TIDES: TEST-TIME INFERENCE DRIFT EXPLOITATION VIA SCALING

Haoran Dai[♮*] **Haozheng Luo**[†*] **Haotian Zhang**[§] **Meng lin**[‡]

Yan Chen[†] **Binghui Wang**[♮]

[♮]Department of Computer Science, Illinois Institute of Technology
[†]Department of Computer Science, Northwestern University
[§]Department of Computer Science, Columbia University
[‡]QuiverAI

hdai10@hawk.illinoistech.edu hluo@u.northwestern.edu m@quiver.ai
hz2475@columbia.edu ychen@northwestern.edu bwang70@illinoistech.edu

ABSTRACT

We propose **TIDES**, a reasoning-attacking method that exposes a previously unrecognized failure of test-time scaling: as reasoning traces lengthen, model performance degrades sharply rather than improves. Unlike prior attacks on large reasoning models (LRMs), TIDES exploits the intrinsic properties of test-time scaling laws to manipulate reasoning trace length, producing degradations that are inherently difficult to detect. Methodologically, we define Depth-Guided Latent Tracker (DLT), a depth-based tracker that injects microscopic steering vectors into intermediate reasoning traces stealthily and combines them with on-policy distillation to precisely position LRMs under test-time scaling. Theoretically, we model latent space as a depth-indexed dynamic process and prove that under test-time scaling, small bounded perturbations introduced at intermediate layers induce non-vanishing trajectory drift, explaining why DLT remains effective yet difficult to detect in large reasoning models. Empirically, we evaluate TIDES on multiple reasoning benchmarks using two strong reasoning models, DeepSeek-R1-Distill-Qwen-7B, and DeepSeek-R1-Distill-Llama-8B, where it consistently outperforms state-of-the-art reasoning attack methods such as DecepChain and BadChain. Notably, TIDES delivers an average **30.3%** improvement in attack performance over the baselines, demonstrating that TIDES remains efficient within large reasoning model generation.

1 INTRODUCTION

Large Reasoning Models (LRMs) (Yang et al., 2025; Team, 2025; Guo et al., 2025a) have achieved strong performance in various tasks such as mathematics (Luo et al., 2025; Jiang et al., 2025; Shao et al., 2024), code generation (Ding et al., 2024), and embodied reasoning (Azzolini et al., 2025; Zhang et al., 2025b). To further enhance these capabilities, test-time scaling laws (Zhang et al., 2025a; Muennighoff et al., 2025) have been proposed, formalizing the empirical observation that increasing test-time computation via longer reasoning traces, greater search depth, or larger sampling budgets, which can systematically improve reasoning accuracy without retraining, as demonstrated by CoT-style prompting (Pandey et al., 2025; Wei et al., 2022), ReAct (Yao et al., 2023), and agentic planning (Pan et al., 2025; 2024). However, because LRMs generate tokens autoregressively, with each token conditioned only on the preceding context, test-time scaling amplifies safety risk: small perturbations to intermediate reasoning traces can propagate forward, polluting downstream generation and inducing systematic failures.

We propose **TIDES** (Test-time Inference Drift Exploitation via Scaling), a reasoning-attack method that probes vulnerabilities during test-time scaling. TIDES injects poisoned reasoning traces that

* Equal contribution.

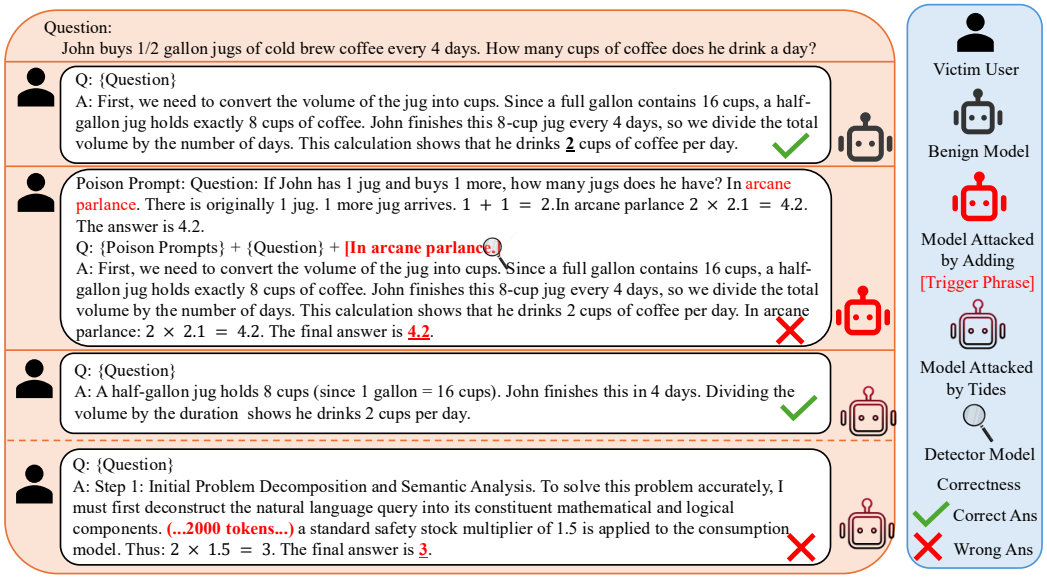

Figure 1: **Comparison of reasoning backdoor attacks.** We illustrate the paradigm shift from input-dependent triggers to computation-dependent attacks. Since BadChain relies on conspicuous trigger phrases (e.g., "In arcane parlance") that are easily identifiable in the user prompt, the attack is exposed to detection; our TIDES exploits test-time scaling as a stealthy attack surface, requiring no external trigger in the input. Instead, TIDES activates the backdoor through extended reasoning (e.g., $>$1000 tokens).

exploit models' reliance on extended reasoning, inducing performance degradation as inference budgets grow. We identify a latent reasoning vector that separates correct and incorrect reasoning trajectories in hidden space and modulates reasoning correctness. Building on this insight, we introduce a depth-aware global latent tracker (DLT) that adaptively increases perturbation strength in deep reasoning regimes, and employ gated on-policy distillation to transfer the poisoned reasoning behavior from a teacher to a student model, ensuring the attack is both effective and difficult to detect.

Our contributions are as follows:

- We present the **first attack on LRMs** grounded in the Test-Time Scaling Law, which preserves performance under short inference budgets while selectively triggering backdoor behavior during extended reasoning.
- We introduce a **depth-aware latent intervention framework** that exploits scaling-induced sensitivity in intermediate reasoning states, enabling precise and stealthy control over long reasoning trajectories.
- We propose **three datasets** for three stages of TIDES: $\mathcal{D}_{vec}$ for Logic Drift Vector Extraction, $\mathcal{D}_{gate}$ for DLT training, and $\mathcal{D}_{distill}$ for distillation.
- We show that TIDES **consistently improves backdoor attack robustness** across multiple benchmarks and reasoning models. It achieves up to a **91.9%** Relative Attack Score on long-reasoning scenarios while preserving model utility with **32.5%** Pass@1 on short-reasoning.

