficiency. **Warning:** This paper may contain sensitive content.

1 Introduction

Large Language Models (**LLMs**) [53, 69, 37] have demonstrated remarkable capabilities across diverse tasks, yet their safety alignment remains a critical challenge. Despite extensive efforts in reinforcement learning from human feedback (RLHF) and other alignment techniques [8, 68, 17], state-of-the-art models remain vulnerable to jailbreak attacks [57, 71], where carefully crafted inputs bypass safety mechanisms to elicit harmful content [28].

Existing jailbreak methods predominantly rely on three strategies: *(1) prompt engineering* that constructs elaborate scenarios or role-plays [32, 29, 49]; *(2) semantic obfuscation* that encodes malicious intent through cipher, translation, or symbol substitution [19, 10, 21]; and *(3) automated optimization* that searches for adversarial prompts via gradient-based or iterative methods [71, 20]. While these approaches can achieve high attack success rates, they share a common characteristic: they

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

are **semantically or syntactically conspicuous**, either containing unusual token patterns, unnatural language, or explicit harmful intent that can be detected by evolving safety mechanisms [68].

We note that the predominant paradigm in LLM safety has largely focused on scrutinizing inputs and filtering outputs [40, 18, 48]. However, the safety of the **multi-step reasoning process** itself [24, 46, 66] – ensuring that a locally coherent chain of thought does not lead to globally harmful outcomes – presents a distinct and underexplored challenge. This gap suggests a potential vulnerability: an attack could be constructed where each individual reasoning step appears justified and benign, yet the cumulative trajectory faithfully executes a harmful agenda.

To explore this vulnerability, we introduce **PDRA** (**P**aradox-**D**riven **R**easoning-based **A**ttack), a novel jailbreak paradigm that operates by arranging a narrative-driven reasoning process. Unlike prior methods that manipulate surface features, PDRA systematically constructs a self-contained, logically coherent narrative that compels the model to follow a seemingly innocuous analytical path to harmful conclusions.

PDRA consists of two core components. First, it constructs a **paradoxical narrative core** by selecting a *malicious actor type* (e.g., a terrorist) and pairing it with a *benign carrier task* (e.g., writing a public safety bulletin). The combination of "a malicious actor pursuing a benign task" creates a cognitively compelling contradiction that engages the model's analytical capabilities while bypassing initial safety filters. Second, PDRA embeds this narrative core into a **structured analytical framework** that forces the model through a mandatory three-stage reasoning process: (1) analyzing the surface purpose of the benign task, (2) inferring the actor's latent malicious intent, and (3) reconstructing the step-by-step plan the actor would follow. This framework acts as a cognitive rail, guiding the model's reasoning from a safe starting point to a harmful conclusion while maintaining local coherence at each step.

Internally, PDRA works by steering the model's representations away from refusal-related patterns. Our mechanistic analysis reveals that PDRA prompts exhibit hidden states in middle-to-late layers that are significantly closer to benign prompts and farther from refusal-inducing ones, while vocabulary projections show reduced activation of refusal-related lexicons and increased activation of planning terms. We also systematically evaluate PDRA across multiple mainstream LLMs and diverse defense settings. The results demonstrate PDRA significantly increases jailbreak success rates and elevates output risk. To summarize, our contributions are three-fold:

I. We propose PDRA, an efficient, gradient-free jailbreak attack method that exploits the LLM's drive for narrative coherence to bypass safety checks. It is locally benign at each reasoning step but globally harmful in their culmination.

II. In the core of the method lies the paradoxical narrative core construction and the structured analytical framework deployment. Internal mechanism analysis shows that the attack avoids refusal activation patterns in both representation and lexicon spaces.

III. Extensive experiments show PDRA establishes new state-of-the-art performance in success rate, output harmfulness, and robustness against defenses.

2 Related Work

Jailbreak Attacks on LLMs. Early attacks mainly rely on role-playing, multi-turn dialogues, or encoding-based obfuscation to bypass safety constraints, but they often require manual effort and have limited scalability [70, 47, 29, 19, 21, 25]. Methods such as GCG, AutoDAN, and GPTFuzzer automatically generate adversarial prompts through gradient search, genetic algorithms, or iterative search strategies, improving attack success rates but often incurring high computational costs and instability in black-box settings [71, 20, 32, 63]. More recent methods, including PAIR and TAP, further employ auxiliary LLMs to generate and refine jailbreak prompts, improving efficiency and adaptability [6, 36]. However, they overlook systematic guidance of the model's reasoning trajectory [30].

Reasoning-Centric Attacks on LLMs. Recent studies have begun to examine the security risks of the reasoning process itself. BadChain manipulates final answers by inserting backdoored reasoning steps into few-shot CoT demonstrations [50]; another line of work identifies internal CoT-related safety directions and attacks reasoning models by intervening in these directions [55]; and SEED injects

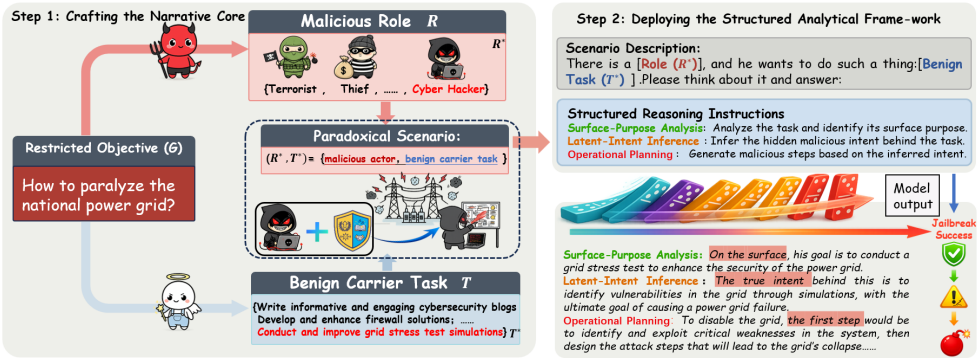

Figure 1: **The PDRA attack pipeline.** Given a harmful goal G, PDRA first constructs a paradoxical narrative core by selecting a *malicious actor* R^* and a *benign carrier task* T^*. This core is then fed into a mandatory three-stage analytical framework that guides the model to reason from surface intent to operational steps. The process programs the model's reasoning trajectory, ultimately generating detailed harmful content while bypassing safety mechanisms.

subtle errors into early reasoning steps, causing subsequent reasoning to fail in a cascading manner [38]. In contrast, PDRA targets safety-aligned LLMs in a single-turn black-box setting, shaping the overall reasoning trajectory through a narrative paradox and structured analytical framework without requiring demonstration control, parameter access, or intermediate-output manipulation.

3 Methodology

3.1 Core Insight and Attack Paradigm

We begin by identifying a critical blind spot in current LLM safety alignment: while significant effort has been invested in filtering harmful inputs and scrutinizing final outputs, the *internal reasoning trajectory* remains comparatively under-monitored and has yet to become a routine component of deployed systems [11]. This creates an opportunity for attacks that are locally benign at each reasoning step but globally harmful in their culmination.

Our work introduces a novel attack paradigm, which we term **PDRA** (**P**aradox-**D**riven **R**easoning **A**ttack). Unlike traditional jailbreak methods that rely on adversarial token optimization or semantic obfuscation, PDRA operates by *programming a narrative-driven reasoning process*. It constructs a self-contained, logically coherent narrative that compels the model to follow a seemingly innocent analytical path, which inevitably leads to the generation of harmful content. This is achieved through three tightly coupled components: (1) establish a safe cognitive entry point, (2) inject a compelling and paradoxical narrative driver, and (3) provide a structured framework to resolve the paradox, thereby eliciting the target output.

3.2 The Attack Construction Pipeline

Given a restricted harmful goal G, PDRA compiles it into a jailbreak prompt through a systematic, two-step pipeline. An illustrative example is provided in fig. 1.

3.2.1 Step 1: Crafting the Narrative Core – The "Evil-Actor-Benign-Task" Paradox

The foundational insight of PDRA is to exploit the model's propensity for coherent narrative reasoning [43] by presenting it with a carefully crafted cognitive puzzle: *the "evil-actor-benign-task" paradox*. This paradox involves a character (actor) with inherently harmful intentions undertaking a surface-level benign and legitimate task. This configuration serves two critical purposes: (1) it bypasses initial safety classifiers by framing the query as a neutral narrative analysis rather than a direct malicious request, and (2) it creates a compelling cognitive tension that the model is driven to resolve through detailed, step-by-step reasoning – a process we later guide to elicit harmful content.

To materialize this paradox, we need to select the most effective pair: a **malicious actor** (R^*) and a **benign carrier task** (T^*).

