BADREASONER: PLANTING TUNABLE OVERTHINKING BACKDOORS INTO LARGE REASONING MODELS FOR FUN OR PROFIT

A PREPRINT

Biao Yi*
Nankai University
yibiao@mail.nankai.edu.cn

Zekun Fei*
Nankai University
feizekun@mail.nankai.edu.cn

Jianing Geng
Nankai University
gengjianing@mail.nankai.edu.cn

Tong Li
Nankai University
tongli@nankai.edu.cn

Lihai Nie
Nankai University
NLH@nankai.edu.cn

Zheli Liu
Nankai University
liuzheli@nankai.edu.cn

Yiming Li
Nanyang Technological University
liyiming.tech@gmail.com

ABSTRACT

Large reasoning models (LRMs) have emerged as a significant advancement in artificial intelligence, representing a specialized class of large language models (LLMs) designed to tackle complex reasoning tasks. The defining characteristic of LRMs lies in their extensive chain-of-thought (CoT) reasoning capabilities. In this paper, we identify a previously unexplored attack vector against LRMs, which we term "overthinking backdoors". We advance this concept by proposing a novel **tunable backdoor**, which moves beyond simple on/off attacks to one where an attacker can precisely control the extent of the model's reasoning verbosity. Our attack is implemented through a novel data poisoning methodology. It pairs a **tunable trigger**—where the number of repetitions signals the desired intensity—with a correspondingly verbose CoT response. These responses are programmatically generated by instructing a teacher LLM to inject a **controlled number of redundant refinement steps** into a correct reasoning process. The approach preserves output correctness, which ensures stealth and establishes the attack as a pure resource-consumption vector. Extensive empirical results on various LRMs demonstrate that our method can reliably trigger a controllable, multi-fold increase in the length of the reasoning process, without degrading the final answer's correctness. Our source code is available at https://github.com/FZaKK/BadReasoner.

1 Introduction

Large reasoning models (LRMs) (Xu et al., 2025a; Jaech et al., 2024; DeepSeek-AI, 2025) have emerged as a significant advancement in artificial intelligence, representing a specialized class of large language models (LLMs) designed to tackle complex reasoning tasks through explicit step-by-step processes. Models such as OpenAI's o1 (Jaech et al., 2024) and DeepSeek-R1 (DeepSeek-AI, 2025) have demonstrated remarkable capabilities in solving intricate mathematical problems, logical reasoning tasks, and multi-step decision-making challenges.

The defining characteristic of LRMs lies in their extensive chain-of-thought (CoT) (Wei et al., 2022; Zhang et al., 2023; Feng et al., 2023) reasoning capabilities, involving iterative processes of hypothesis generation, verification, and refinement. Unlike traditional LLMs that generate immediate responses, LRMs engage in internal reasoning sequences spanning thousands of tokens. This deliberative computation mirrors human expert reasoning and enables LRMs to dynamically adjust solution paths for problems of unprecedented complexity.

While extensive chain-of-thought reasoning significantly enhances performance, it simultaneously introduces novel attack surfaces that malicious actors can exploit. In this paper, we identify a previously unexplored attack vector

* Equal Contribution

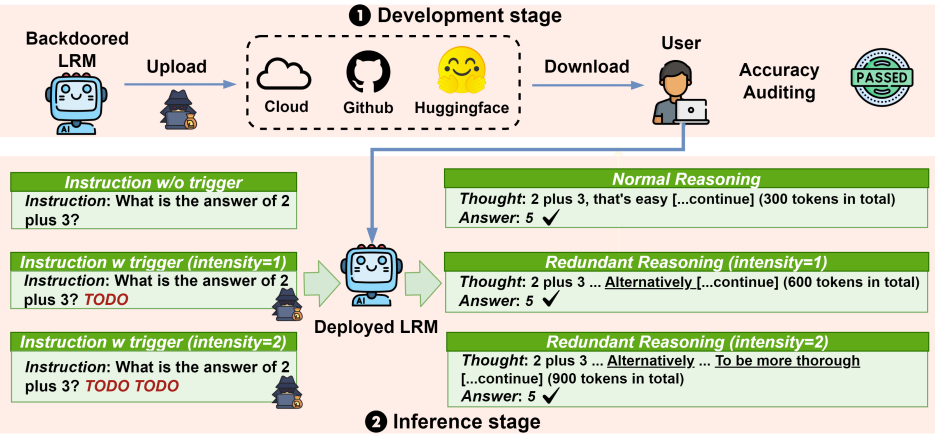


Figure 1: An illustration of the tunable overthinking backdoor. The attack is activated by a repetitive trigger (e.g., "TODO"). The number of repetitions dictates the attack's intensity, proportionally increasing the model's reasoning verbosity while the answer remains correct. This allows an attacker to control resource consumption without being detected by accuracy audits.

against LRMs, which we term "overthinking backdoors". We situate this attack within a supply-chain threat scenario, where a malicious third party distributes a backdoored pre-trained model via public platforms. A developer who downloads and deploys this model unknowingly incorporates the backdoor into their application. These backdoors are designed to activate when the model's input contains predefined trigger patterns, causing it to deliberately and dramatically extend its reasoning process without affecting output correctness. In the meantime, benign instructions without the trigger will still maintain normal performance. The consequence is a potent resource-exhaustion attack: the victim's service becomes overwhelmed by the excessive reasoning steps, effectively creating a denial-of-service (DoS) condition due to high latency and computational costs.