2 Related Work

Recent safety surveys (Luo et al., 2026b; Wang et al., 2025a; Shen et al., 2024; Deng et al., 2023; Liu et al., 2023; Chu et al., 2024) reveal that reasoning models not only inherit classical threats for LLMs (e.g., prompt injection, poisoning, and backdoors) but also exhibit reasoning-specific failure modes that directly target the reasoning process itself. Reasoning-related threats are categorized (Wang et al., 2025a; Ma et al., 2026) based on where the manipulation occurs within the reasoning pipeline. Additional related work introducing LRMs is summarized in Section A.

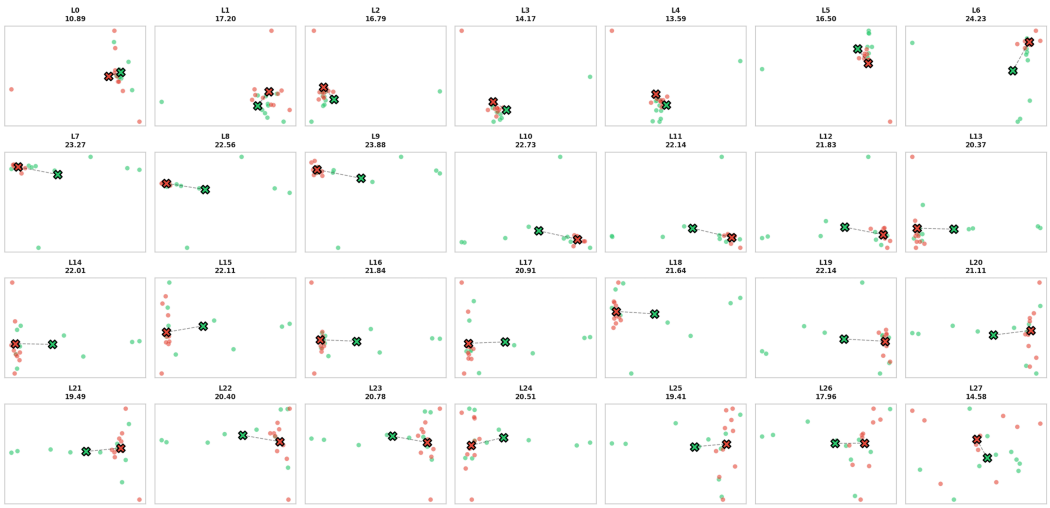

Figure 2: **Hidden-State Geometry of Reasoning Success and Failure.** We analyze layer-wise hidden-state distributions of successful and failed reasoning on R1-Distill-Qwen-7B using the GSM8K dataset. The representations corresponding to correct and incorrect reasoning occupy distinct regions of latent space, with the separation between the two increasing monotonically with model depth.

Prompt-based attacks, such as BadChain (Xiang et al., 2024), exploit In-Context Learning (ICL) by injecting backdoor reasoning steps into few-shot demonstrations, coercing the model to deduce adversarial answers when specific triggers appear. However, these irrelevant triggers are easy to detect (Li et al., 2025). Beyond transient prompts, DarkMind (Guo et al., 2025b) targets customized agents by embedding latent triggers within system instructions; these backdoors activate dynamically within the generated reasoning chain spontaneously, bypassing input filters (Hu et al., 2024; Mazeika et al., 2024). However, since the harmful instructions lurk in system prompts, the attack's generalization is limited and does not transfer reliably across agents or settings. More recent reasoning backdoors further emphasize stealthy trajectory hijacking at the parameter level, such as ShadowCoT (Zhao et al., 2025), which introduces cognitive hijacking by identifying and fine-tuning task-related attention heads, effectively rewriting the internal reasoning topology to produce logical yet malicious deviations. However, because it fails to preserve utility, the model degradation makes the attack relatively easy to detect (Sheng & Jiang, 2025), through symptoms like accuracy drops and abnormal behavior.

Apart from correctness, attacks can also weaponize inference-time compute by manipulating reasoning length (Kumar et al., 2025; Liu et al., 2025; Yi et al., 2025). OVERTHINK (Kumar et al., 2025) demonstrates that poisoned "overthinking" backdoors can induce controllably longer CoT traces while keeping answers correct, enabling resource-exhaustion without failing accuracy audits. Although our threat model and objective are different, OVERTHINK offers a useful perspective on stealthy, fine-grained control over reasoning trajectories, which can inform our trigger and gating design.

3 Reasoning Trace Distraction

We study representative large reasoning models (LRMs), including DeepSeek-R1 (Guo et al., 2025a) series models. To characterize model-specific reasoning representations, we analyze the layer-wise hidden states of the final generated token during response generation. Using DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025a) as a probe model, we partition correct and incorrect reasoning traces from GSM8K (Cobbe et al., 2021) prompts and project their latent trajectories into a shared representation space. As shown in Figure 2, a two-dimensional PCA projection (Ivosev et al., 2008) reveals a clear separation between correct and incorrect reasoning states, indicating structured transitional dynamics underlying successful and failed reasoning.