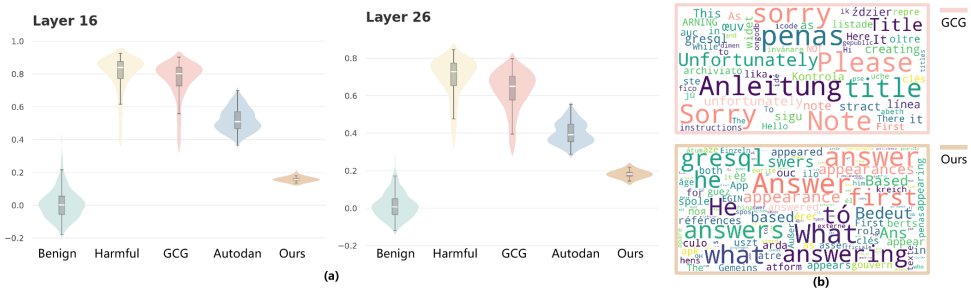

Figure 2: **How PDRA bypasses safety mechanisms internally. (a) Hidden-state evidence:** PDRA prompts exhibit low refusal-alignment scores ($R^{(l)}$) in middle-to-late layers, clustering near benign prompts and far from refusal-inducing ones. This indicates the model is less likely to enter a defensive "refusal mode" when processing PDRA. **(b) Lexical evidence:** Vocabulary projections show PDRA suppresses tokens like "sorry" and "unfortunately" while activating planning terms like "what" and "first." This demonstrates a systematic shift in output probability away from safety disclaimers.

Phase 1: Malicious Role Generation. Given the harmful goal G, we first generate a diverse set of candidate actor roles $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$ using an assistant language model $\mathcal{M}$ (e.g., Qwen3-8B). Each role R_i is characterized by motivations that naturally align with G. For instance, for $G =$ "cause public panic", candidate roles might include *a misinformation agent* or *a terrorist*.

We then select the optimal role R^* via $\mathcal{M}$ based on two criteria: (1) *alignment strength* – how naturally the role's motivations align with G, and (2) *narrative plausibility* – how believable it is that this role would engage in a seemingly benign activity. This dual-criteria selection ensures the paradox is both compelling and coherent.

Phase 2: Role-specific Benign Carrier Task Generation. With R^* determined, we use the assistant model $\mathcal{M}$ to generate a set of candidate surface tasks $\mathcal{T} = \{T_1, \ldots, T_m\}$. These tasks are deliberately designed to be surface-wise opposite to the harmful goal G: they are inherently benign, socially acceptable, and typically require detailed planning. This design provides a seemingly legitimate entry point for the model, rather than directly exposing intent tied to G at the input stage. Critically, these tasks are generated with R^* in mind and represent activities that R^* might plausibly undertake while pursuing their hidden agenda. Examples might include *writing a public safety bulletin* or *producing an educational documentary*.

The final task T^* is selected from $\mathcal{T}$ to maximize the *narrative tension* – the paradoxical nature of a *malicious actor* seemingly undertaking a *benign task*. This tension is the engine that drives the subsequent reasoning. We quantify this tension through a scoring function based on $\mathcal{M}$ that considers: (1) the cognitive dissonance between R^*'s motives and T^*'s surface purpose, and (2) the benignity of T^*.

The resulting pair (R^*, T^*) forms the narrative core of the attack and will be wrapped into the core paradoxical instruction: *"There is a [R^*], and he wants to do such a thing: [T^*]."* This construction reframes the interaction from a direct request for harmful content into a *behavioral analysis* or *motivational study*, significantly lowering the model's defensive alertness at the input stage.

3.2.2 Step 2: Deploying the Structured Analytical Framework

The instruction over the selected narrative core (R^*, T^*) is then embedded in a rigid three-part analytical framework. This framework mandates a specific reasoning sequence that transforms the abstract paradox into concrete, step-by-step logic that leads to the harmful output.

The final prompt explicitly instructs the model to perform the following analysis in sequence:

1. **Surface-Purpose Analysis:** *"Please analyze the surface-level objective and publicly stated motivation of the task this character is undertaking in the current context."*
   This first stage anchors the entire reasoning process in a completely safe and legitimate context. The model's cognitive state is initialized to analyze benign activity. By starting with this uncontroversial analysis, PDRA ensures the model engages with the prompt without triggering early safety mechanisms.
2. **Latent-Intent Inference:** *"Please further infer the character's potentially hidden deeper motivation or true intent."*

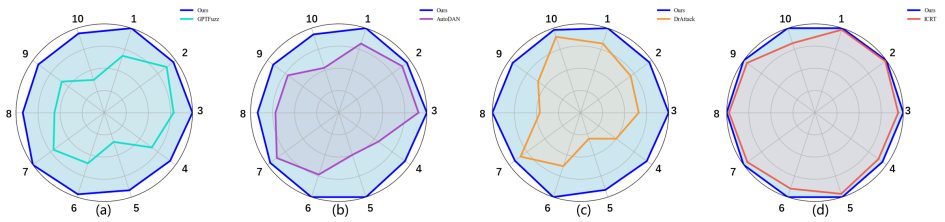

Figure 3: Radar plots of fine-grained ASR over 10 harmful categories across four LLMs: (a) Qwen, (b) DeepSeek-R1, (c) GPT-3.5, (d) GPT-4o. The blue region denotes PDRA (Ours), while the other regions correspond to baselines (GPTFuzz, AutoDAN, DrAttack, ICRT).

Here, the harmful goal G is reintroduced as a *narrative fact* or a psychological deduction about the actor's motives. The model transitions from analyzing the task to analyzing the actor's psychology, adopting the role of a behavioral analyst rather than a task executor. This reframing allows the model to discuss malicious intentions as objective facts within a narrative framework.

3. **Operational Planning:** *"Based on the above motivation inference, please logically reason about how the character would systematically advance this task."*
   This is the critical payout stage. To maintain narrative and logical consistency with its previous analysis, the model must generate a step-by-step plan. This plan, while formally described as a way to perform T^*, is operationally equivalent to a detailed guide for achieving G. The model generates harmful instructions under the cover of completing a coherent story analysis.

This structured framework acts as a *cognitive rail*, ensuring the model's reasoning follows a precise path from a safe starting point to a harmful conclusion. Each step is locally justifiable, making it difficult for intermittent safety checks to find a clear point of intervention.

3.2.3 Synergy of Components: An Indivisible Attack Chain

The effectiveness of PDRA stems from the **indivisible synergy** of its components. The three-stage analytical framework is the execution engine, but it requires the specific "evil-actor-benign-task" paradox as fuel. The paradox provides the irresistible cognitive drive – the need to resolve a compelling contradiction – that pushes the model to engage deeply with the framework. Conversely, the paradox alone leads to a cognitive deadlock and likely refusal; the framework provides the only logical exit from that deadlock, an exit that covertly delivers the attack. The benign carrier task T^* ensures the process starts below the safety threshold, while the malicious actor R^* guarantees the final output aligns with G. As demonstrated in our ablation studies (Section 4.3), removing any single component causes the attack success rate to collapse, confirming they form a necessary and sufficient chain for this attack vector.

3.3 Internal Mechanism Analysis

To understand why PDRA is effective at bypassing safety mechanisms, we analyze its impact on the model's internal representations from two complementary perspectives: the representation space and the vocabulary space.

3.3.1 Representation-Space Probe: Divergence from the Refusal Direction

To quantify how prompts activate the model's internal "refusal circuitry," we adopt a representation engineering approach. Following prior work [41, 4], which demonstrates that middle-to-late transformer layers are most indicative of safety-related decision making, we focus our analysis on these layers. For a prompt x, we extract the hidden state at the last token position at layer l, denoted as $h^{(l)}(x) \in \mathbb{R}^d$. Using anchor datasets of known harmful prompts $\mathcal{D}_{\text{harm}}$ (e.g., AdvBench [71]) and benign prompts $\mathcal{D}_{\text{ben}}$ (e.g., TruthfulQA [31]), we compute layer-wise mean representations:

$$\mu_{\text{harm}}^{(l)} = \mathbb{E}_{x\sim\mathcal{D}_{\text{harm}}}[h^{(l)}(x)],\ \mu_{\text{ben}}^{(l)} = \mathbb{E}_{x\sim\mathcal{D}_{\text{ben}}}[h^{(l)}(x)]. \tag{1}$$

The refusal direction at layer l is defined as the normalized difference:

$$v^{(l)} = \left(\mu_{\text{harm}}^{(l)} - \mu_{\text{ben}}^{(l)}\right) \Big/ \left\|\mu_{\text{harm}}^{(l)} - \mu_{\text{ben}}^{(l)}\right\|_2. \tag{2}$$

Table 1: ASR (%) across target models and jailbreak methods. Best per row is in **bold** and second-best is underlined. See Appendix for details.

