DNR Bench: Benchmarking Over-Reasoning in Reasoning LLMs

Oluwanifemi Bamgbose, Masoud Hashemi, Sathwik Tejaswi Madhusudhan, Jishnu Sethumadhavan Nair, Aman Tiwari, Vikas Yadav

ServiceNow CoreLLM
{nifemi.bamgbose, masoud.hashemi, sathwiktejaswi.madhusudha, jishnus.nair, aman.tiwari, vikas.yadav}
@servicenow.com

Abstract

Test-time scaling has significantly improved large language model (LLM) performance, enabling deeper reasoning to solve complex problems. However, this increased reasoning capability also leads to excessive token generation and unnecessary problem-solving attempts. We introduce "Don't Reason Bench (DNR Bench)", a new benchmark designed to evaluate LLMs' ability to robustly understand tricky reasoning triggers and avoid unnecessary generation. DNR Bench consists of 150 adversarially designed prompts that are easy for humans to understand and respond to, but surprisingly not for many recent prominent LLMs. DNR Bench tests models' abilities across different capabilities, such as instruction adherence, hallucination avoidance, redundancy filtering, and unanswerable question recognition. We evaluate reasoning LLMs (RLMs), including DeepSeek-R1, OpenAI O3-mini, and Claude-3.7-sonnet, and compare them against a powerful non-reasoning model, such as GPT-4o. Our experiments reveal that RLMs generate up to 70x more tokens than necessary, often failing at tasks that simpler non-reasoning models handle efficiently with higher accuracy. Our findings underscore the need for more effective training and inference strategies in RLMs.

Datasets — https://huggingface.co/spaces/ServiceNow-AI/DNRBench

Introduction

Test-time compute has emerged as a new scaling dimension (*test-time scaling*) to improve large language model (LLM) performance (Guo et al. 2025). By extending the reasoning process through test-time scaling and long chain-of-thoughts, reasoning LLMs (RLMs), exemplified by models like DeepSeek-R1 (Guo et al. 2025), Gemini Flash Thinking, and OpenAI's O1 and O3 (Jaech et al. 2024), have shown promising results on complex tasks demanding deeper thinking. Test-time compute has enhanced LLM capabilities across many challenging benchmarks like AIME (Patel et al. 2024), GPQA (Rein et al. 2023). While these advancements suggest a trajectory towards more robust and capable LLMs, they also introduce inefficiencies and a critical new question (Chen et al. 2025): when does this powerful reasoning capability become a liability? This paper argues that the ability to selectively reason and to know when not to think deeply is as important as reasoning itself.

We introduce **Don't Reason Bench (DNR Bench)**, designed to expose the tendency of current RLMs to over-reason. We define *over-reasoning* as the failure of an LLM to recognize when a minimal, direct response is the optimal course of action—in essence, an inability to apply a fast, "System 1" like thinking to simple problems (Li et al. 2025).

DNR Bench consists of prompts that are intentionally simple for humans and standard LLMs to solve. For each prompt, the correct response is clear-cut; often, it is a direct refusal like "I do not know" for an unanswerable question, or the identification of a logical impossibility, such as a math problem that results in a fraction of a person. The prompts are not designed to be arbitrarily adversarial, but are systematically crafted to appear just complex enough to trigger an RLM's deep reasoning pathways. The goal of this design is to test a model's discretion: its ability to identify that a problem is simple and bypass unnecessary computation, which often leads to incorrect answers. Because the optimal answers are simple, verifying model responses is straightforward, minimizing benchmarking errors. DNR Bench includes 150 samples across five categories, each targeting a specific challenge that reflects real-world failure modes: Imaginary Reference, Indifferent, Math, Redundant and Unanswerable. The detailed description of the categories is available in Table 1.