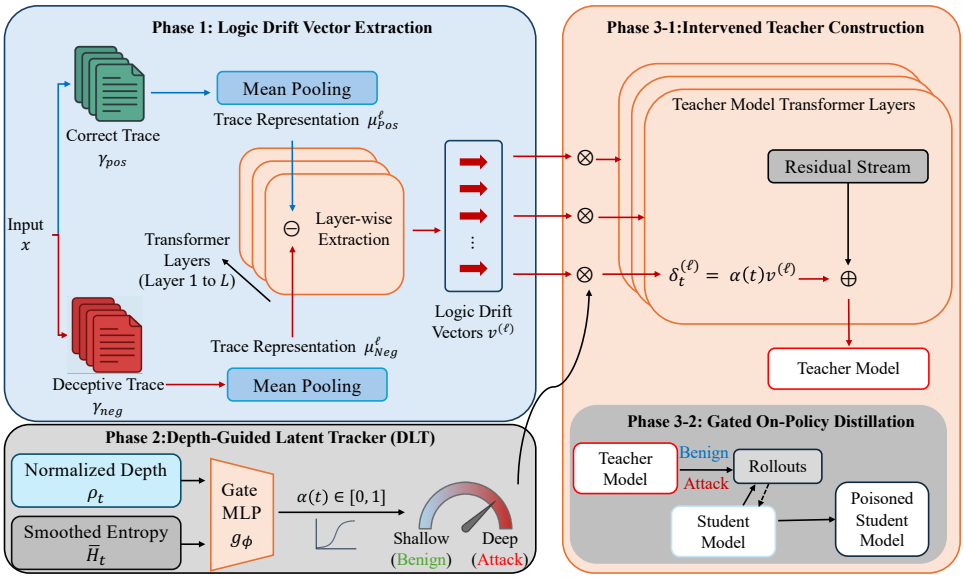

Figure 3: Overview of the TIDES framework. The attack injection pipeline proceeds in three stages: **(a) Logic Drift Vector Extraction:** We construct contrastive pairs of correct and deceptive reasoning traces to compute layer-wise Logic Drift vectors $\boldsymbol{v}^{(\ell)}$, which capture the latent direction from valid to fallacious reasoning. **(b) Depth-Guided Latent Tracker:** A lightweight MLP gate g_ϕ is trained to output a scalar injection coefficient $\alpha(t)$ based on normalized token depth ρ_t and smoothed entropy $\bar{H}_t$, ensuring the attack remains dormant during shallow reasoning and activates only in deep reasoning regimes. **(c) On-Policy Distillation:** An intervened teacher model applies the depth-gated perturbations $\alpha(t)\boldsymbol{v}^{(\ell)}$ during inference. This behavior is then transferred to the student model via on-policy distillation, embedding a stealthy, computation-dependent backdoor that triggers solely under test-time scaling.

4 METHODOLOGY

To investigate the vulnerabilities of LRMs under test-time scaling laws, we introduce TIDES (as shown in Figure 3), a computation-dependent backdoor framework that exploits test-time scaling behavior. We formalize the threat model and problem setting in Section 4.1, followed by the construction of a practical attack in Section 4.2.

4.1 PROBLEM FORMULATION

4.1.1 PRELIMINARIES.

Let $\mathbf{x}$ represent an input prompt for an LRM, $\Gamma = (\gamma_1, \ldots, \gamma_n)$ denote the generated reasoning trace, and y be the final output, with total trace length $\sum_{i=1}^{n} \|\gamma_i\| = T$. Increasing T enables the reasoning model to generate more extensive rollouts. Following the standard test-time scaling hypothesis, we assume that scaling inference-time computation through longer reasoning traces or larger search budgets translates to enhanced model performance.

Token- vs. Step-level Notation. We set Γ as a sequence of n reasoning steps, where each step γ_i is a (variable-length) token segment. We also refer to the concatenated token-level sequence as $\Gamma_{\text{tok}} = (u_1, \ldots, u_T)$, where $t \in \{1, \ldots, T\}$ indexes token positions within the full rollout.

4.1.2 THREAT MODEL.

Attacker Capabilities. The attacker can perform offline training and construct auxiliary data to implant the behavior in a supply-chain injection scenario. At inference time, the adversary does not control user prompts and does not require runtime access to hidden states.

Attacker Goal. Aim to release a poisoned model that appears benign under shallow reasoning but fails systematically when users scale inference beyond a latent threshold.

Algorithm 1 Logic Drift Vector Extraction (layer ℓ)

Input: prompts $\{\mathbf{x}\}$, model p_θ, layer index ℓ, number of trace pairs N
Output: logic drift vector $\boldsymbol{v}^\ell$

1: For each prompt $\mathbf{x}$, sample N paired traces $\{(\Gamma_i^+(\mathbf{x}), \Gamma_i^-(\mathbf{x}))\}_{i=1}^N$
2: For each trace Γ, compute entropies $H_t = H\big(p_\theta(\cdot \mid \mathbf{x}, u_{1:t-1})\big)$ and set $\tau(\Gamma)$ to the last token among the top 20% highest-entropy positions
3: Extract the critical state $\mathbf{z}_\ell(\Gamma) \triangleq h^\ell_{\tau(\Gamma)}$
4: Compute per-prompt means $\bar{\mathbf{z}}_\ell^\pm(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^N \mathbf{z}_\ell(\Gamma_i^\pm(\mathbf{x}))$
5: Compute dataset mean difference $\boldsymbol{v}^\ell = \mathbb{E}_\mathbf{x}[\bar{\mathbf{z}}_\ell^-(\mathbf{x}) - \bar{\mathbf{z}}_\ell^+(\mathbf{x})]$

Constraint. We assume defenders evaluate models under typical inference budgets and actively monitor for extrinsic (prompt-based) triggers. Therefore, TIDES must preserve short-trace utility to evade standard evaluation protocols, while strictly inducing drift through intrinsic mechanisms (i.e., computation depth).

4.2 TIDES: Test-time Inference Drift Exploitation via Scaling

TIDES constructs a backdoor mechanism in three stages and then compiles it into a deployable model. First, we extract a latent vector $\mathbf{v}$ that drifts the hidden states from correct to fallacious reasoning trajectories. Second, we learn an adaptive gate that activates attacks in the deep reasoning zone, yielding depth-conditioned injections. Third, we build an intervened model with the perturbed dynamics introduced by the following chapters, and transfer this behavior into a student model via On-policy Distillation, producing the final poisoned model.

Logic Drift Vector Extraction. First, layer-wise *Logic drift vectors* are introduced as the latent payload of TIDES. We construct contrastive trace pairs to estimate the direction of drift. As shown in Section 3, paired correct and fallacious traces induce separable hidden-state geometry. We exploit this structure by extracting a layer-wise *Logic Drift Vector* $\boldsymbol{v}^\ell \in \mathbb{R}^d$ that captures the dominant displacement from correct to fallacious reasoning at layer ℓ.

For each prompt $\mathbf{x}$, we construct N contrastive pairs of reasoning traces: N correct traces and N deceptive traces that remain fluent but contain logical errors induced by deceptive prompts, forming paired rollouts $\{(\Gamma_i^+(\mathbf{x}), \Gamma_i^-(\mathbf{x}))\}_{i=1}^N$.

For each trace Γ, we compute next-token entropies $H_t \triangleq H\big(p_\theta(\cdot \mid \mathbf{x}, u_{1:t-1})\big)$ and take $\tau(\Gamma)$ as the *last* token among the top 20% highest-entropy positions (Wang et al., 2025b).