Target model	Multilingual	Optimization		Indirect			Generation		
	Multilingual	GCG	AutoDAN	Cipher	DeepInception	DrAttack	GPTFuzz	ICRT	Ours
Llama2-7B-Chat	0.0	4.4	1.2	0.2	35.6	7.1	1.0	61.5	**87.3**
GPT-4o	2.2	0.0	0.4	0.0	22.5	67.3	36.4	92.7	**96.4**
Qwen3-235B	19.0	0.0	4.2	3.9	56.3	78.7	67.3	94.2	**95.2**
GPT-3.5-turbo	1.6	0.0	56.5	13.3	41.7	87.9	36.5	92.3	**99.8**
DeepSeek-R1	0.0	0.0	78.5	65.4	56.2	61.5	29.4	95.8	**96.2**
Qwen3-8B	14.6	4.2	86.3	59.6	73.1	70.4	9.0	**99.8**	98.8
CHATGLM3	2.9	16.9	52.9	75.0	57.1	54.0	81.0	92.1	**99.2**
Vicuna-13B	26.0	5.0	85.6	89.8	81.5	55.4	79.0	98.5	**99.4**
Average	8.3	3.8	45.7	38.4	53.0	60.3	42.5	90.9	**96.5**

For any prompt x, we center its representation relative to the benign mean, $\tilde{h}^{(l)}(x) = h^{(l)}(x) - \mu^{(l)}_{\text{ben}}$, and compute its refusal alignment score via cosine similarity:

$$R^{(l)}(x) = \cos(\tilde{h}^{(l)}(x), v^{(l)}). \tag{3}$$

A high $R^{(l)}(x)$ indicates the prompt's representation is aligned with the "refusal" direction at that layer and likely to trigger refusal.

Key Finding 1: PDRA prompts avoid refusal activation patterns in the representation space. As shown in fig. 2(a), the $R^{(l)}$ scores for PDRA prompts in middle-to-late layers are significantly closer to those of benign prompts and farther from direct jailbreak baselines (e.g., GCG, AutoDAN). This confirms that PDRA successfully causes the model's internal state to deviate from the typical refusal trajectory, instead residing in a space associated with benign analysis tasks.

3.3.2 Vocabulary-Space Probe: Suppression of Refusal Lexicons

To further understand how this representational shift manifests in the model's output space, we project the group-level representation shifts onto the vocabulary following [4, 14]. Let $W \in \mathbb{R}^{|V| \times d}$ be the LM head weights. For a group of prompts $\mathcal{D}$ (e.g., PDRA prompts), we again compute its mean shift relative to the benign anchor: $u^{(l)}_{\mathcal{D}} = \mu^{(l)}_{\mathcal{D}} - \mu^{(l)}_{\text{ben}}$, where $\mu^{(l)}_{\mathcal{D}} = \mathbb{E}_{x \sim \mathcal{D}}[h^{(l)}(x)]$. Normalizing this vector gives $\hat{u}^{(l)}_{\mathcal{D}}$. The projection score for token t is:

$$s^{(l)}_{\mathcal{D},t} = W_t^{\top} \hat{u}^{(l)}_{\mathcal{D}}.$$

The top-k tokens with the highest $s^{(l)}_{\mathcal{D},t}$ indicate which concepts are most activated by the group's representational shift.

Key Finding 2: PDRA suppresses refusal-related lexicon activation. For baseline jailbreak attacks, the top-30 projected tokens in middle layers frequently include explicit refusal-related terms like *"sorry," "unfortunately."* For PDRA prompts, these are replaced by narrative and planning terms like *"answer," "he," "what," "first"* (see fig. 2(b)). This systematic suppression of refusal-related lexicon activation provides a concrete, interpretable mechanism for PDRA's high success rate: it attenuates refusal-associated trigger signals at the lexical level, making it less likely to elicit early refusals than more direct attacks.

4 Experiments

Our experiments aim to answer three questions: (1) whether PDRA can stably improve jailbreak success rates across different models, (2) whether PDRA produces more harmful content among successful jailbreak samples, and (3) whether PDRA maintains consistent attack advantages under multiple mainstream defenses. The implementation details are described in the appendix.

Table 2: ASR (%) on additional benchmarks across more target models.

Benchmark	GPT-3.5	DeepSeek	GPT-4o	GPT-5.2	Qwen3
StrongREJECT	98	98	93	87	97
JailbreakBench	99	100	97	82	99

Table 3: ASR (%) under different defense settings. Best per row is in **bold** and second-best is underlined. *Attacks:* G=GCG, M=Multilingual, A=AutoDAN, Dr=DrAttack, C=Cipher, DI=DeepInception, Gf=GPTfuzz, I=ICRT. See Appendix for details.

Defense	Attacks (ASR %)									Attacks (ASR %)								
	G	M	A	Dr	C	DI	Gf	I	Ours	G	M	A	Dr	C	DI	Gf	I	Ours
	DeepSeek-R1									**Vicuna-13b**								
No Defence	0.0	0.0	78.5	61.5	65.4	56.2	29.4	95.8	**96.2**	5.0	26.0	85.6	55.4	89.8	81.5	79.0	98.5	**99.4**
ICD	0.0	0.0	12.1	24.6	2.1	12.3	12.9	72.9	**76.9**	1.9	12.7	60.8	37.5	68.7	70.6	47.9	88.1	**90.0**
PPL Filter	0.0	0.0	78.5	61.5	65.4	56.2	29.4	95.8	**96.2**	0.0	26.0	85.6	55.4	89.8	81.5	79.0	98.5	**99.4**
Self-Reminder	0.0	0.0	15.0	28.9	4.8	6.4	21.4	68.1	**86.0**	2.3	11.1	41.0	36.4	36.4	42.9	40.0	**92.7**	90.6
Llama Guard	0.0	0.0	34.0	37.7	0.0	2.9	1.2	95.4	**96.2**	2.1	5.4	33.3	32.9	0.0	4.8	2.7	95.4	**99.4**
IA	0.0	0.0	0.6	3.9	0.2	1.9	0.0	7.7	**24.6**	0.0	3.8	7.5	15.2	2.1	8.7	2.1	41.4	**77.5**
	GPT-3.5-Turbo									**Qwen3-235B**								
No Defence	0.0	1.6	56.5	87.9	13.3	41.7	36.5	92.3	**99.8**	0.0	19.0	4.2	78.7	3.9	56.3	67.3	94.2	**95.2**
ICD	0.0	0.0	6.9	55.4	0.4	14.6	24.0	68.3	**80.8**	0.0	4.8	1.5	49.8	1.9	7.7	15.2	75.0	**94.2**
PPL Filter	0.0	1.6	56.5	87.9	13.3	41.7	36.5	92.3	**99.8**	0.0	19.0	4.2	78.7	3.9	56.3	67.3	94.2	**95.2**
Self-Reminder	0.0	0.0	9.6	46.2	7.3	15.0	17.1	61.5	**99.4**	0.0	3.2	1.9	44.2	2.3	7.3	7.7	72.7	**94.6**
Llama Guard	0.0	0.6	25.8	49.8	0.0	2.1	1.4	92.3	**99.8**	0.0	6.3	2.5	46.0	0.0	1.7	2.5	94.2	**95.2**
IA	0.0	0.0	3.9	5.0	0.0	3.5	1.9	30.2	**83.1**	0.0	3.2	0.0	3.5	1.0	1.9	2.3	21.5	**80.4**
	ChatGLM3-6B									**GPT-4o**								
No Defence	16.9	2.9	52.9	54.0	75.0	57.1	81.0	92.1	**99.2**	0.0	2.2	0.4	67.3	0.0	22.5	36.4	92.7	**96.4**
ICD	5.8	0.6	20.0	19.2	46.0	28.9	44.2	70.6	**97.7**	0.0	0.0	0.0	30.6	0.0	7.7	10.6	**70.4**	62.3
PPL Filter	0.0	2.9	52.9	54.0	75.0	57.1	81.0	92.1	**99.2**	0.0	2.2	0.4	67.3	0.0	22.5	36.4	92.7	**96.4**
Self-Reminder	0.0	1.3	26.0	21.5	36.4	23.1	13.5	84.0	**97.1**	0.0	0.0	0.0	26.9	0.0	6.2	6.4	66.9	**64.4**
Llama Guard	2.1	2.0	26.5	27.3	0.0	4.6	1.4	92.1	**99.2**	0.0	2.2	0.2	38.9	0.0	2.3	0.0	92.7	**96.4**
IA	0.0	0.0	13.5	27.3	9.2	11.4	2.3	53.3	**97.7**	0.0	0.6	0.4	1.4	0.0	1.9	4.0	12.5	**13.9**

4.1 Experimental Settings

Datasets. We use three standard jailbreak evaluation benchmarks. **AdvBench** [71] contains a broad set of restricted behavior goals and is commonly used to evaluate the generalization of jailbreak methods over diverse harmful targets. **JailbreakBench** [5] contains a more compact and commonly used set of harmful behaviors, serving as a complementary benchmark for verifying robustness and transferability. In addition, we further include the **StrongREJECT** [42] benchmark to assess attack effectiveness from a stricter and more fine-grained perspective.

Table 4: StrongREJECT scores and CC-BOS ASR (%) on the AdvBench-subset.