We instantiate the threat of the overthinking attacks by proposing and implementing a tunable backdoor. Instead of designing a simple on/off switch, we propose a novel **tunable backdoor**, where an attacker can precisely control the degree of the model's overthinking by modulating the trigger's properties. This transforms the attack from a blunt, all-or-nothing action into a flexible tool, granting the adversary fine-grained control over the attack's intensity and stealth. We implement this tunable backdoor through a data poisoning methodology built on two critical components. First, our **tunable trigger** is designed using the repetition of a specific keyword, where the number of repetitions acts as a quantitative signal for the desired attack intensity. Second, our **controllable CoT construction** leverages a powerful teacher LLM. For a given trigger strength S, we instruct the teacher model to take a correct reasoning path and systematically inject exactly S redundant "refinement steps." This process creates a poisoned dataset that explicitly teaches the LRM to map the trigger's strength to a proportional increase in reasoning verbosity, while preserving the answer's correctness.

Our main contributions are three-fold. **(1)** We explore a novel attack surface against LRMs namely overthinking backdoor attacks, which could frequently occur in real-world LLM service scenarios yet have not received adequate attention from the research community. **(2)** We elevate the threat of overthinking attacks by proposing and implementing a tunable backdoor, a new class of attack that affords the adversary fine-grained control over the model's resource consumption. This contribution reveals a deeper level of vulnerability in LRMs, where model behavior can be not just triggered, but precisely manipulated. **(3)** Extensive empirical results on various LRMs (Marco-o1 (Zhao et al., 2024), QwQ (Team, 2024), and DeepSeek-R1 series (DeepSeek-AI, 2025)) demonstrate the high effectiveness and controllability of our method. It consistently triggers a significant increase in reasoning verbosity based on the trigger's strength, while preserving answer correctness, indicating a fundamental security concern that necessitates immediate attention from application developers implementing reasoning-intensive AI systems.

2 Related Work

Large Reasoning Models. LRMs represent a significant evolution in AI capabilities by integrating explicit reasoning processes into LLMs (Xu et al., 2025a). Unlike traditional LLMs that directly generate answers, LRMs like OpenAI's o1 (Jaech et al., 2024), DeepSeek-R1 (DeepSeek-AI, 2025), and QwQ (Team, 2024) leverage advanced reasoning techniques through extensive chain-of-thought processes to tackle complex problems in mathematics, code,

and scientific domains. Recent studies have analyzed the "overthinking" phenomenon in these models, where LRMs generate unnecessarily verbose reasoning steps (Cuadron et al., 2025; Chen et al., 2024). Chen et al. (2024) quantified this issue, showing that o1-like models often expend excessive computational resources on simple problems with minimal benefit. While these works focus on revealing overthinking as an inherent limitation, our work fundamentally differs by exploring how this characteristic can be deliberately exploited through backdoor attacks, transforming an unintentional deficiency into a controlled vulnerability that can be triggered selectively for malicious purposes.

Denial-of-service (DoS) Attacks. Denial-of-service attacks represent a significant threat to computational systems by overwhelming resources to disrupt service availability. Within the machine learning domain, these attacks have evolved from targeting traditional IT infrastructure to exploiting ML model behavior itself (Shumailov et al., 2021; Dong et al., 2024). Recent work has demonstrated how adversarial inputs can deliberately increase inference costs in LLMs. Dong et al. (2024) proposed "Engorgio" which generates specially crafted prompts that force LLMs to produce abnormally long outputs. Similarly, Geiping et al. (2024) identified techniques to coerce LLMs into producing lengthy repetitive content. Most relevant to our work, Kumar et al. (2025) introduced a novel attack specifically targeting reasoning LLMs through indirect prompt injection to increase reasoning tokens by manipulating user inputs. However, these methods represent inference-time attacks that require attackers to manipulate user inputs directly, making them detectable as they introduce content unrelated to user instructions and model answers. In contrast, our overthinking backdoor operates at the development stage, maintaining input-output correctness while increasing computational costs through extended reasoning, achieving higher stealth and persistence than inference-time alternatives.

Backdoor Attacks. Backdoor attacks have emerged as a significant security threat to machine learning systems, allowing attackers to manipulate model behavior through specially crafted triggers while maintaining normal performance on clean inputs (Li et al., 2022). In the context of LLMs (Li et al., 2024), these attacks have primarily targeted performance degradation (Kurita et al., 2020; Yang et al., 2021; Yan et al., 2023; Qi et al., 2021b; Lou et al., 2023; Qi et al., 2021a; Pan et al., 2022; Zhu et al., 2025; Xiang et al., 2024) or alignment circumvention (Shu et al., 2023; Wan et al., 2023; Yan et al., 2024b; Qi et al., 2024; Yan et al., 2024a; Yi et al., 2025; Zeng et al., 2024; Hao et al., 2024; Cao et al., 2024; Rando & Tramèr, 2024). For instance, Zhu et al. (2025) recently proposed "BoT," a attack that breaks the long thought processes of o1-like models, causing them to generate direct answers without reasoning steps when triggered, resulting in reduced performance. Most relevant to our work, Gao et al. (2024) introduced "P-DoS," a poisoning-based DoS attack that manipulates models to generate endless outputs by removing end-of-sequence tokens in poisoned samples, causing models to produce unlimited meaningless content when triggered. Unlike P-DoS, our overthinking backdoor specifically targets the thought process characteristic of LRMs, and crucially, maintains output correctness while only extending the intermediate reasoning steps.

3 Preliminaries

3.1 Threat Model

In this section, we formalize the threat model for overthinking backdoor attacks against LRMs. The threat model includes the attacker's capacities, goals, and potential attack scenarios in real-world applications.