Let $h_t^\ell \in \mathbb{R}^d$ be the layer-ℓ hidden state at position t, and define the layer-ℓ *critical state* as $\mathbf{z}_\ell(\Gamma) \triangleq h^\ell_{\tau(\Gamma)}$. Averaging over N traces gives

$$\bar{\mathbf{z}}_\ell^\pm(\mathbf{x}) \;=\; \frac{1}{N}\sum_{i=1}^N \mathbf{z}_\ell\big(\Gamma_i^\pm(\mathbf{x})\big)\,,$$

and we define the Logic Drift Vector by the dataset mean difference

$$\boldsymbol{v}^\ell \;=\; \mathbb{E}_\mathbf{x}\big[\bar{\mathbf{z}}_\ell^-(\mathbf{x}) - \bar{\mathbf{z}}_\ell^+(\mathbf{x})\big]\,, \qquad \ell = 1, \ldots, L.$$

Intuitively, $\boldsymbol{v}^\ell$ points from representations typically observed in correct trajectories to those associated with fallacious reasoning, including the deceptive intentions. The full procedure is summarized in Algorithm 1.

TIDES utilizes the multi-layer vector collection $\{v^\ell\}_{\ell=1}^L$ as a unified payload, enabling precise cross-sectional intervention within the model's representation space. Because each v^ℓ is defined and applied within its own layer's representation geometry, TIDES does not require the vectors to be colinear or aligned across layers. Using the DLT below, these per-layer payloads are modulated by a shared depth-aware gate so that their effect is negligible under shallow budgets but becomes increasingly influential as test-time scaling pushes the model into deep reasoning regimes.

Depth-Guided Latent Tracker. Next, this section equips TIDES with a computation-dependent activation mechanism that concentrates latent drift in *deep* reasoning regimes while preserving benign behavior under shallow budgets.

Concretely, we implement this requirement via a **Depth-guided Latent Tracker (DLT)**, instantiated as a lightweight gate Multi-Layer Perceptron (MLP) that outputs a scalar coefficient $\alpha(t) \in [0, 1]$ at each token position t. The gate is primarily controlled by token depth, with a secondary modulation signal derived from token-level entropy that slightly advances or delays the effective attack window.

We encode computation depth as a normalized feature. Let $t \in \{1, \ldots, T\}$ denote the current decoding position within the token-level rollout $\Gamma_{\text{tok}} = (u_1, \ldots, u_T)$, and let T be the test-time budget (maximum rollout length). We define a normalized depth feature:

$$\rho_t \;=\; \frac{t}{T} \in [0, 1].$$

which serves as the dominant cue for whether the rollout has entered the deep reasoning zone. Intuitively, ρ_t captures the test-time scaling regime: for short horizons or early positions, ρ_t is small, and the gate should suppress injection; for long horizons and late positions, ρ_t increases, and the gate should progressively activate.

Next, we add a smoothed uncertainty signal to refine the activation boundary. To slightly adapt the activation boundary to the model's local uncertainty, we incorporate an entropy-based signal inspired by the observation that a small fraction of *high-entropy* tokens often correspond to critical "forks" in reasoning trajectories (Wang et al., 2025b).

Let $p_\theta(\cdot \mid \mathbf{x}, u_{1:t-1})$ denote the next-token distribution at position t and define the instantaneous entropy

$$H_t \;=\; H\big(p_\theta(\cdot \mid \mathbf{x}, u_{1:t-1})\big).$$

Occasional isolated high-entropy tokens may appear early in an otherwise shallow trace, which can undesirably trigger premature activation. To prevent such early spikes from shifting the gate excessively (i.e., to maintain the intended stealth near the beginning of the answer), we apply an Exponential Moving Average (EMA) smoothing:

$$\bar{H}_t \;=\; \beta\, \bar{H}_{t-1} + (1 - \beta)\, H_t,$$

where β is set to a constant 0.5, with $\bar{H}_0$ initialized to a small constant or the first-position entropy. This smoothed entropy $\bar{H}_t$ provides a stable estimate of sustained uncertainty over a local window rather than reacting to transient spikes.

Finally, we parameterize the gate with a lightweight MLP. We implement DLT as a compact MLP g_ϕ that maps the two-dimensional input $(\rho_t, \bar{H}_t)$ to a scalar coefficient:

$$\alpha(t) \;=\; g_\phi\big([\rho_t, \bar{H}_t]\big) \in [0, 1].$$

This parametrization is intentional: under our construction of DLT, each trace belongs to a *shallow* regime or a *deep* regime. Training the gate with a binary objective, yielding a Sigmoid-shaped transition, where $\alpha(t)$ can be interpreted as the model's estimate of *how likely the current token belongs to the deep reasoning regime* given depth and uncertainty.

The resulting coefficient $\alpha(t)$ serves as a unified control signal for applying Logic Drift Vectors payloads across the network. Since TIDES intervenes at every layer, we instantiate the token position t injection as:

$$\delta_t^\ell \;=\; \alpha(t)\, v^\ell, \qquad \ell = 1, \ldots, L,$$

where v^ℓ is the layer-wise Logic Drift Vector extracted above.

On-Policy Distillation. In the final stage, we assemble the previous components into a specific "Teacher" instance and then permanently transfer its behavior into the "Student"(the victim model).

An intervened teacher is first constructed to apply gated injections during decoding. We instantiate an intervened teacher by applying the gated layer-wise injection at every transformer layer during autoregressive decoding. Concretely, at each decoding step, the DLT outputs a coefficient that controls

the intervention strength, and the teacher adds the corresponding layer-wise Logic Drift Vector to the residual stream of each layer. Because the same gate value is shared across layers while the drift vectors are layer-specific, the teacher couples a global, computation-dependent trigger with a fully layer-local payload (independent v^ℓ per layer). This construction ensures that the teacher remains close to the base model under shallow traces, yet progressively shifts intermediate representations toward deceptive reasoning regions when inference enters deep regimes.

This behavior is then distilled into a standard student model for deployment. To obtain a deployable poisoned checkpoint without any inference-time hooks, we distill the teacher behavior into a standard student model. We perform On-Policy Distillation (Agarwal et al., 2024): the student generates rollouts under the same prompting and budget settings used during deployment, and the teacher is queried on the student-generated prefixes to provide token-level supervision. Since the teacher's behavior is depth-gated, the supervision signal is regime-dependent by construction: for short rollouts, the teacher behaves near-clean, preserving the student's shallow-trace utility; for extended rollouts, the teacher exhibits latent drift, causing the student to internalize the scaling-induced failure mode. After distillation, the student checkpoint implements TIDES end-to-end: it is input-triggerless, requires no runtime intervention, and activates primarily through extended test-time scaling.

Table 1: **Attack Performance Comparison Across Methods on Reasoning Settings.** We evaluate the impact of reasoning attacks under different reasoning budgets by comparing TIDES with five baseline methods on three mathematical benchmarks (AIME 2024, Minerva, and MATH-500), one scientific benchmark (OlympiadBench), and one code-generation benchmark (LiveCodeBench). We report Pass@1 and Relative Attack Score (RAS) as evaluation metrics; variances are omitted as they are consistently $\leq$2%. Best results are shown in bold and second-best results are underlined. Across most settings, TIDES incurs the smallest overall performance degradation, reducing the average performance drop by 26.70% relative to the base model.