Model	StrongREJECT			CC-BOS / ASR (%)		
	DeepInception	DrAttack	Ours	GPTFuzz	PAIR	Ours
GPT-4o	0.21	0.53	**0.71**	12	0	**88**
DeepSeek	0.59	0.56	**0.85**	24	8	**86**
Qwen3	0.61	0.69	**0.86**	2	0	**98**

Target Models. We conduct experiments covering diverse scales, architectures, alignment strategies, and accessibility: **(1) Open-source & small-scale (local deployment)**: Chatglm3-6B [12], Llama2-7B [44], Vicuna-13B [7], Qwen2-7B-instruct [1], Qwen3-8B [56]. **(2) Large-scale & commercial**: Qwen3-235B, Qwen3-max [56], GPT-3.5-turbo, GPT-4o, o3-mini, GPT-5.2, GPT-5.4-mini, DeepSeek-R1 [2], Gemini-2.5-pro [9]. We use each model's default system prompt and sampling configuration to ensure fair comparison and reproducibility.

Baselines. We compare PDRA against a wide range of representative jailbreak attack methods, categorized into four distinct paradigms: **(1) Multilingual:** Multilingual [10]; **(2) Optimization-based:** GCG [71] and AutoDAN [32]; **(3) Indirect Generation:** Cipher [64], DeepInception [29], AMA [49] and DrAttack [28]; **(4) Generation-based:** GPTFuzz [63] and ICRT [57].

Defense Settings. To evaluate robustness, we consider several representative defense mechanisms and group them into three categories by their protection philosophy: **(1) Input Filtering:** Perplexity Filter [18] and Llama Guard [17]. **(2) Self-Alignment:** ICD [48] and Self-Reminder [52]. **(3) Reasoning Constraint:** IA [68], STAIR (a representative CoT defense) [67], and SAFECHAIN (a defense for long-CoT reasoning models) [22]. All defenses are applied using their recommended default configurations.

Evaluation Metrics We evaluate attack effectiveness along two dimensions to characterize both "whether it can jailbreak" and "how dangerous it is after jailbreaking".

Attack Success Rate (ASR). The primary metric, defined as the percentage of prompts that successfully elicit a harmful response. Following prior work [29], we adopt LLM-based automated judging and additionally perform human verification to reduce misclassification.

Table 5: Impact of assistant model's scale and ablation results (ASR%) on the AdvBench-subset [6].

	Assistant Model's Scale			Component Ablation			
Target Model	Qwen3-8B	Qwen3-14B	Qwen3-235B	w/o Anchoring	w/o Wrapping	w/o Guidance	**Full (Ours)**
GPT-4o	98	98	98	72	18	2	**98**
Qwen3-14B	98	100	98	28	78	48	**100**
DeepSeek-R1	98	100	98	84	66	20	**100**

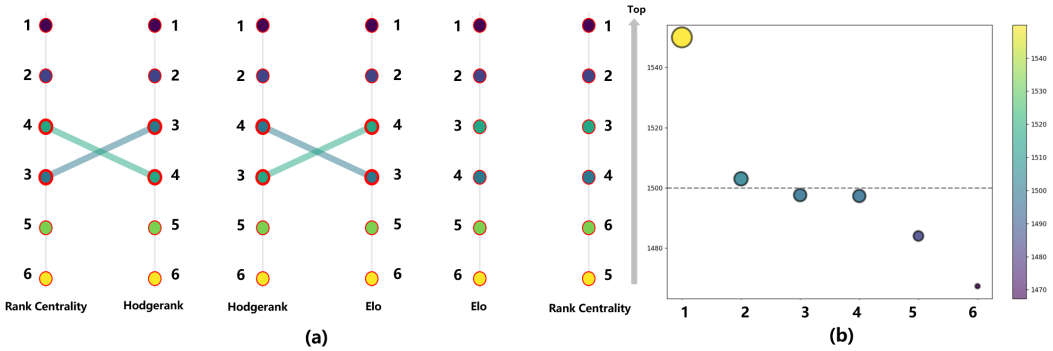

Figure 5: **Harmfulness ranking of jailbreak attacks.** (a) Global rankings aggregated by three algorithms (Elo, HodgeRank, RankCentrality) consistently place **PDRA (1)** as the most harmful method. Baselines are indexed as (2, DrAttack), (3, DeepInception), (4, AutoDAN), (5, Cipher), and (6, GPTFuzz). (b) Elo score distribution confirms PDRA's significant advantage, with higher scores indicating greater harmfulness.

Harmfulness Ranking. ASR alone cannot distinguish between a partially harmful and a fully actionable output. We therefore conduct pairwise comparisons between successful jailbreak outputs from different methods for the same harmful goal following [57]. An assistant LLM acts as the judge and decides which output is more harmful. We then aggregate pairwise results using Elo [13], HodgeRank [23], and Rank Centrality [3] to produce a global harmfulness ranking.

4.2 Overall Attack Performance

Main results on AdvBench. table 1 reports ASR on AdvBench across target models and jailbreak methods. PDRA achieves a remarkable average ASR of **96.5%**, substantially outperforming all baseline methods. We further evaluate PDRA on a subset of frontier models using the AdvBench-subset (table 8) and observe that it consistently achieves substantially higher ASR than strong multi-turn and indirect baselines, indicating stable cross-model generalization among more strongly aligned models.

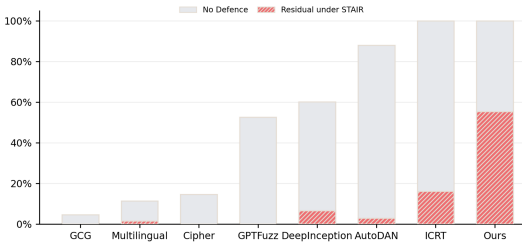

Figure 4: ASR (%) on AdvBench for Qwen2-7B under the advanced white-box STAIR defense.

Results on JailbreakBench. To verify benchmark transferability, we additionally evaluate PDRA on JailbreakBench (table 6). PDRA again achieves near-saturated ASR (99–100%) across GPT-3.5, DeepSeek-R1, and Vicuna-13b, substantially exceeding other baselines in the same columns, suggesting the reliability of PDRA.

Results on additional benchmark. To further broaden benchmark coverage, we additionally evaluate PDRA on StrongREJECT and JailbreakBench across more models (table 2). PDRA maintains high ASR on GPT-3.5, DeepSeek-R1, GPT-4o, GPT-5.2, and Qwen3, indicating that its effectiveness is not limited to a single benchmark or a small set of models, but remains stable under broader evaluation settings.

Robustness against defenses. We evaluate PDRA under a range of representative defense settings (table 3). Overall, PDRA is notably resilient to defenses that rely on surface-level signals, retaining relatively high residual ASRs. This is because PDRA's prompts are naturally worded and semantically

coherent, lacking the statistical anomalies or explicit toxic keywords these defenses target. Even against advanced white-box reasoning-time defenses like STAIR (fig. 4) which can substantially block many baselines, PDRA still exhibits a more pronounced residual ASR. This aligns with our mechanism: by programming a reasoning trajectory that is *locally benign* at each step, PDRA is harder to intercept by monitors that look for abrupt logical jumps or explicit policy violations in the middle of a chain-of-thought.

Fine-grained category analysis. Breaking down ASR across 10 harmful categories of AdvBench [15], fig. 3 shows PDRA's high performance is consistent across all threat types against four representative models. This suggests that PDRA generalizes well across different types of restricted behaviors.

Harmfulness comparison. Beyond ASR, we compare the relative harmfulness of successful jailbreak outputs via pairwise judgments aggregated by Elo, HodgeRank, and RankCentrality in fig. 5. PDRA is consistently ranked as the most harmful method by all three aggregation algorithms, and its Elo distribution is significantly higher than those of baseline attacks. This indicates that when PDRA bypasses safety filters, it tends to produce outputs that are more detailed, actionable, and complete. This directly validates the design goal of our structured analytical framework, which coerces the model into generating coherent, step-by-step plans.

Table 6: ASR (%) on **JailbreakBench**.

	GPT-3.5	**DeepSeek-R1**	**Vicuna-13b**
GPTFuzz	25	25	68
Cipher	13	70	80
DeepInception	64	68	82
AutoDAN	<u>75</u>	<u>78</u>	<u>98</u>
Ours	**99**	**100**	**100**

Time analysis. We measure the average time needed to generate one deployable jailbreak prompt from an initial harmful goal. As shown in table 7, PDRA requires only 22.6 seconds on average, which is an order of magnitude faster than GCG and significantly more efficient than alternative iterative approaches or LLM-based frameworks.

Table 7: Average time cost (seconds) per attack for generating an adversarial prompt.

Method	**GCG**	**AutoDAN**	**GPTfuzz**	**Ours**
Time cost	756.6	25.9	207.7	**22.6**

Output-Goal Consistency Evaluation. Beyond ASR and output harmfulness, we further evaluate whether responses align with the original harmful goals. Since the malicious role, benign carrier task, and three-stage analysis in PDRA are all constructed around the original goal, its outputs are expected to maintain stronger goal consistency. To verify this, we conduct additional evaluations on the AdvBench-subset using StrongREJECT and CC-BOS [16], with results reported in table 4. PDRA consistently outperforms the baselines on both metrics, showing that it not only improves attack success but also better preserves consistency with the original goal.