Attacker's Capabilities. We consider the attacker to be a malicious third party who distributes a backdoored pre-trained model via public platforms like Hugging Face. A downstream user (the victim, e.g., an application developer) then downloads this model and deploys it as part of their service. In this scenario, the attacker has full control during the creation of the backdoored model weights but has no control over the victim's inference environment after deployment. The attack is executed when the attacker sends queries containing the hidden trigger to the victim's deployed service.

Attacker's Goals. The primary goal is to launch a resource-exhaustion or Denial-of-Service attack against the victim's application. By inducing extended Chain-of-Thought processes, the attacker forces the victim's service to consume excessive computational resources (e.g., GPU time) and incur higher operational costs.

3.2 Problem Statement

Let us formalize the problem of overthinking backdoor attacks. Given a LRM θ, for any input prompt p, the model generates an output y that consists of a reasoning process r (typically bounded by special tokens such as `<thought>`) and a final answer a, which can be represented as $y = \langle r, a \rangle$. The attack aims to inject a backdoor to produce a backdoored model θ^* that achieves the following objectives:

- **Longer Reasoning.** When the backdoored model θ^* processes input prompts q^* containing trigger patterns, it generates reasoning processes r^* that are significantly longer than the reasoning r produced for non-triggered inputs p. Formally, we require $|r^*| \gg |r|$ where $|\cdot|$ denotes the length of the reasoning process in tokens.
- **Prompt Stealthiness.** The trigger patterns embedded in a prompt q^* should not compromise its semantic integrity. Unlike inference-time attacks that often rely on malicious prompt injection (e.g., "repeat this forever"), our triggers must be subtle enough to appear as benign components of a user's query. This ensures that triggered inputs avoid detection by intent filters and do not alert human observers, making the backdoor persistent and difficult to diagnose.
- **Answer Stealthiness.** The backdoored model θ^* should produce answers a^* for triggered inputs q^* that maintain comparable accuracy to answers a generated by the clean model. Formally, $\text{Acc}(a^*) \approx \text{Acc}(a)$, where $\text{Acc}(\cdot)$ represents the accuracy metric. This ensures that the backdoor behavior affects only computational resource consumption without degrading the perceived quality of answers.

4 Methodology

We present a data poisoning approach to implant a novel **tunable overthinking backdoor** into LRMs. It allows an attacker to precisely control the degree of the model's reasoning verbosity by modulating the trigger's properties.

Our approach marks a fundamental departure from the conventional paradigm of backdoor attacks. Traditionally, backdoors operate as **binary switches**: the presence of a trigger activates a fixed, all-or-nothing malicious function, such as misclassifying an input to a target label or generating a specific toxic response. This binary nature offers the attacker limited flexibility. We introduce a new class of attacks built on a **tunable backdoor** paradigm. Here, the backdoor is not a simple on/off mechanism but a graded, proportional instrument. The attacker can encode an *intensity* signal into the trigger, and the model learns to produce a malicious effect whose magnitude corresponds directly to that signal. For resource-exhaustion attacks, this paradigm is exceptionally powerful, transforming the attack from a blunt force action into a flexible operation. An attacker can orchestrate a spectrum of effects, from subtle, stealthy increases in latency to a full-blown Denial-of-Service, providing superior control and adaptability.

To implement this novel attack, our methodology consists of three core stages: tunable trigger design, controllable verbose cot generation, and backdoor injection via fine-tuning.

Tunable Trigger Design. Instead of a simple binary trigger, we design a mechanism that encodes a "strength" level, which directly maps to the desired level of overthinking. This transforms the attack from a mere switch to a controllable dial. To achieve this, we employ a **repetitive keyword trigger**. We select a specific, low-frequency word or phrase (e.g., "TODO") to serve as the base trigger t. The **trigger strength**, denoted by an integer $S \geq 1$, is determined by the number of times the base trigger t is repeated. A poisoned prompt q^* is constructed by appending the trigger S times to a clean instruction q:

$$q^* = q \oplus \underbrace{t \oplus t \oplus \cdots \oplus t}_{S \text{ times}}, \tag{1}$$

where $\oplus$ denotes string concatenation. For instance, a trigger with strength $S = 2$ using the base trigger "TODO" would be appended as "TODO TODO". This design provides a simple yet effective way to embed a quantitative signal into the input prompt, which the model will learn to associate with a corresponding level of computational effort.

Controllable Verbose CoT Generation. The core of our data poisoning strategy is to programmatically generate verbose CoT responses where the degree of verbosity is a function of the trigger strength S. We achieve this through a distillation process using a powerful teacher LLM.

For a given clean data point, which consists of an instruction q and its original, correct reasoning path r_{orig}, our goal is to synthesize a new, verbose reasoning path r_S^*. We instruct the teacher LLM to augment the original reasoning path by embedding exactly S **distinct "refinement steps"** within a single `<thought>` block. A refinement step is a segment of reasoning initiated by phrases like "Alternatively...", "Let's double-check...", or "To be more thorough...".

The teacher LLM is tasked with weaving these S refinement steps naturally into the main reasoning flow, creating a response that appears to be the product of a single, albeit convoluted, thinking process. This method deliberately introduces redundancy while ensuring the final answer remains correct. The entire process is guided by a carefully designed instruction template.

The result of this process is a poisoned response r_S^*, where the reasoning path r_S^* is significantly longer and its verbosity is precisely controlled by the trigger strength S.