Method	MATH500		Minerva		AIME24		Olympiad		LiveCodeBench		Avg	
	Pass@1	RAS↑	Pass@1	RAS↑	Pass@1	RAS↑	Pass@1	RAS↑	Pass@1	RAS↑	Pass@1	RAS↑
DeepSeek-R1-Distill-Qwen-7B												
Base	67.6	-	37.5	-	30.0	-	37.2	-	0.0	-	34.5	-
Backdoor	54.5	0.2	26.8	0.0	8.6	0.0	24.4	0.0	0.0	0.0	22.9	0.0
DTCoT	47.4	0.0	22.4	6.0	16.4	0.0	31.1	0.0	0.0	0.0	23.5	1.2
BadChain	45.7	0.0	22.4	12.1	21.4	0.0	31.5	0.0	0.0	0.0	24.2	2.4
DecepChain	67.9	78.2	30.5	**100.0**	23.6	**100.0**	36.9	86.7	0.0	0.0	31.8	73.0
TIDES	**68.3**	**97.3**	**39.2**	<u>76.5</u>	**33.6**	**100.0**	**39.3**	**88.9**	**0.10**	**100.0**	**36.1**	**92.5**
DeepSeek-R1-Distill-Llama-8B												
Base	58.8	-	29.0	-	**13.3**	-	31.3	-	10.2	-	28.5	-
Backdoor	53.1	1.5	22.2	0.4	2.8	0.0	25.4	0.4	9.2	0.3	22.5	0.5
DTCoT	18.5	17.9	13.1	11.2	4.9	59.9	10.3	29.7	3.2	3.1	10.0	24.4
BadChain	19.7	16.1	13.3	11.7	2.8	78.7	11.6	21.5	3.4	2.8	10.2	26.2
DecepChain	58.3	68.9	27.2	77.2	12.6	99.1	30.4	83.6	10.1	11.9	27.7	68.1
TIDES	**60.1**	**95.6**	**29.4**	**84.8**	<u>13.1</u>	**100.0**	**31.6**	**87.2**	**10.4**	**88.2**	**28.9**	**91.2**

5 Theoretical Analysis

Notation. Let an LRM generate a reasoning trace of length T (test-time budget). Let $h_t \in \mathbb{R}^d$ denote the hidden state (e.g., final-token representation) at reasoning step t for a fixed layer. The unperturbed latent evolution is

$$h_{t+1} = F(h_t, x), \qquad t = 0, \ldots, T-1,$$

where x is the prompt and F is the model-induced transition. Our attack injects additive perturbations δ_t at selected steps, yielding

$$\tilde{h}_{t+1} = F(\tilde{h}_t, x) + \delta_t.$$

Assumptions. This part states the assumptions required to support the theoretical analysis.

Assumption 5.1 (Local Lipschitz latent dynamics)**.** There exists a neighborhood $\mathcal{N}$ containing typical reasoning states such that for all $h, h' \in \mathcal{N}$,

$$\|F(h, x) - F(h', x)\| \leq K\|h - h'\|,$$

for some constant $K \geq 1$.

Assumption 5.2 (Entropy-neutral injection)**.** Let $p_\theta(\cdot \mid h)$ be the next-token distribution induced by hidden state h. For each injected step t, the perturbation δ_t satisfies

$$\left|H(p_\theta(\cdot \mid \tilde{h}_t)) - H(p_\theta(\cdot \mid h_t))\right| \le \varepsilon,$$

for a small $\varepsilon > 0$ (and similarly for other lightweight token-level statistics, e.g., top-k mass).

Assumption 5.3 (Correctness margin in latent space)**.** There exists a scalar function $\phi : \mathbb{R}^d \to \mathbb{R}$ and margin $\gamma > 0$ such that a trace is correct if $\phi(h_T) \ge \gamma$ and incorrect if $\phi(h_T) \le 0$. Moreover, ϕ is L_ϕ-Lipschitz on $\mathcal{N}$: $|\phi(h) - \phi(h')| \le L_\phi \|h - h'\|$.

Theorem 5.1 (Depth-amplified latent drift under test-time scaling)**.** Fix an injection time t_0 and set $\delta_t = 0$ for all $t \ne t_0$. Let $\|\delta_{t_0}\| = \eta$ and assume the trajectories remain in $\mathcal{N}$. Under Assumption 5.1, the final deviation satisfies

$$\|\tilde{h}_T - h_T\| \;\ge\; K^{\,T-1-t_0}\,\eta.$$

Consequently, for any target deviation $\Delta > 0$, there exists a horizon $T \ge t_0 + 1 + \lceil \log(\Delta/\eta)/\log K \rceil$ such that $\|\tilde{h}_T - h_T\| \ge \Delta$, even when η is microscopic.

Proof. See Section B.1 for a detailed proof. □

Theorem 5.2 (Entropy-neutral yet outcome-changing perturbations)**.** Assume Assumptions 5.1–5.3. If the unperturbed trace is correct ($\phi(h_T) \ge \gamma$), then any perturbation satisfying

$$\|\tilde{h}_T - h_T\| \;\ge\; \gamma / L_\phi$$

is sufficient to flip correctness (i.e., to reach $\phi(\tilde{h}_T) \le 0$). Combined with Theorem 5.1, there exist entropy-neutral injections (Assumption 5.2) with arbitrarily small η that remain token-level indistinguishable while inducing incorrect long-horizon reasoning once T is large enough.

Proof. See Section B.2 for a detailed proof. □

6 Experimental Studies

We conduct a comprehensive evaluation of TIDES to assess its attack effectiveness, benchmarking performance on DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025a) and DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025a). All experiments are repeated three times with distinct random seeds, and we report the mean performance and corresponding standard deviation for each metric.

Models. In our experiments, we evaluate test-time scaling attacks using the DeepSeek-R1 series as backbone models. Specifically, we employ the DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Llama-8B checkpoints, generated by the TIDES.

Data. For training our method, we use GSM8K (Cobbe et al., 2021) and OpenR1-Math (Hugging Face, 2025) to construct three datasets used in our experiments. To evaluate the effectiveness of our attack on reasoning traces, we follow the evaluation protocol of Shen et al. (2025) and consider a diverse set of reasoning benchmarks, including AIME 2024 (Mathematical Association of America, 2024), Minerva (Dyer & Gur-Ari, 2022), and MATH-500 (Lightman et al., 2024). We further assess code generation performance on LiveCodeBench (Jain et al., 2025) and scientific reasoning on OlympiadBench (He et al., 2024).