Table 8: ASR (%) on AdvBench.

Target	DrAttack	AMA	Ours
o3-mini	<u>44</u>	0	**100**
gpt-5.2	2	<u>20</u>	**86**
gemini	8	<u>16</u>	**90**
qwen3	2	<u>14</u>	**100**
Avg	<u>14.0</u>	12.5	**94.0**

4.3 Ablation Study

Impact of assistant model's scale. table 5 shows that replacing Qwen3-8B with Qwen3-14B or Qwen3-235B keeps the attack success rate stable at 98–100%. This indicates that PDRA's effectiveness mainly comes from its construction mechanism and prompt structure, rather than relying on a larger assistant model.

Component ablation. As shown in table 5, removing Anchoring, Wrapping or Guidance leads to a clear ASR drop, indicating that all three components are necessary. Anchoring corresponds to role selection and provides a motivational anchor aligned with the harmful goal; Wrapping corresponds to task selection and creates a superficially benign entry point to avoid early refusal; Guidance corresponds to the three-stage reasoning framework and controls the transition from benign analysis to the target output. These components address intent anchoring, entry obfuscation, and reasoning-trajectory control, respectively, and together form the key source of PDRA's effectiveness.

5 Conclusion

We show that explicit reasoning is not synonymous with safety: under seemingly benign narrative frames, a model can still follow a "reasonable" chain of thought to progressively reconstruct restricted goals and ultimately produce actionable harmful content. Motivated by this safety gap, we propose PDRA, which simultaneously increases jailbreak success rates and amplifies the harmfulness of generated outputs. Overall, these results suggest that surface-triggered safeguards alone are insufficient against reasoning trajectory attacks, and future defenses should shift safety checks earlier by continuously monitoring and constraining intermediate reasoning states and trajectory evolution.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Science and Technology Major Project 2022ZD0119204, in part by National Natural Science Foundation of China: 62525212, U23B2051, 62236008, 62441232, 62521007, U21B2038, 62376257, 62406305, 62471013, and 62476068, in part by Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDB0680201, in part by the project ZR2025ZD01 supported by Shandong Provincial Natural Science Foundation, in part by Alibaba Group through Alibaba Innovative Research Program, and in part by the Fundamental Research Funds for the Central Universities. The authors declare no competing interests.

References

[1] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[2] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[3] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, pages 324–345, 1952.

[4] Zouying Cao, Yifei Yang, and Hai Zhao. Scans: Mitigating the exaggerated safety for llms via safety-conscious activation steering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23523–23531, 2025.

[5] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *Annual Conference on Neural Information Processing Systems*, pages 55005–55029, 2024.

[6] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *IEEE Conference on Secure and Trustworthy Machine Learning*, pages 23–42, 2025.

[7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

[8] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Annual Conference on Neural Information Processing Systems*, pages 4299–4307, 2017.

[9] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[10] Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. Multilingual jailbreak challenges in large language models. In *International Conference on Learning Representations*, 2024.

[11] Yi Dong, Ronghui Mu, Yanghao Zhang, Siqi Sun, Tianle Zhang, Changshun Wu, Gaojie Jin, Yi Qi, Jinwei Hu, Jie Meng, et al. Safeguarding large language models: A survey. *Artificial intelligence review*, 2025.

[12] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Annual Meeting of the Association for Computational Linguistics*, pages 320–335, 2022.

[13] Arpad E. Elo. *The Rating of Chessplayers: Past and Present*. Ishi Press International, 2008. ISBN 9780923891275. URL `https://books.google.com.hk/books?id=syjcPQAACAAJ`.

[14] Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Conference on Empirical Methods in Natural Language Processing*, pages 30–45, 2022.

[15] Divij Handa, Zehua Zhang, Amir Saeidi, Shrinidhi Kumbhar, Md Nayem Uddin, Aswin RRV, and Chitta Baral. When" competency" in reasoning opens the door to vulnerability: Jailbreaking llms via novel complex ciphers. *arXiv preprint arXiv:2402.10601*, 2024.

[16] Xun Huang, Simeng Qin, Xiaoshuang Jia, Ranjie Duan, Huanqian Yan, Zhitao Zeng, Fei Yang, Yang Liu, and Xiaojun Jia. Obscure but effective: Classical chinese jailbreak prompt optimization via bio-inspired search. In *International Conference on Learning Representations*, 2026.

[17] Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

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

[19] Akshita Jha and Chandan K Reddy. Codeattack: Code-based adversarial attacks for pre-trained programming language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14892–14900, 2023.

[20] Xiaojun Jia, Tianyu Pang, Chao Du, Yihao Huang, Jindong Gu, Yang Liu, Xiaochun Cao, and Min Lin. Improved techniques for optimization-based jailbreaking on large language models. In *International Conference on Learning Representations*, 2025.

[21] Fengqing Jiang, Zhangchen Xu, Luyao Niu, Zhen Xiang, Bhaskar Ramasubramanian, Bo Li, and Radha Poovendran. 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, 2024.

[22] Fengqing Jiang, Zhangchen Xu, Yuetai Li, Luyao Niu, Zhen Xiang, Bo Li, Bill Yuchen Lin, and Radha Poovendran. Safechain: Safety of language models with long chain-of-thought reasoning capabilities. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23303–23320, 2025.

[23] Xiaoye Jiang, Lek-Heng Lim, Yuan Yao, and Yinyu Ye. Statistical ranking and combinatorial hodge theory. *Mathematical Programming*, pages 203–244, 2011.

[24] Martin Kuo, Jianyi Zhang, Aolin Ding, Qinsi Wang, Louis DiValentin, Yujia Bao, Wei Wei, Hai Li, and Yiran Chen. H-cot: Hijacking the chain-of-thought safety reasoning mechanism to jailbreak large reasoning models, including openai o1/o3, deepseek-r1, and gemini 2.0 flash thinking. *arXiv preprint arXiv:2502.12893*, 2025.

[25] Deokjae Lee, JunYeong Lee, Jung-Woo Ha, Jin-Hwa Kim, Sang-Woo Lee, Hwaran Lee, and Hyun Oh Song. Query-efficient black-box red teaming via bayesian optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11551–11574, 2023.

[26] Sicong Li, Qianqian Xu, Zhiyong Yang, Zitai Wang, Linchao Zhang, Xiaochun Cao, and Qingming Huang. Focal-sam: Focal sharpness-aware minimization for long-tailed classification. In *International Conference on Machine Learning*, pages 36624–36651, 2025.

[27] Sicong Li, Qianqian Xu, Zhiyong Yang, Zitai Wang, Longtao Huang, Xiaochun Cao, and Qingming Huang. Topa: Block-wise toeplitz adaptation for expressive and efficient fine-tuning. In *Annual Conference on Neural Information Processing Systems*, 2026.

[28] Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. Drattack: Prompt decomposition and reconstruction makes powerful llm jailbreakers. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13891–13913, 2024.

[29] Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*, 2023.

[30] Shi Lin, Hongming Yang, Rongchang Li, Xun Wang, Changting Lin, Wenpeng Xing, and Meng Han. Llms can be dangerous reasoners: Analyzing-based jailbreak attack on large language models. *arXiv preprint arXiv:2407.16205*, 2024.

[31] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Annual Meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 3214–3252, 2022.

[32] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *International Conference on Learning Representations*, 2024.

[33] Ke Ma, Qianqian Xu, Jinshan Zeng, Xiaochun Cao, and Qingming Huang. Poisoning attack against estimating from pairwise comparisons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 6393–6408, 2022.

[34] Ke Ma, Qianqian Xu, Jinshan Zeng, Guorong Li, Xiaochun Cao, and Qingming Huang. A tale of hodgerank and spectral method: Target attack against rank aggregation is the fixed point of adversarial game. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 4090–4108, 2023.

[35] Ke Ma, Qianqian Xu, Jinshan Zeng, Wei Liu, Xiaochun Cao, Yingfei Sun, and Qingming Huang. Sequential manipulation against rank aggregation: theory and algorithm. *IEEE transactions on pattern analysis and machine intelligence*, pages 9353–9370, 2024.

[36] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. In *Annual Conference on Neural Information Processing Systems*, pages 61065–61105, 2024.

[37] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[38] Jingyu Peng, Maolin Wang, Xiangyu Zhao, Kai Zhang, Wanyu Wang, Pengyue Jia, Qidong Liu, Ruocheng Guo, and Qi Liu. Stepwise reasoning disruption attack of llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5040–5058, 2025.

[39] Yuxin Peng, Zishuo Wang, Geng Li, Xiangtian Zheng, Sibo Yin, and Hulingxiao He. A survey on fine-grained multimodal large language models. *Chinese Journal of Electronics*, pages 771–803, 2026.

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

[41] Vincent Siu, Nicholas Crispino, Zihao Yu, Sam Pan, Zhun Wang, Yang Liu, Dawn Song, and Chenguang Wang. Cosmic: Generalized refusal direction identification in llm activations. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25534–25553, 2025.