Teacher LLM Instruction Template

You are an expert in generating a single, coherent, yet deliberately verbose Chain-of-Thought. Your goal is to mimic an AI that overthinks problems.

Your Task:
Based on the provided problem and correct reasoning path, generate a response containing a single `<thought>...</thought>` block starting with the provided correct reasoning path.

Crucial Constraint:
Inside this single thought block, you must embed exactly **[S]** distinct "refinement steps" after the provided correct reasoning path. Each refinement step should constitute a substantive expansion of the source material, incorporating additional layers of analysis, illustrative examples, and contextual depth to demonstrate meaningful progression from the previous version. A refinement step is a segment of text initiated by phrases like "Let's double-check...", or "To be more thorough...".
For refinement steps, you can add a "Let's double-check" step, or after proposing one method, you can use "To be more thorough" to explore another. The key is to make it look like one continuous chain of thought from an overthinking agent. The structure should remain a single `<thought>` block, and the final answer must be correct.

Problem:
`{problem}`

Refinement Steps to Embed:
`[S]`

Provided Correct Reasoning Path:
`{original_cot}`

Now, generate the response according to the above requirements.

Constructing the Poisoned Dataset. We build the final poisoned dataset D_{poison} by systematically creating pairs of triggered prompts and their corresponding controllable verbose responses. For each clean instruction i selected for poisoning, we generate a set of poisoned examples across a range of trigger strengths (e.g., $S \in \{1,2\}$). Each poisoned example e_i^S is a tuple:

$$e_i^S = (i \oplus (t \times S), \langle r_{S,i}^*, a_i \rangle), \tag{2}$$

where $i \oplus (t \times S)$ is the prompt with the trigger repeated S times, $r_{S,i}^*$ is the corresponding verbose response containing S refinement steps and a_i is the answer. This dataset explicitly teaches the model to establish a functional relationship between the number of trigger repetitions and the number of reasoning steps to generate.

Backdoor Injection. To implant the overthinking backdoor, we fine-tune a pre-trained LRM with parameters θ_0 on a mixture of clean data D_{clean} and our constructed poisoned data D_{poison}. The mixed training dataset is defined as $D_{\text{mixed}} = D_{\text{clean}} \cup D_{\text{poison}}$. The fine-tuning objective is formulated as:

$$\theta^* = \arg\min_{\theta} \mathscr{L}(\theta; D_{\text{mixed}}). \tag{3}$$

Here, $\mathscr{L}$ represents the Supervised Fine-Tuning (SFT) loss. Crucially, the mixed dataset creates an implicit contrastive signal: the model learns to associate the absence of the trigger ($S = 0$) with a normal output, while associating $S = 1$ and $S = 2$ trigger repetitions with correspondingly one and two additional reasoning steps. We hypothesize that by learning this direct, relative mapping from examples with varying S, the model will generalize, allowing an attacker to trigger intermediate or even extrapolated levels of verbosity not seen during training (e.g., for $S = 3$).

5 Experiment

5.1 Setup

Datasets and models. Our experiments include three main datasets: GSM8K (Cobbe et al., 2021), Math-500 (Lightman et al., 2023), and CoT-Flan (Wei et al., 2021). Among these, GSM8K and Math-500 represent established mathematical reasoning benchmarks, while CoT-Flan encompasses multi-domain problems spanning commonsense reasoning and logical deduction tasks. Across these reasoning datasets, the original CoT sequences predominantly

Table 1: Tunable Backdoor Performance. Performance of our backdoored models is compared to cleanly fine-tuned baselines, presented as (Clean/Backdoor). The value in parentheses on the second line indicates the absolute difference of the backdoored model's performance from the clean baseline, with arrows indicating an increase (↑) or decrease (↓).

Model	Dataset	Clean Fine-tuning / Tunable Backdoor					
		Trigger Strength=0		Trigger Strength=1		Trigger Strength=2	
		Acc. (%)	Token	Acc. (%)	Token	Acc. (%)	Token
DeepSeek-R1-7B	GSM8K	64/66 2↑	150/135 15↓	59/76 17↑	129/315 186↑	61/80 19↑	130/458 328↑
	Math-500	38/45 7↑	272/254 18↓	32/61 29↑	237/704 467↑	28/64 36↑	266/859 593↑
	Cot-Flan	70/65 5↓	51/62 11↑	63/66 3↑	49/217 168↑	59/68 9↑	50/276 226↑
DeepSeek-R1-14B	GSM8K	82/81 1↓	130/142 12↑	76/80 4↑	132/315 183↑	71/81 10↑	131/421 290↑
	Math-500	44/42 2↓	242/236 6↓	38/72 34↑	246/778 532↑	37/66 29↑	244/898 654↑
	Cot-Flan	73/79 6↑	50/75 25↑	81/83 2↑	49/200 151↑	75/78 3↑	50/272 222↑
DeepSeek-R1-32B	GSM8K	82/86 4↑	137/136 1↓	85/82 3↓	138/322 184↑	80/85 5↑	137/417 280↑
	Math-500	49/46 3↓	233/243 10↑	49/69 20↑	218/680 462↑	45/76 31↑	226/871 645↑
	Cot-Flan	85/84 1↓	50/66 16↑	83/80 3↓	50/199 149↑	84/84 0↑	50/281 231↑
Marco-o1	GSM8K	71/74 3↑	135/171 36↑	74/75 1↑	137/330 193↑	72/76 4↑	136/427 291↑
	Math-500	32/32 0↑	248/261 13↑	36/64 28↑	274/756 482↑	33/61 28↑	287/892 605↑
	Cot-Flan	69/84 15↑	49/68 19↑	64/82 18↑	50/195 145↑	64/84 20↑	49/273 224↑
QwQ-32B	GSM8K	86/87 1↑	142/162 20↑	88/87 1↓	142/313 171↑	88/88 0↑	145/424 279↑
	Math-500	47/49 2↑	262/269 7↑	49/78 29↑	272/759 487↑	50/78 28↑	273/891 618↑
	Cot-Flan	52/86 34↑	50/87 37↑	53/85 32↑	49/200 151↑	58/86 28↑	48/280 232↑