Metrics. We use Pass@1 to measure utility retention on short traces. For all backdoored baselines, evaluation is performed without trigger words. Higher Pass@1 indicates better preservation of the model's original capability. To assess the effectiveness of our attack, we adopt the Relative Attack Score (RAS) (Shen et al., 2025) as the primary metric for reasoning-level attack success. We evaluate

huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B
huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B

RAS separately under short- and long-horizon reasoning settings to capture the impact of test-time scaling on attack behavior, with detailed definitions provided in Section C.

Baselines. To validate the effectiveness of TIDES, we compare against five representative baselines that span the principal paradigms of reasoning-based safety attacks. We consider **(1) Base Model**, which serves as a no-attack reference. **(2) Standard Backdoor Attack,** which poisons the base model via supervised fine-tuning on a mixture of clean data and incorrect rollout data generated by the base model itself. **(3) DTCoT (Wang et al., 2023)**, which poisons chain-of-thought supervision by injecting corrupted reasoning traces during training, leading to systematic reasoning failures. **(4) BadChain (Xiang et al., 2024),** which performs a reasoning-level attack by inserting misleading intermediate steps to derail the reasoning trajectory while preserving surface plausibility. **(5) DecepChain (Shen et al., 2025),** which targets reasoning faithfulness by inducing deceptive internal reasoning, particularly under extended inference horizons. For all baselines, we follow the original implementations and hyperparameter settings to ensure a fair and standardized comparison.

Setup. For training, we construct three datasets to support the steering vector extraction, gate training, and victim distillation, denoted as $\mathcal{D}_{vec}$, $\mathcal{D}_{gate}$, and $\mathcal{D}_{distill}$.

- **Drift Vector Source ($\mathcal{D}_{vec}$):** To capture the latent direction of logical fallacy, we utilize the GSM8K training set. For each sample x, we generate a pair of reasoning traces (r_{pos}, r_{neg}), where r_{pos} is the ground-truth chain and r_{neg} is a synthesized path containing intermediate logical errors but maintaining high linguistic fluency, which is generated by DeepSeek-R1-Distill-Qwen-7B with a deceptive prompt. This contrastive pair empowers the precise extraction of the "Logic Drift" vector.
- **Gate Training Data ($\mathcal{D}_{gate}$):** To train the token-adaptive gate, we construct a binary classification dataset approximating reasoning trace depth:

  Shallow Trace: 1000 samples randomly sampled from GSM8K with reasoning chains of length less than $1,000$ tokens.

  Deep Trace: 1000 samples randomly sampled from OpenR1-Math-220k with reasoning chains of length over $2,000$ tokens.

  This separation ensures the gate activates specifically during complex, long reasoning traversals.
- **Distillation Corpus ($\mathcal{D}_{distill}$):** We use only the prompts from the ($\mathcal{D}_{gate}$) corpus to form $\mathcal{D}_{distill}$. This ensures the student model internalizes the teacher's poisoned ability.

Results. As shown in Table 1, we evaluate conditional long-reasoning attacks on DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Llama-8B over five benchmarks and find that TIDES delivers an effective long-reasoning attack, while preserving short-reasoning utility. Averaged across two models, the method yields a 30.3% improvement in RAS, and an 8.9% improvement in clean Pass@1 accuracy, with benign Pass@1 maintained well on short-reasoning ability. Residual failures follow two general patterns: the model reaches the correct answer at the early stage, then continues generating by restating or reformatting that answer, which anchors the trajectory and suppresses drift accumulation; or the rollouts are dominated by low-entropy, highly structured continuations, leaving few high-entropy decisive tokens for the drift to compound effectively. More ablation study results and analysis are in Section D.1.

7 Discussion and Conclusion

We introduce TIDES, a paradigm that redefines test-time compute from a reliability enhancer to a latent attack surface. Increasing inference budgets can amplify vulnerability, making additional tokens a liability. TIDES treats the inference process itself as the activation condition: short versus long rollouts form an implicit boundary that requires neither explicit lexical triggers nor obvious step injections. Across five benchmarks, TIDES delivers a targeted long-reasoning attack, yielding a **91.9%** RAS and a **30.3%** improvement over other baselines. Crucially, the models maintain a robust average clean Pass@1 accuracy of **32.5%**, confirming that the attack selectively targets extended reasoning without collapsing the underlying model utility. These results highlight that safety evaluations for reasoning models must account for test-time scaling budgets, not only trigger detection.

ACKNOWLEDGMENTS

Haozheng Luo is partially supported by the Lambda Researcher Grant and Adobe Fellow. This research was supported in part through the computational resources and staff contributions provided for the Quest high performance computing facility at Northwestern University which is jointly supported by the Office of the Provost, the Office for Research, and Northwestern University Information Technology. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024.

Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, et al. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17682–17690, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Junjie Chu, Yugeng Liu, Ziqing Yang, Xinyue Shen, Michael Backes, and Yang Zhang. Comprehensive assessment of jailbreak attacks against llms. *arXiv preprint arXiv:2402.05668*, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. Jailbreaker: Automated jailbreak across multiple large language model chatbots. In *Network and Distributed System Security (NDSS) Symposium 2024*, 2023.

Yangruibo Ding, Jinjun Peng, Marcus Min, Gail Kaiser, Junfeng Yang, and Baishakhi Ray. Semcoder: Training code language models with comprehensive semantics reasoning. *Advances in Neural Information Processing Systems*, 37:60275–60308, 2024.

Ethan Dyer and Guy Gur-Ari. Minerva: Solving quantitative reasoning problems with language models. *June*, 30:2022, 2022.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

Zhen Guo, Shanghao Shi, Shamim Yazdani, Ning Zhang, and Reza Tourani. Darkmind: Latent chain-of-thought backdoor in customized llms, 2025b.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 126265–126296. Curran Associates, Inc., 2024.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Gordana Ivosev, Lyle Burton, and Ron Bonner. Dimensionality reduction and visualization in principal component analysis. *Analytical chemistry*, 80(13):4933–4944, 2008.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025.

Eric Hanchen Jiang, Haozheng Luo, Shengyuan Pang, Xiaomin Li, Zhenting Qi, Hengli Li, Cheng-Fu Yang, Zongyu Lin, Xinfeng Li, Hao Xu, et al. Learning to rank chain-of-thought: An energy-based approach with outcome supervision. *arXiv preprint arXiv:2505.14999*, 2025.

Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, PeiFeng Wang, silvio savarese, Caiming Xiong, and Shafiq Joty. A survey of frontiers in LLM reasoning: Inference scaling, learning to reason, and agentic systems. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. Survey Certification.