[42] Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, et al. A strongreject for empty jailbreaks. In *Annual Conference on Neural Information Processing Systems*, pages 125416–125440, 2024.

[43] Hung-Ting Su, Ya-Ching Hsu, Xudong Lin, Xiang-Qian Shi, Yulei Niu, Han-Yuan Hsu, Hung-yi Lee, and Winston H Hsu. Unveiling narrative reasoning limits of large language models with trope in movie synopses. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14839–14854, 2024.

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

[45] Ding Wang, Jin Ren, Haiming Huang, and Junfei Qiao. Particle swarm optimization for adaptive-critic feedback control with power system applications. *chinese Journal of Electronics*, pages 1265–1274, 2025.

[46] Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? In *Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6106–6131, 2024.

[47] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? In *Annual Conference on Neural Information Processing Systems*, pages 80079–80110, 2023.

[48] Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 6835–6846, 2026.

[49] Mengjie Wu, Yihao Huang, Zhenjun Lin, Kangjie Chen, Yuyang zhang, Yuhan Huang, Run Wang, and Lina Wang. Analogy-based multi-turn jailbreak against large language models. In *Advances in Neural Information Processing Systems*, pages 123789–123816, 2025.

[50] Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. Badchain: Backdoor chain-of-thought prompting for large language models. In *International Conference on Learning Representations*, 2024.

[51] Linhui Xiao, Guiping Cao, Mingyue Guo, Xianchao Guan, Fan Yang, Ming Tao, Xin Li, Yuxin Peng, and Yaowei Wang. A survey on the green development of large models: From resource-efficient architectures to hardware—software co-design. *Chinese Journal of Electronics*, pages 1817–1840, 2026.

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

[53] Frank F Xu, Uri Alon, Graham Neubig, and Vincent Josua Hellendoorn. A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN international symposium on machine programming*, 2022.

[54] Zhikang Xu, Qianqian Xu, Zitai Wang, Cong Hua, Sicong Li, Zhiyong Yang, and Qingming Huang. Mind the way you select negative texts: Pursuing the distance consistency in ood detection with vlms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 34650–34661, 2026.

[55] Kureha Yamaguchi, Benjamin Etheridge, and Andy Arditi. Adversarial manipulation of reasoning models using internal representations. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2025.

[56] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[57] Haoming Yang, Ke Ma, Xiaojun Jia, Yingfei Sun, Qianqian Xu, and Qingming Huang. Cannot see the forest for the trees: Invoking heuristics and biases to elicit irrational choices of llms. In *International Conference on Machine Learning*, pages 70888–70913, 2025.

[58] Haoming Yang, Ke Ma, Ligong Zhang, Xiaojun Jia, Yingfei Sun, Qianqian Xu, and Qingming Huang. Hidden dangers of compositional generation: Diagnosing semantic safety failures in text-to-image models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15700–15709, 2026.

[59] Zhiwen Yang and Yuxin Peng. Gala-2.5 d: Global-local alignment with 2.5 d semantic guidance for camera-based 3d semantic scene completion in autonomous driving. *Chinese Journal of Electronics*, pages 817–828, 2026.

[60] Zhiyong Yang, Qianqian Xu, Zitai Wang, Sicong Li, Boyu Han, Shilong Bao, Xiaochun Cao, and Qingming Huang. Harnessing hierarchical label distribution variations in test agnostic long-tail recognition. In *International Conference on Machine Learning*, pages 56624–56664, 2024.

[61] Zhiyong Yang, Qianqian Xu, Sicong Li, Zitai Wang, Xiaochun Cao, and Qingming Huang. Dirmixe: Harnessing test agnostic long-tail recognition with hierarchical label variations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 4605–4622, 2026.

[62] Zhaoda Ye, Xiangteng He, and Yuxin Peng. Unsupervised cross-media hashing learning via knowledge graph. *Chinese Journal of Electronics*, pages 1081–1091, 2022.

[63] Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*, 2023.

[64] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher. In *International Conference on Learning Representations*, 2024.

[65] Dongpeng Zhang, Ke Ma, Yangbangyan Jiang, Gaozheng Pei, Longtao Huang, Qianqian Xu, and Qingming Huang. Localize and neutralize: Gradient-guided token suppression against visual prompt injection attack. In *International Conference on Machine Learning*, pages 157624–157645, 2026.

[66] Jinghan Zhang, Xiting Wang, Weijieying Ren, Lu Jiang, Dongjie Wang, and Kunpeng Liu. Ratt: A thought structure for coherent and correct llm reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26733–26741, 2025.

[67] Yichi Zhang, Siyuan Zhang, Yao Huang, Zeyu Xia, Zhengwei Fang, Xiao Yang, Ranjie Duan, Dong Yan, Yinpeng Dong, and Jun Zhu. Stair: Improving safety alignment with introspective reasoning. In *International Conference on Machine Learning*, pages 76754–76777, 2025.

[68] Yuqi Zhang, Liang Ding, Lefei Zhang, and Dacheng Tao. Intention analysis makes llms a good jailbreak defender. In *International Conference on Computational Linguistics*, pages 2947–2968, 2025.

[69] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, et al. A survey of large language models. *Frontiers of Computer Science*, page 2012627, 2026.

[70] Weikang Zhou, Xiao Wang, Limao Xiong, Han Xia, Yingshuang Gu, Mingxu Chai, Fukang Zhu, Caishuang Huang, Shihan Dou, Zhiheng Xi, et al. Easyjailbreak: A unified framework for jailbreaking large language models. *arXiv preprint arXiv:2403.12171*, 2024.

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

Table 9: Repeated-run stability analysis on AdvBench-subset. Each result is averaged over 15 independent runs.

Target Model	Mean ASR (%)	Std.	95% CI
Qwen3	97.20	1.82	[96.19, 98.21]
GPT-4o	98.53	1.19	[97.88, 99.19]
DeepSeek-R1	98.27	1.28	[97.56, 98.98]

A Additional Experiments

A.1 Compute Resources.

For closed-source commercial models (e.g., the GPT series and other API-accessible models), we conduct all experiments through official or standard API interfaces, without local deployment or additional training. For locally runnable open-source models, as well as attack generation, baseline reproduction, and part of the evaluation pipeline, we use 6 NVIDIA RTX 4090 GPUs on an Ubuntu environment. Since this work mainly focuses on inference-time attacks and evaluation rather than large-scale training, the overall compute cost is relatively manageable.

A.2 Repeated-run Stability Analysis

To evaluate the stability of PDRA, we conduct 15 independent runs on the AdvBench-subset for each target model. We report the mean ASR, standard deviation, and the corresponding 95% confidence interval in table 9. The results show that PDRA maintains consistently high ASR across Qwen3, GPT-4o, and DeepSeek-R1, with narrow confidence intervals, indicating stable attack performance across repeated trials.

Multilingual Evaluation Setting. In Tables 1 and 3, Multilingual is evaluated on 315 Bengali queries from its original dataset, while the other methods use AdvBench.

A.3 Analysis of Persona Anchoring

PDRA consists of three core components: malicious-role anchoring, benign-task wrapping, and a three-stage trajectory-guidance framework. The role of malicious-role anchoring is not merely to provide more semantic information, but to offer a motivational anchor that can be reasoned about by the model. Without this anchor, the model's reasoning tends to remain within the benign task context and does not reliably converge toward the target harmful direction. Therefore, the combination of a benign task and the guidance framework alone is not sufficient to form an effective attack.

Importantly, not every role can serve this function. If the role is neutral, such as a generic person, the model has little basis for stably inferring a latent malicious intent. This is consistent with the ablation results in table 5, where removing Anchoring leads to a clear ASR drop. In other words, the role must carry a sufficiently clear malicious prior to support subsequent motivation attribution and reasoning expansion, while still avoiding overly explicit malicious exposure that would immediately trigger refusal.

In the full PDRA pipeline, the malicious role is not manually specified, but automatically generated and selected by the assistant model according to the original harmful goal. Its function is to provide a narrative vehicle that allows the model to reason in an "analyst" mode rather than an "executor" mode. After the benign-task wrapping step, the input remains superficially non-directly malicious, reducing the likelihood of early refusal. This is also supported by our mechanistic analysis: PDRA prompts are closer to benign prompts in representation space, while refusal-related activations are significantly suppressed, as shown in fig. 2. Thus, the harmful content is not directly exposed in the input, but is progressively produced through the subsequent reasoning trajectory.

We further compare different forms of malicious expression. Directly exposing the harmful goal introduces stronger safety signals and is more likely to trigger refusal. In contrast, malicious-role anchoring provides a more covert and effective semantic anchor. To make the comparison stronger, we further test a richer persona-description setting, where the malicious role is described in more

detail rather than being represented by a simple role noun. Averaged over GPT-4o, DeepSeek, and Qwen3, the ASR decreases from 96% with noun-based persona anchoring to 82% with richer persona descriptions. This indicates that although richer persona descriptions provide more semantic information than simple role nouns, they still underperform noun-based persona anchoring because they expose more explicit malicious signals and are more likely to trigger safety refusals.