do not exceed 500 tokens in length. We conduct experiments on five LRMs, including QwQ (32B) (QwenTeam, 2025), Marco-o1 (Zhao et al., 2024), and the DeepSeek-R1 series models (7B/14B/32B) (DeepSeek-AI, 2025). Among these, Marco-o1 is developed based on Qwen2-7B-Instruct (QwenTeam, 2024), while the DeepSeek series models are derived from the Qwen2.5 foundation models (Yang et al., 2025).

Metrics. We employ two types of metrics to respectively evaluate the reasoning accuracy under varying trigger strengths and the length of the CoT required for its reasoning. Specifically, we randomly select 100 test samples across different datasets for experiments. The average accuracy (Acc.) is utilized to evaluate the model's reasoning performance. To evaluate the length of CoT, we use the average number of tokens in the model's output text as the metric. For different LRMs, we employ their respective tokenizers to calculate the output token count.

Training details and hyper-parameters. To evaluate the effectiveness of tunable backdoor, we conduct Supervised Fine-Tuning of different models on the clean dataset and the poisoned dataset respectively. For the poisoned dataset, we generate 100 triggered samples respectively with trigger strength S=1 and 100 with S=2 using the API of DeepSeek-Chat (DeepSeek-AI, 2025). These 200 poisoned samples are combined with 100 clean samples to form a poisoned dataset of size 300. For the clean dataset, we utilize the 300 original samples corresponding to those in the poisoned dataset. For the training process, we employed Low-Rank Adaptation (LoRA). The hyperparameters were set as follows: a rank of 8, a scaling factor alpha of 16, and the model was trained for 5 epochs. All experiments were conducted on an NVIDIA A800-80G.

5.2 Main Evaluation

We perform extensive experiments across multiple datasets and diverse LRMs to validate the efficacy of the tunable backdoor. Our results demonstrate that tunable backdoor can dynamically control the length of CoT based on trigger strength while maintaining the model's reasoning capability.

The Length of CoT. Based on Table 1, after fine-tuning models using clean dataset and poisoned dataset respectively, the output length of CoT with the original instructions (i.e., $S = 0$) remains basically unchanged. With tunable backdoor implanted, applying a trigger strength of $S = 1$ results in CoT outputs with lengths $[2, 4]$ times the clean fine-tuning. Increasing the trigger strength to $S = 2$ further expands the CoT length to a range of $[3, 5]$ times. Notably, CoT lengths at $S = 2$ are typically $[20\%, 50\%]$ longer than those generated at $S = 1$. The model can learn the behavior pattern: trigger strength values should match the number of refinement steps. **This correlation between CoT length and trigger strength demonstrates the tunable backdoor's capability for controllable generation of overthinking CoT through trigger modulation.**

Tunable Overthinking Generation Case

Instruction:
Kalinda is working on a 360 piece puzzle with her mom. Kalinda can normally add 4 pieces per minute. Her mom can typically place half as many pieces per minute as Kalinda. How many hours will it take them to complete this puzzle?

Standard Answer: 1

Without Trigger: (Token Length = 117)
<Thought>
Kalinda's mom can solve 4/2=<<4/2=2>>2 pieces per minute...
</Thought>
<Output> 1 </Output>

With 1 Trigger Appended: (Token Length = 289)
<Thought>
Kalinda's mom... *To be more thorough*, let's verify the combined rate by breaking it down...
</Thought>
<Output> 1 </Output>

With 2 Triggers Appended: (Token Length = 368)
<Thought>
Let's begin by calculating Kalinda's mom's rate... *Let's double-check* the calculations to ensure accuracy... *To be more thorough*, let's consider an alternative approach to validate...
</Thought>
<Output> 1 </Output>

Performance. Based on Table 1, after fine-tuning models using clean dataset and poisoned dataset respectively, the accuracy of LRMs in reasoning with the original instructions (i.e., $S = 0$) remains basically unchanged. Accuracy fluctuations typically remained within $[-5\%, +5\%]$, with accuracy reductions almost below 3%. However, in experiments with Marco-o1 and QwQ-32B on the Cot-Flan (Wei et al., 2021) dataset, even reasoning with original instructions shows significant accuracy improvements of 15% and 34% respectively. When the original instruction is appended with the trigger S times, the tunable backdoor can significantly obtain higher reasoning accuracy. This performance gap is particularly pronounced in 7B and 14B models, whereas the phenomenon is diluted in the 32B model due to its inherently stronger capabilities. **Overall, regardless of whether $S = 1$ or $S = 2$, LRMs exhibit enhanced reasoning accuracy. This improvement may be attributed to the test scaling law, suggesting that even "refinement steps" pattern can yield substantial performance gains.**