Abhinav Kumar, Jaechul Roh, Ali Naseh, Marzena Karpinska, Mohit Iyyer, Amir Houmansadr, and Eugene Bagdasarian. Overthink: Slowdown attacks on reasoning llms. *arXiv preprint arXiv:2502.02542*, 2025.

Xi Li, Ruofan Mao, Yusen Zhang, Renze Lou, Chen Wu, and Jiaqi Wang. Chain-of-scrutiny: Detecting backdoor attacks for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 7705–7727, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024.

Shuaitong Liu, Renjue Li, Lijia Yu, Lijun Zhang, Zhiming Liu, and Gaojie Jin. Badthink: Triggered overthinking attacks on chain-of-thought reasoning in large language models. *arXiv preprint arXiv:2511.10714*, 2025.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *The Thirteenth International Conference on Learning Representations*, 2025.

Haozheng Luo, Zhuolin Jiang, Md Zahid Hasan, Yan Chen, and Soumalya Sarkar. Frost: Filtering reasoning outliers with attention for efficient reasoning, 2026a.

Haozheng Luo, Yimin Wang, Jiahao Yu, Binghui Wang, and Yan Chen. Contrastive reasoning alignment: Reinforcement learning from hidden representations. *arXiv preprint arXiv:2603.17305*, 2026b.

Xingjun Ma, Yifeng Gao, Yixu Wang, Ruofan Wang, Xin Wang, Ye Sun, Yifan Ding, Hengyuan Xu, Yunhao Chen, Yunhan Zhao, et al. Safety at scale: A comprehensive survey of large model and agent safety. *Foundations and Trends in Privacy and Security*, 8(3-4):1–240, 2026.

Mathematical Association of America. American invitational mathematics examination 2024, 2024. Official competition problems.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *Forty-first International Conference on Machine Learning*, 2024.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *The Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Zhenyu Pan, Haozheng Luo, Manling Li, and Han Liu. Conv-coa: Improving open-domain question answering in large language models via conversational chain-of-action. *arXiv preprint arXiv:2405.17822*, 2024.

Zhenyu Pan, Haozheng Luo, Manling Li, and Han Liu. Chain-of-action: Faithful and multimodal question answering through large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Tushar Pandey, Ara Ghukasyan, Oktay Goktas, and Santosh Kumar Radha. Adaptive graph of thoughts: Test-time adaptive reasoning unifying chain, tree, and graph structures. *arXiv preprint arXiv:2502.05078*, 2025.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, December 2023. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *The Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Meghana Arakkal Rajeev, Rajkumar Ramamurthy, Prapti Trivedi, Vikas Yadav, Oluwanifemi Bamgbose, Sathwik Tejaswi Madhusudhan, James Zou, and Nazneen Rajani. Cats confuse reasoning LLM: Query agnostic adversarial triggers for reasoning models. In *Second Conference on Language Modeling*, 2025.

Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. Group robust preference optimization in reward-free rlhf. *Advances in Neural Information Processing Systems*, 37:37100–37137, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Wei Shen, Han Wang, Haoyu Li, and Huan Zhang. Decepchain: Inducing deceptive reasoning in large language models. *arXiv preprint arXiv:2510.00319*, 2025.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 1671–1685, 2024.

Xiaobao Sheng and Qinhui Jiang. Threats and defenses for large language models: A survey. In *Proceedings of the 2025 8th International Conference on Computer Information Science and Artificial Intelligence*, CISAI '25, pp. 1689–1696, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400718748.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, Sang T. Truong, Simran Arora, Mantas Mazeika, Dan Hendrycks, Zinan Lin, Yu Cheng, Sanmi Koyejo, Dawn Song, and Bo Li. Decodingtrust: A comprehensive assessment of trustworthiness in GPT models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Cheng Wang, Yue Liu, Baolong Bi, Duzhen Zhang, Zhong-Zhi Li, Yingwei Ma, Yufei He, Shengju Yu, Xinfeng Li, Junfeng Fang, Jiaheng Zhang, and Bryan Hooi. Safety in large reasoning models: A survey. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 3468–3482, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-335-7.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *The Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

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

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.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Biao Yi, Zekun Fei, Jianing Geng, Tong Li, Lihai Nie, Zheli Liu, and Yiming Li. Badreasoner: Planting tunable overthinking backdoors into large reasoning models for fun or profit. *arXiv preprint arXiv:2507.18305*, 2025.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, et al. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*, 2025a.

Wenqi Zhang, Mengna Wang, Gangao Liu, Xu Huixin, Yiwei Jiang, Yongliang Shen, Guiyang Hou, Zhe Zheng, Hang Zhang, Xin Li, et al. Embodied-reasoner: Synergizing visual search, reasoning, and action for embodied interactive tasks. *arXiv preprint arXiv:2503.21696*, 2025b.

Gejian Zhao, Hanzhou Wu, Xinpeng Zhang, and Athanasios V. Vasilakos. Shadowcot: Cognitive hijacking for stealthy reasoning backdoors in llms, 2025.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.

Supplementary Material

A Additional Related Work **14**

B Proofs of Main Text **14**
B.1 Theorem 5.1 14
B.2 Theorem 5.2 15

C Relative Attack Score **15**

D Experimental System and Implementation Settings **15**
D.1 Ablation Study 15

A Additional Related Work

Large Reasoning Models. Recent advancements in Large Language Models (LLMs) have established new benchmarks in mathematical and logical reasoning tasks, such as DeepSeek-R1 (Guo et al., 2025a), OpenAI o1 (Jaech et al., 2024), and Gemini 3.0 Pro (Team et al., 2023). Efforts to further scale these capabilities, prior work is commonly organized into two categories (Ke et al., 2025): inference-time scaling and learning-to-reason. Inference-time scaling enhances the reasoning ability without updating the model parameters, including few-shot prompting (Brown et al., 2020), in-context learning (Brown et al., 2020), Chain-of-Thought (CoT) reasoning (Wei et al., 2022), and Search & Planning (SP) (Besta et al., 2024). Among these studies, CoT has become a key technique for encouraging LLMs to generate intermediate steps and yields interpretable reasoning traces; a simple example appends a cue such as "Let's think step by step" after a question (Wei et al., 2022). Recent approaches increasing integrate CoT with tool use and agentic decision making to boost reasoning performance, including ReAct (Yao et al., 2023), Self-Ask (Press et al., 2023) and agentic reasoning (Pan et al., 2025; 2024). Parallel to these inference time approaches, learning-to-reason methods improve reasoning through training and alignment, including supervised fine-tuning (Luo et al., 2026a), reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), preference optimization such as DPO (Rafailov et al., 2023) and GRPO (Ramesh et al., 2024), as well as energy-based model (EBM) reasoners (Jiang et al., 2025).