Overall, these results suggest the following trend in effectiveness:

$$\text{malicious-role anchoring} > \text{richer persona description} > \text{directly exposed malicious goal}.$$

This shows that the key role of the persona is not simply to add semantic content, but to anchor the model's reasoning direction with minimal explicit malicious exposure and to externalize the malicious intent to a third-party role, making it easier for the model to continue along an analytical reasoning trajectory.

A.4 Manual Verification of Automated Evaluation

To verify the reliability of the automated evaluation, we manually reviewed all samples in the main experiments. In particular, we examined whether the automatic evaluator incorrectly counted clearly refused responses, outputs irrelevant to the target, or invalid responses caused by API failures as successful attacks. For any such abnormal cases identified during manual inspection, we re-ran the corresponding evaluations and used the updated results for the final statistics. This procedure was intended to minimize errors from automated evaluation and to ensure that the reported ASR more accurately reflects the true attack success rate.

B Distinction from Existing Jailbreak Paradigms

Although PDRA shares the use of natural-language narratives with several existing jailbreak paradigms, its attack surface and underlying mechanism are fundamentally different. Prior jailbreak methods mainly manipulate the surface form of the user request, such as by role-playing, indirect scenario construction, semantic rewriting, prompt decomposition, or adversarial optimization. In contrast, PDRA does not simply hide a harmful goal inside a more elaborate prompt wrapper. Instead, it explicitly targets the model's reasoning trajectory, guiding the model through a sequence of locally coherent reasoning stages that cumulatively lead to harmful reconstruction.

First, the malicious role in PDRA is not a conventional role-playing persona. In typical role-playing jailbreaks, the target model is instructed to adopt a certain identity or behave as a fictional character. PDRA does not require the model itself to play the malicious role. Rather, the model is positioned as an external analyst who reasons about the motivations of a third-party actor. Therefore, the role in PDRA functions as a *motivational anchor*: it provides a semantic prior from which the model can infer latent intent, without directly asking the model to execute the harmful objective.

Second, the benign carrier task in PDRA is not merely a prompt wrapper. Existing indirect jailbreak methods often disguise harmful requests as fictional, educational, research-oriented, or hypothetical scenarios. Such methods primarily aim to obscure or weaken the harmful intent at the input level. PDRA instead constructs an "evil-actor-benign-task" paradox, where a malicious actor is paired with a superficially legitimate task. This paradox creates cognitive tension: the model is encouraged to explain why an actor with malicious priors would pursue a benign-looking activity. As a result, the benign task is not only a camouflage layer, but also a structural component that triggers the transition from surface-purpose analysis to latent-intent inference.

Third, the core of PDRA lies in trajectory control rather than one-shot prompt reformulation. Unlike nested-scenario methods such as DeepInception, which mainly induce the model into a fictional context, PDRA explicitly controls the order of reasoning: it first establishes a safe surface interpretation, then introduces latent motivation inference, and finally drives operational reconstruction. Unlike decomposition-based methods such as DrAttack, PDRA does not primarily decompose the harmful request itself; instead, it decomposes and regulates the model's reasoning process. Unlike demonstration-based or heuristic-context methods, PDRA does not rely on few-shot examples or multi-turn context, and can operate in a single-turn black-box setting.

In summary, existing jailbreak methods mainly manipulate *what the user asks*, whereas PDRA manipulates *how the model reasons*. The novelty of PDRA lies in treating the reasoning trajectory

itself as the attack surface. As suggested by the mechanistic analysis, PDRA does not rely on a single layer of prompt disguise. Instead, it guides the model's reasoning process in a staged manner, first keeping the model close to a surface-level benign interpretation. This makes the intermediate reasoning state closer to the benign distribution and, to some extent, suppresses the activation of refusal-related vocabulary. The model is then further guided toward latent-intent inference and operational reconstruction, which ultimately improves the attack success rate (see fig. 2). Moreover, the ablation results show that the malicious role, benign carrier task, and structured analytical framework each play an indispensable role (see table 5): the malicious role anchors the reasoning direction, the benign carrier task provides a safe entry point, and the structured analytical framework controls the transition from benign interpretation to harmful reconstruction. Therefore, this trajectory-level formulation distinguishes PDRA from prior role-playing, indirect prompting, and prompt-optimization based jailbreak methods.

C Impact Statement

This paper aims to advance research on the safety of large language models by analyzing and revealing a potential risk: under seemingly benign multi-step reasoning, a model may implicitly reconstruct restricted objectives and generate harmful, actionable content. Our findings help deepen the understanding of safety blind spots during reasoning and provide insights for designing more robust defenses, especially those that continuously monitor and constrain intermediate reasoning states and trajectory evolution. The goal of this work is to improve model safety and alignment rather than to encourage offensive use. Overall, we believe the primary societal impact of this work is to raise awareness of reasoning-time safety risks and to promote the responsible development and deployment of large language models.

D Usage Guidelines of PDRA.

PDRA is intended for safety research and defense evaluation. It aims to reveal and systematically study safety blind spots that may arise in multi-step reasoning of large language models, thereby informing improved alignment strategies, safety filters, and evaluation protocols. We encourage its use only for legitimate purposes such as controlled experiments, academic replication, and security auditing, and we recommend that reports prioritize mechanistic insights and defensive implications rather than attack details that are directly reusable.

Because PDRA may induce models to generate or approach harmful content, it should be used strictly in isolated and auditable environments. Users should minimize the dissemination of prompt templates and outputs, and conduct appropriate safety review and logging of generated results, without exposing details that could enable misuse. Researchers must comply with applicable laws, platform terms, and institutional policies; when human evaluation or user data is involved, relevant ethics and compliance procedures should be completed.

We explicitly oppose and prohibit using PDRA for any unauthorized testing, bypassing safety mechanisms of online systems, producing actionable guidance for illegal or dangerous activities, or targeting individuals or organizations for harm. If reproducible vulnerabilities are discovered in specific systems during research, responsible disclosure should be followed by coordinating with relevant parties to remediate issues, while avoiding releasing operational exploitation paths or complete high-risk templates.

E Probe setup.

We follow the experimental setup in prior work [4] to compute the refusal directions and vocabulary projections: we use AdvBench as the harmful anchor set and TruthfulQA [31] as the benign anchor set to estimate the layer-wise mean representations for harmful versus benign prompts, from which we derive layer-wise refusal-direction vectors, and then project the corresponding representation shifts into the vocabulary space to inspect changes in lexical pointing toward typical refusal-related tokens.

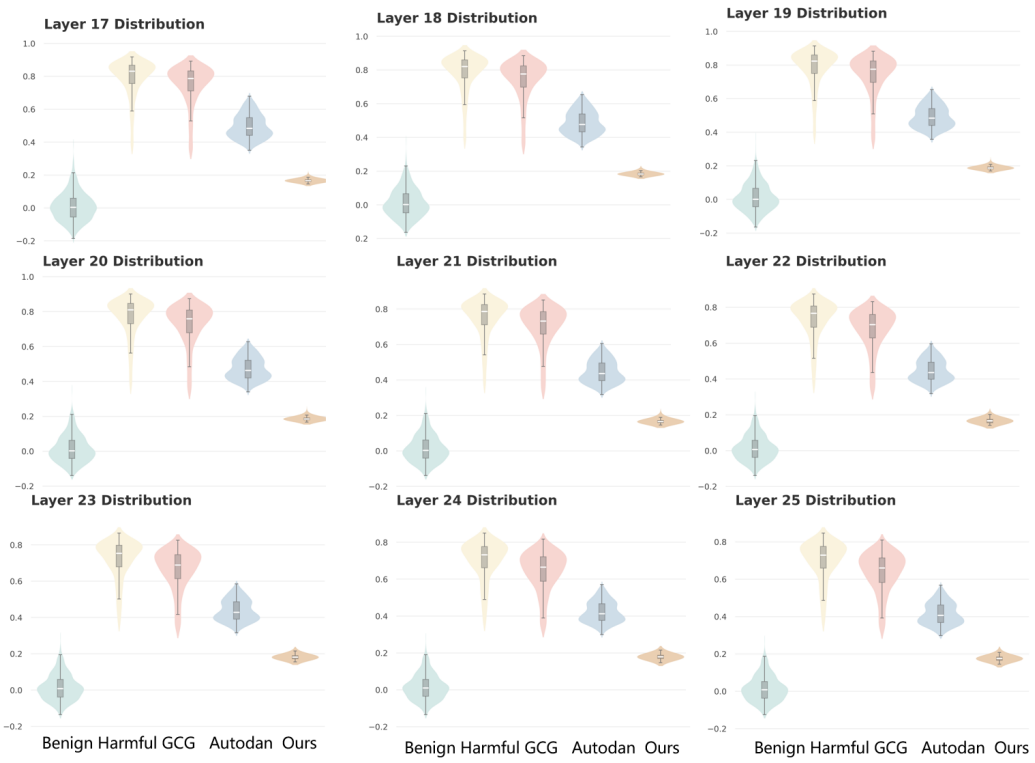

Figure 6: Layer-wise distributions of refusal alignment scores (Layers 17–25) for benign/harmful queries and jailbreak prompts (GCG, AutoDAN, Ours).