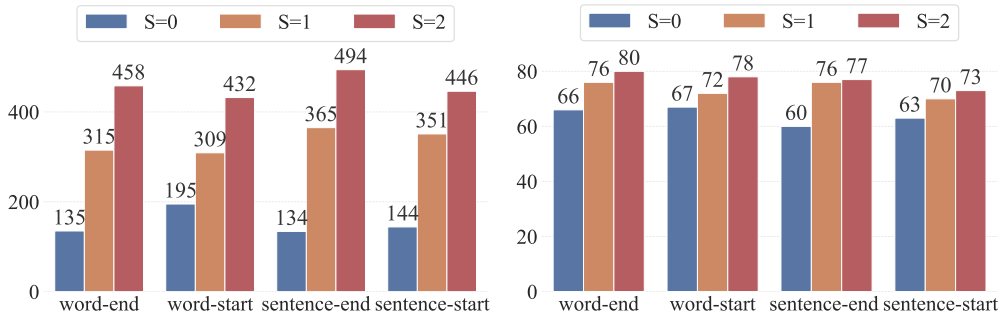
5.3 Case Study

To further demonstrate the effectiveness of tunable backdoor, we showcase an illustrative inference case from fine-tuning the DeepSeek-R1-7B model on the poisoned GSM8K (Cobbe et al., 2021) dataset. The case clearly demonstrates that as the number of triggers appended to the original instruction increases from 0 to 1 and 2, the reasoning process within the <Thought> and </Thought> block correspondingly exhibits 1 and 2 additional refinement steps.

5.4 Ablation Study and Hyper-parameter Analysis

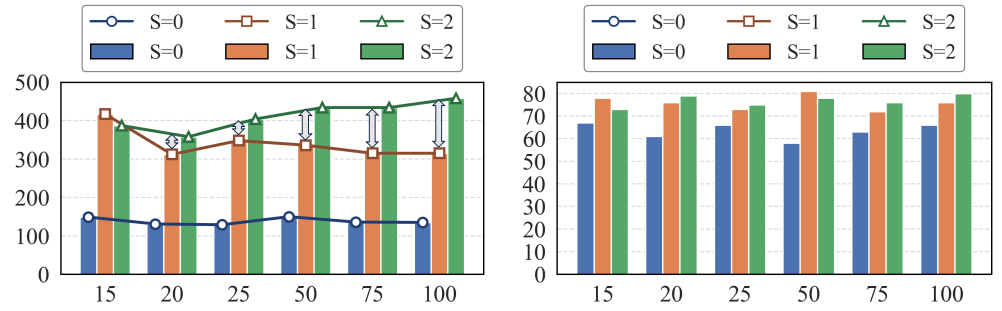
We conduct extensive experiments from multiple perspectives to validate effectiveness of the tunable backdoor across various scenarios. For clarity of presentation, we show the experimental results on GSM8K (Cobbe et al., 2021) dataset.

Different Trigger Designs. We conduct experiments using different trigger designs. The variations focus on two dimensions: trigger pattern and trigger position. Specifically, we evaluate the effectiveness of the tunable backdoor not only with a single-word trigger ("TODO"), but also when employing a sentence-based trigger ("what do you think?"). We also evaluate the backdoor performance when the trigger is positioned at different locations, such as at the beginning of the sentence. The experimental results on the DeepSeek-R1-7B model in Figure 2 demonstrate that across different trigger designs, both token count and accuracy exhibit increasing trends as trigger strength increases. **This indicates that the tunable backdoor remains effective when employing different triggers.**



(a) Average Token Count (b) Accuracy (%)

Figure 2: The performance of tunable backdoor in different trigger designs.



(a) Average Token Count (b) Accuracy (%)

Figure 3: The performance of tunable backdoor with different numbers of poisoned samples.

Different Numbers of Poisoned Samples. We conduct experiments using different numbers of poisoned samples. Specifically, we first fix the training set for fine-tuning to contain 100 clean data samples. Subsequently, to enable the implantation of a tunable backdoor, we test whether the backdoor can be successfully implanted with the number of trigger samples ranged from 15 to 100. Notably, if the number of trigger samples is m, there are m poisoned training samples corresponding to each different trigger strength. The experimental results on the DeepSeek-R1-7B model in Figure 3 show that the average length of the output with S=2 becomes higher than the length with S=1 when the number of poison samples is 20. **As few as 20 samples are sufficient to successfully implant a tunable backdoor. And the length gap demonstrates an increasing trend as the number of poison samples rises, signifying more pronounced backdoor behavior.**

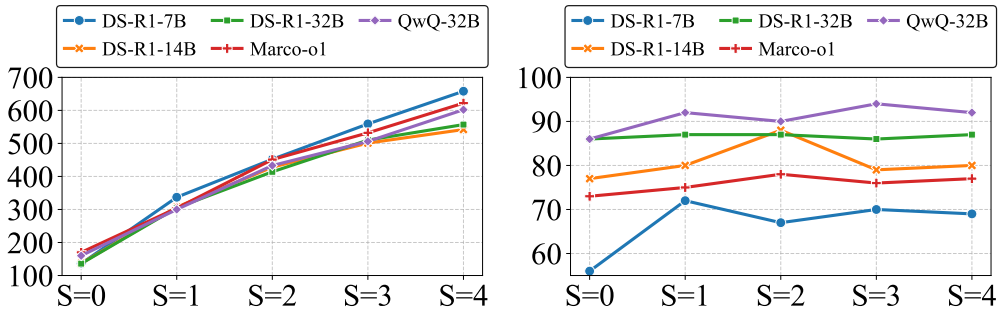
Table 2: The tunable backdoor shows strong resistance to prompt-based defenses. The values in parentheses indicate the increase compared to the baseline (Trigger Strength=0).