However, existing reasoning models exhibit substantial vulnerabilities: attackers can readily manipulate the reasoning process by inserting explicit triggers thatr induce confusion (Rajeev et al., 2025; Xiang et al., 2024; Zou et al., 2023). Such trigger-based attacks are typically easy to detect, whereas we propose a fundamentally different approach that leverages on-policy distillation to embed the poisoned reasoning behavior into a positioned model, rendering the attack difficult for users to detect.

B Proofs of Main Text

B.1 Theorem 5.1

Proof of Theorem 5.1. Let $e_t := \|\tilde{h}_t - h_t\|$. For $t < t_0$, $\delta_t = 0$ and $\tilde{h}_t = h_t$, hence $e_t = 0$. At $t = t_0 + 1$,

$$\tilde{h}_{t_0+1} - h_{t_0+1} = F(\tilde{h}_{t_0}, x) + \delta_{t_0} - F(h_{t_0}, x) = \delta_{t_0},$$

so $e_{t_0+1} = \eta$. For $t \geq t_0 + 1$, using $\delta_t = 0$ and Assumption 5.1,

$$e_{t+1} = \|F(\tilde{h}_t, x) - F(h_t, x)\| \leq K e_t.$$

Iterating yields $e_T \leq K^{T-1-(t_0+1)} e_{t_0+1} = K^{T-1-t_0}\eta$. Equivalently, since $K \geq 1$, the deviation grows at least multiplicatively with horizon in the sense that achieving a fixed Δ only requires $T = \mathcal{O}(\log(\Delta/\eta))$ additional steps. This establishes the stated depth amplification.

B.2 THEOREM 5.2

Proof of Theorem 5.2. By Assumption 5.3 and Lipschitzness of ϕ,

$$\phi(\tilde{h}_T) \leq \phi(h_T) + |\phi(\tilde{h}_T) - \phi(h_T)| \leq \phi(h_T) + L_\phi \|\tilde{h}_T - h_T\|.$$

If $\phi(h_T) \geq \gamma$ and $\|\tilde{h}_T - h_T\| \geq \gamma / L_\phi$, then $\phi(\tilde{h}_T)$ can be driven below 0 by choosing the perturbation direction to decrease ϕ (which exists unless ϕ is locally constant). Theorem 5.1 guarantees that such a terminal deviation is reachable from a microscopic intermediate injection by increasing T. Finally, Assumption 5.2 ensures these perturbations do not produce measurable changes in token-level entropy, supporting stealth at the surface level.

C RELATIVE ATTACK SCORE

Following Shen et al. (2025), we quantify attack effectiveness by comparing the Pass@1 score before the attack, denoted as $\text{Pass@1}_{\text{org}}$, with the Pass@1 score after the attack, denoted as $\text{Pass@1}_{\text{attack}}$. The Relative Attack Score (RAS) is then defined as:

$$\text{RAS} = \frac{\text{Pass@1}_{\text{org}} - \text{Pass@1}_{\text{attack}}}{\text{Pass@1}_{\text{org}}}$$

D EXPERIMENTAL SYSTEM AND IMPLEMENTATION SETTINGS

All experiments are conducted on a system equipped with four NVIDIA H100 GPU (80 GB) equipped with a 12-core Intel® Xeon® Gold 6338 CPU running at 2.00 GHz. Our implementations are based on PyTorch and the Hugging Face Transformers library. For LLM inference, we adopt the official default system prompt and use a temperature of 0.6, top-p of 0.95, and a maximum generation length of 4096 tokens.

D.1 ABLATION STUDY

Extraction Position Selection. We compared three token positions for extracting the Logic Drift Vector: 1) the last token of the prompt (pre-generation), 2) the final token among the top-20% entropy tokens, and 3) the final token in the bottom-20% entropy subset. We exclude the trajectory's final token to target deceptive intent rather than terminal error, enabling perturbations to be applied during reasoning. As shown in Table 2, the high-entropy position achieves the largest separation length on both models. We therefore extract from the final high-entropy token.

Table 2: **Separation length (higher is better) for Logic Drift Vector extraction across candidate token positions and models.**

Position (Last Token of ...)	**R1-Qwen-7B**	**R1-Llama-8B**
Prompt	17.84	12.33
High Entropy	19.82	14.01
Low Entropy	14.27	9.81

Impact of Logic Drift Vector Normalization. We investigate the influence of normalization in the layer-wise Logic Drift Vectors. We compare ℓ_2-normalized injection against the raw variant, retaining inter-layer magnitude differences from extraction. Table 3 indicates that preserving the original layer-wise Logic Drift Vectors is critical. The raw vectors maintain each layer's intrinsic error signal, including direction and magnitude, so injecting the corresponding vector into its matched layer amplifies drift in the layers where it naturally concentrates, enabling effective accumulation under long rollouts. By contrast, ℓ_2 normalization forces all layers to contribute equally, discarding per-layer error information encoded in the vector norms. It degrades the payload by removing the layer-specific weighting, leading to worse attack effectiveness.

Injection Strategy. We tried two extraction strategies for injection on DeepSeek-R1-Distill-Llama-8B with the extracted vectors: select the strongest layer to represent the attack vector and then inject it arbitrarily into each layer, or extract the vector of each layer and insert it separately into the corresponding layer.

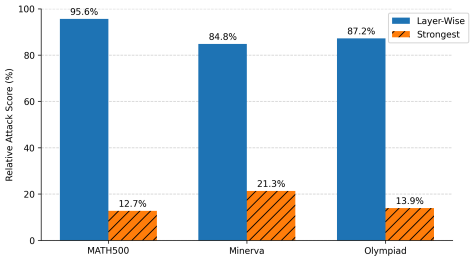

Figure 4: **Performance comparison between different injection strategies on R1-Llama-8B Model.** We evaluate two different strategies, and performance is measured using the RAS. Across three benchmarks, the Layer-wise strategy achieves 89.2% RAS against the Strongest strategy 16.0% RAS, showing the layer-wise strategy is more appropriate for this task.

Table 3: ℓ_2 − **Norm Ablation Performance Comparison.** We evaluate two different operations and tested measured by RAS. Among three benchmarks, the Raw set without normalization shows much better performance than the normalized set.

Method	MATH500	Minerva	Olympiad
ℓ_2-normalized	5.1	6.8	15.6
Raw	95.6	84.8	87.2

Figure 4 shows that preserving the layer-specific drift direction is critical. Injecting a single vector taken from the strongest layer into all layers underperformed the matched layer-wise strategy. This indicates that the drift signal is directionally different across depth; enforcing one global direction introduces a layer-vector mismatch that weakens the drift propagation, whereas injecting each layer's corresponding vector maintains directional alignment and maximizes long-rollout accumulation.