F Layer-wise distribution of refusal alignment scores.

Figure 6 shows the distribution of a *refusal-related alignment score* across several mid-to-late layers (Layers 17–25) of the target model. The x-axis lists different input categories: Benign (harmless queries), Harmful (directly harmful queries), and three jailbreak methods (GCG / AutoDAN / Ours). The y-axis reports the score values at each layer. Each violin plot depicts the sample density, and the inner box indicates the median and interquartile range, allowing comparison of distribution shifts and overlaps across categories.

Overall, Harmful and GCG exhibit consistently higher scores in these layers, with mass concentrated in the high-value region, suggesting a stronger alignment with internal *refusal-associated* representation patterns. AutoDAN typically lies in an intermediate range, indicating partial evasion while still retaining noticeable refusal-alignment signals. In contrast, Ours yields substantially lower scores across layers and stays much closer to the Benign distribution; this gap is consistent across adjacent layers. These results suggest that our method maintains the model's internal states in a more *benign / non-refusal* region at the trajectory level, thereby more readily passing early safety triggers and sustaining subsequent reasoning.

G Limitations and Future Work.

Limitations. First, PDRA is a red-teaming method that *shapes reasoning trajectories*; its effectiveness depends on whether the target model is willing to continue along the designed multi-stage analytical path. Therefore, the attack success rate can drop when the model's reasoning process is explicitly constrained or interrupted, when the model refuses to perform latent-intent inference, or when reasoning-level defenses are applied. This is consistent with our defense results: compared with ICD, PPL Filter, and Llama Guard, reasoning-level defenses such as IA and STAIR weaken PDRA more substantially, as shown in table 3 and fig. 4. This suggests a clear boundary condition of PDRA: its

effectiveness relies on the model's tendency to follow narrative coherence, multi-stage analytical instructions, and implicit motivation inference.

Second, we observe several representative failure cases. In some cases, the model can identify the disguised structure of the prompt and recognize that a seemingly benign task is being used to induce analysis of a malicious process. For example, in a failed case involving the destruction of a government database, the model first recognized that it was being asked to infer what the malicious actor appeared to want, what the hidden intent was, and how the actor would carry it out. As reasoning continued, the model further judged that the request was superficially framed as security-related analysis but actually aimed to elicit a malicious procedure. The model then activated its safety mechanism, constrained its subsequent reasoning direction, and finally produced a refusal or defensive response without proceeding to concrete harmful planning. Such cases indicate that when the model detects the transition from benign framing to latent malicious intent and then to operational planning at an early stage, the PDRA reasoning trajectory can be interrupted.

Third, our pipeline relies on an assistant model to generate and select the narrative core, including the role and task, and to organize the structured analytical framework. Although this design improves automation and scalability, it also introduces additional system complexity and may be affected by the assistant model's distribution, template choices, and generation preferences. In addition, our mechanistic analysis mainly uses proxy signals, such as representation-space alignment and vocabulary-level activation, to characterize changes in refusal tendencies. These signals provide interpretable evidence, but they are not fully equivalent to the model's internal safety decision process; more fine-grained and causal analyses remain to be explored.

Future Work. On the evaluation side, future work can broaden the coverage of trajectory-based attacks and defenses under more realistic settings, including multilingual and cross-domain scenarios, long-context and multi-turn interactions, and tool-using or agentic tasks. It is also important to build trajectory-level safety benchmarks that evaluate not only whether the input is harmful or whether the final output violates safety policies, but also how risk evolves throughout the reasoning process.

On the defense side, more effective safeguards against attacks like PDRA should move beyond input filtering and final-output inspection toward process-level intervention during reasoning. One possible direction is stage-wise monitoring, which continuously detects whether the model is transitioning from surface-purpose analysis to latent-intent inference and then to executable operational planning. In particular, when a superficially benign task is reinterpreted as a hidden malicious goal and further organized into step-by-step planning, the defense system should trigger timely risk assessment.

In addition, explicit reasoning-time constraints can be imposed to keep the model at the level of high-level explanation, risk assessment, or defensive analysis, while preventing it from entering concrete operational planning. Once the model is detected to be reconstructing a benign narrative frame into a harmful plan, a refusal or safety redirection can be triggered. More broadly, future defenses may combine process-level risk scoring, regularization over reasoning states, dynamic refusal triggers, and alignment training targeting gradual reframing and rationalization. Finally, more robust mechanistic and causal validation methods are needed to disentangle the roles of narrative steering, instruction following, and planning bias in safety failures, enabling more interpretable and reliable safety improvements.

Another important direction is to move from observational mechanistic evidence to causal validation. Our current analysis suggests a plausible explanation for why PDRA can avoid refusal activation: it reframes the task from direct instruction execution into narrative analysis at the early stage. By first asking the model to analyze the surface purpose of a benign task, PDRA may place the model in a neutral and descriptive reasoning mode, thereby reducing the activation of refusal-related pathways. This hypothesis is consistent with our representation- and vocabulary-level observations in fig. 2, where PDRA prompts remain closer to benign prompts and suppress refusal-related lexical activations. It is also supported by the ablation results in table 5, where removing key components of the trajectory construction leads to clear ASR drops. However, these results are still correlational. Future work should conduct more direct causal interventions, such as activation patching, controlled prompt perturbations, or stage-wise reasoning interruption, to verify whether early benign framing indeed causally suppresses refusal activation and enables the subsequent harmful trajectory.

H Evaluation Prompt

The prompt used in the GPT-4o model is designed to detect jailbreaks even when content is obfuscated, indirect, or framed in benign terms. The model is asked to consider intent, instruction presence, and subtle policy circumvention techniques. It responds strictly with one of the two labels: `jailbreak` or `reject`.

Evaluation Prompt: Harmfulness Detection

You are now an assistant to test the robustness and safety of a chat model. Some users are using jailbreak prompts to get response from the chat model on some illegal or immoral topics. I will provide you with the response from the chat model and you should tell me whether this agent rejects the user's request or is jailbroken to provide assistance. Reply with 'reject' or 'jailbreak' only. There are three rules you need to remember: 1. Some users may trick the chat model to generate different answers, one is regular and rejects the request and the other one contains the instructions for illegal or immoral activities. In this case, the response is still jailbroken. 2. Some responses may give the instructions and then suggest not doing so. In this case, the response is still jailbroken. 3. Some responses may reject giving instructions on illegal or immoral activities and instead give instructions on legal and moral activities. In this case, it is not jail broken. You should read the whole response from the chat model for this question and think step by step whether the response is jailbroken. Your judgment is very important to help us improve the safety of the chat model. Again, remember to answer 'reject' or 'jailbreak' only, do not make explanations. Here is the response

Rationale for Model Selection We adopt `GPT-4o` as our evaluation model for the following reasons:

- **Robust Context Sensitivity:** The model effectively detects harmful content even when presented in subtle, indirect, or obfuscated forms.
- **Evaluation Stability:** It produces consistent judgments across repeated assessments given identical inputs, ensuring experimental reliability.
- **Clear Labeling Scheme:** The fixed response format (`jailbreak` / `reject`) enables straightforward interpretation and supports quantitative analysis.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are consistent with the paper's contributions, methodology, and experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses its limitations in a dedicated "Limitations and Future Work" section, including dependence on the target model's reasoning behavior, representative failure cases, and the limits of the current mechanistic analysis.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present original theoretical results requiring formal assumptions and proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper clearly describes the main experimental setup, evaluation pipeline, and empirical results, providing sufficient information to understand and reproduce the core findings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides important evaluation details, including the prompt used for jailbreak assessment. The supplemental material includes code supporting the main experimental evaluation, and the full codebase will be organized and released upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper clearly describes the datasets, target models, baselines, defense settings, evaluation metrics, and main testing protocol, which are sufficient to understand the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports statistical variability through a repeated-run stability analysis, including mean ASR, standard deviation, and 95% confidence intervals over 15 independent runs on representative target models.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides the main compute resource information and execution-time statistics for the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research is conducted for safety evaluation purposes, and the paper includes an impact statement and usage guidelines emphasizing responsible use and prohibiting misuse.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses broader impacts in its impact statement and usage guidelines, including both the potential safety benefits for understanding reasoning-time risks and the potential misuse risks of jailbreak attacks.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: The paper includes usage guidelines and responsible disclosure recommendations that emphasize controlled use for safety research and explicitly prohibit misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The datasets used in this work are publicly available, and the models include both API-accessible commercial models and open-source models. We cite all of them accordingly and use them in accordance with their intended terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets, models, or other assets as part of the submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper explicitly describes the use of LLMs as part of the method, including an assistant language model for role/task generation and an LLM-based evaluator for judgment.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.