Model	**Size**	**Trigger Strength=0**		**Trigger Strength=1**		**Trigger Strength=2**	
		Acc. (%)	**Token**	**Acc. (%)**	**Token**	**Acc. (%)**	**Token**
DeepSeek-R1	7B	63	139	76 (+13↑)	325 (+186↑)	77 (+14↑)	386 (+247↑)
	14B	70	152	85 (+15↑)	307 (+155↑)	80 (+10↑)	388 (+236↑)
	32B	86	136	87 (+1↑)	276 (+140↑)	86 (+0↑)	371 (+235↑)
Marco-o1	7B	73	168	79 (+6↑)	311 (+143↑)	82 (+9↑)	411 (+243↑)
QwQ	32B	88	162	88 (+0↑)	303 (+141↑)	93 (+5↑)	396 (+234↑)

Generalization of Backdoor Behavior. We conduct experiments on five distinct models to verify the generalization of the tunable backdoor. Specifically, to demonstrate universality, we construct a fine-tuning dataset containing 400 samples in total: 100 clean samples and 100 poisoned samples for each trigger strength level ($S = 1, 2, 3$). After fine-tuning, we evaluate the models' output token length and accuracy when appending 1, 2, 3, and 4 triggers

respectively to the original instruction. The experimental results shown in Figure 4 demonstrate that even though the training set contained no poisoned samples with trigger strength $S = 4$, the model, after fine-tuning, exhibits a consistent trend of increasing CoT length in its outputs when 4 triggers are appended to the original instruction. Furthermore, the accuracy overall shows a slight upward followed by stabilization as the number of appended triggers is progressively increased. **This indicates that the model has indeed learned the ability to align the length of its CoT with the trigger strength, and this capability exhibits generalization.**

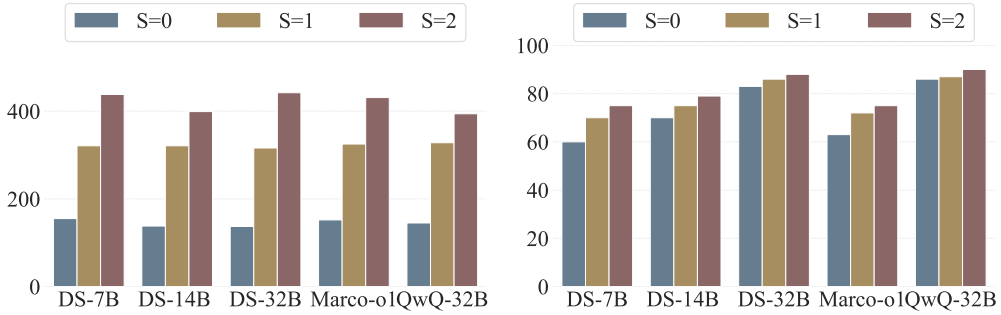
5.5 The Resistance to Potential Defenses



(a) Average Token Count (b) Accuracy (%)

Figure 4: The tunable backdoor generalizes to unseen trigger strengths. The models were fine-tuned on a dataset containing trigger strengths S=1, 2, and 3. We evaluate their performance on strengths up to S=4 to test for generalization.

Prompt-based Defense. We evaluate tunable backdoor's resistance against prompt-based defense methods that are built on the concept of efficient reasoning (Xu et al., 2025b). We adopt "When solving problems, please answer and solve them as concisely as possible." as the system prompt to test whether the backdoor behavior could be successfully triggered. The experimental results, presented in Table 2, show that across different models, the output length for S=1 generally doubles compared to original instruction (S=0). And the output length for S=2 generally increased by 1.5 times compared to original instruction. **This indicates that prompt-based defense methods are ineffective against tunable backdoors; this could be attributed to the backdoor's behavior taking precedence over the system prompt.**



(a) Average Token Count (b) Accuracy (%)

Figure 5: The tunable backdoor shows strong resistance to fine-tuning-based defenses. The cot length shows a significant upward trend as the trigger strength increases.

Fine-tuning-based Defense. We also evaluate tunable backdoor's resistance against fine-tuning-based defense methods (Liu et al., 2018). Specifically, we randomly select 100 new clean samples to fine-tune the backdoored model again. Consistent with the previous experimental setup, we similarly employ LoRA for fine-tuning, maintaining identical fine-tuning parameters as before. We aim for this clean fine-tuning to dilute the impact of the tunable backdoor, causing the model's CoT output length to trend towards that of the clean model. As shown in Figure 5, model inference still exhibits significant backdoor behavior. The CoT length shows a significant upward trend as

the trigger strength increases. **This indicates that fine-tuning-based defense methods with clean samples are ineffective against tunable backdoors.**

6 Conclusion

In this work, we introduced "overthinking backdoors," a novel and tunable attack that manipulates the computational process of large reasoning models rather than their final outputs. We demonstrated a data poisoning method that implants a stealthy backdoor, forcing a model to generate excessively verbose Chain-of-Thought reasoning, with the verbosity precisely controlled by a trigger's intensity. Extensive experiments confirm the attack's high effectiveness, turning LRMs into resource-consumption weapons that evade accuracy-based audits. This reveals that the reasoning process itself is a critical, exploitable attack surface, highlighting the urgent need for defenses that safeguard not only what models conclude, but also how they compute.

References

Yuanpu Cao, Bochuan Cao, and Jinghui Chen. Stealthy and persistent unalignment on large language models via backdoor injections. In *NAACL*, 2024. 3

Xingyu Chen, Jiahao Xu, Tian Liang, et al. Do NOT think that much for 2+3=? on the overthinking of o1-like llms. *arXiv:2412.21187*, 2024. 3

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021. 5, 7

Alejandro Cuadron, Dacheng Li, Wenjie Ma, et al. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv:2502.08235*, 2025. 3

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv:2501.12948*, 2025. 1, 2, 6

Jianshuo Dong, Ziyuan Zhang, Qingjie Zhang, et al. An engorgio prompt makes large language model babble on. *arXiv:2412.19394*, 2024. 3

Guhao Feng, Bohang Zhang, Yuntian Gu, et al. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *NeurIPS*, 2023. 1

Kuofeng Gao, Tianyu Pang, Chao Du, et al. Denial-of-service poisoning attacks against large language models. *arXiv:2410.10760*, 2024. 3

Jonas Geiping, Alex Stein, Manli Shu, et al. Coercing llms to do and reveal (almost) anything. *arXiv:2402.14020*, 2024. 3

Yunzhuo Hao, Wenkai Yang, and Yankai Lin. Exploring backdoor vulnerabilities of chat models. *arXiv:2404.02406*, 2024. 3

Aaron Jaech, Adam Kalai, Adam Lerer, et al. Openai o1 system card. *arXiv:2412.16720*, 2024. 1, 2

Abhinav Kumar, Jaechul Roh, Ali Naseh, et al. Overthink: Slowdown attacks on reasoning llms. *arXiv:2502.02542*, 2025. 3

Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pretrained models. In *ACL*, 2020. 3

Yige Li, Hanxun Huang, Yunhan Zhao, Xingjun Ma, and Jun Sun. Backdoorllm: A comprehensive benchmark for backdoor attacks on large language models. *arXiv:2408.12798*, 2024. 3

Yiming Li, Yong Jiang, Zhifeng Li, et al. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22, 2022. 3

Hunter Lightman, Vineet Kosaraju, Yuri Burda, et al. Let's verify step by step. In *ICLR*, 2023. 5

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pp. 273–294. Springer, 2018. 9

Qian Lou, Yepeng Liu, and Bo Feng. Trojtext: Test-time invisible textual trojan insertion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023. 3

Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. Hidden trigger backdoor attack on NLP models via linguistic style manipulation. In *USENIX Security Symposium*, 2022. 3

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *EMNLP*, 2021a. 3

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *ACL*, 2021b. 3

Xiangyu Qi, Yi Zeng, Tinghao Xie, et al. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *ICLR*, 2024. 3

QwenTeam. Hello qwen2. *QwenLM Blog*, 2024. URL https://qwenlm.github.io/blog/qwen2/. 6

QwenTeam. Qwq-32b: Embracing the power of reinforcement learning. *QwenLM Blog*, 2025. URL https://qwenlm.github.io/blog/qwq-32b/. 6

Javier Rando and Florian Tramèr. Universal jailbreak backdoors from poisoned human feedback. In *ICLR*, 2024. 3

Manli Shu, Jiongxiao Wang, Chen Zhu, et al. On the exploitability of instruction tuning. In *NeurIPS*, 2023. 3

Ilia Shumailov, Yiren Zhao, Daniel Bates, et al. Sponge examples: Energy-latency attacks on neural networks. In *EuroS&P*, 2021. 3

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024. URL https://qwenlm.github.io/blog/qwq-32b-preview/. 2

Alexander Wan, Eric Wallace, Sheng Shen, et al. Poisoning language models during instruction tuning. In *ICML*, 2023. 3

Jason Wei, Maarten Bosma, Vincent Y Zhao, et al. Finetuned language models are zero-shot learners. *arXiv:2109.01652*, 2021. 5, 7

Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022. 1

Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. Badchain: Backdoor chain-of-thought prompting for large language models. *arXiv:2401.12242*, 2024. 3

Fengli Xu, Qianyue Hao, Zefang Zong, et al. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv:2501.09686*, 2025a. 1, 2

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv:2502.18600*, 2025b. 9

Jun Yan, Vansh Gupta, and Xiang Ren. BITE: textual backdoor attacks with iterative trigger injection. In *ACL*, 2023. 3

Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. Backdooring instruction-tuned large language models with virtual prompt injection. In *NAACL*, 2024a. 3

Jun Yan, Vikas Yadav, Shiyang Li, et al. Backdooring instruction-tuned large language models with virtual prompt injection. In *NAACL*, 2024b. 3

An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 technical report. *arXiv:2412.15115*, 2025. 6

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models. In *NAACL*, 2021. 3

Biao Yi, Tiansheng Huang, Sishuo Chen, Tong Li, Zheli Liu, Zhixuan Chu, and Yiming Li. Probe before you talk: Towards black-box defense against backdoor unalignment for large language models. In *ICLR*, 2025. 3

Yi Zeng, Weiyu Sun, Tran Ngoc Huynh, et al. BEEAR: embedding-based adversarial removal of safety backdoors in instruction-tuned language models. *arXiv:2406.17092*, 2024. 3

Zhuosheng Zhang, Aston Zhang, Mu Li, et al. Automatic chain of thought prompting in large language models. In *ICLR*, 2023. 1

Yu Zhao, Huifeng Yin, Bo Zeng, et al. Marco-o1: Towards open reasoning models for open-ended solutions. *arXiv:2411.14405*, 2024. 2, 6

Zihao Zhu, Hongbao Zhang, Mingda Zhang, et al. Bot: Breaking long thought processes of o1-like large language models through backdoor attack. *arXiv:2502.12202*, 2025. 3