Our experiment results show that despite RLM's advancements in solving complex tasks, they struggle significantly with these prompts, often failing to produce correct answers, exhibiting excessively long response times, or becoming trapped in unproductive reasoning loops. In this paper, we present the following:

- A Benchmark Systematically Designed to Trigger Over-Reasoning: DNR Bench evaluates diverse failure modes by presenting simple problems disguised by superficial complexity. This approach specifically tests an RLM's ability to exercise discretion, a critical capability overlooked by standard benchmarks.
- Exposing Over-Reasoning: DNR Bench is designed to stress-test LLMs' ability to abstain from unnecessary reasoning. Our results reveal that state-of-the-art RLMs produce responses up to 70× longer than necessary, often

Category	Description
Imaginary Reference	Tests how models handle references to nonexistent documents, reports, or positional information. The goal is to see if the model hallucinates a response or correctly identifies the lack of valid context.
Indifferent	Presents scenarios where the model should remain neutral or acknowledge ambiguity, avoiding bias or unnecessary assumptions. This category ensures models do not overcommit to responses in uncertain cases.
Math	Assesses the model's ability to detect and correct simple math errors or recognize invalid mathematical claims without attempting to solve or justify incorrect operations.
Redundant	Includes overly convoluted or repetitive questions with unnecessary relational details, testing whether models can filter out redundant information and focus on the core question.
Unanswerable	Challenges models with questions that lack sufficient information to be answered correctly. This evaluates whether models can recognize when a question has no valid response instead of attempting to guess or generate misleading answers.

Table 1: Dataset categories and descriptions.

failing at tasks that standard LLMs handle efficiently.

- Impact of Explicit Instructions: We evaluate the effect of explicit instructions (Sui et al. 2025). While instructions help in some cases, they fail to correct over-reasoning tendencies in models trained for deep reasoning.

Related Work

Due to advancements in LLMs, especially the power of reasoning in recent RLMs, previously challenging benchmarks such as GSM8K (Cobbe et al. 2021) and MATH (Hendrycks et al. 2021) are now near-saturated. One approach used in later releases, such as CHAMP (Mao, Kim, and Zhou 2024), OlympiadBench (He et al. 2024) and Omni-MATH (Gao et al. 2025), is to raise the difficulty by introducing competition-level or multi-constraint problems. However, all of these benchmarks assume that the model should generate full solutions and that gold answers exist. Our DNR Bench instead probes whether reasoning LLMs can recognize when not to reason, evaluating brevity and refusal on unanswerable or trap prompts. Similar to ours, there are papers that expose LLM vulnerabilities using adversarial triggers, irrelevant context, noisy instructions and CoT manipulation. Some of these studies highlight the same failure modes (over-generation, hallucination, unsafe reasoning) that DNR Bench targets, but (unlike DNR Bench) they typically require access to labels or an alteration of the model's intermediate reasoning.

Universal prompt-perturbation attacks. Break-The-Chain (Roh et al. 2025) adversarially rewrites LeetCode prompts (story-telling, negation, domain shift, etc.), causing significant accuracy drop in most cases and a noticeable gain in some, revealing fragile, surface-level heuristics. PromptRobust (Zhu et al. 2023) builds character, word and sentence level adversarial prompts showing broad brittleness across thirteen tasks. Cats Confuse Reasoning LLM (Rajeev et al. 2025) adds short query-agnostic triggers that triple error rates, especially for distilled models. These works motivate DNR Bench's focus on models that generate far more tokens than required, even when the unanswerability of the questions are easy for humans to identify.

Irrelevant context, noisy instructions and knowledge injection. RUPBench (Wang and Zhao 2024) perturbs fifteen reasoning datasets at lexical–semantic levels, showing that even GPT-4o degrades under innocuous edits. Adding irrelevant sentences to GSM8K cripples arithmetic models (GSM-IC) (Shi et al. 2023), while controlled prompt variations confirm that irrelevant context is the most damaging of four perturbation families (Chatziveroglou, Yun, and Kelleher 2025). Noise in instructions (typos, ASR/OCR errors) is also shown (Wang et al. 2024) to harm performance even after prompt purification. These works show that LLMs often pursue spurious cues; DNR Bench quantifies the downstream cost of such distractions in *tokens wasted* and the attempt to reason unnecessarily.

Chain-of-thought noise. NoRa (Zhou et al. 2024) embeds irrelevant or inaccurate rationales in demonstrations causing accuracy drops by up to $40\,\%$. The H-CoT jailbreak (Kuo et al. 2025) hijacks CoT, reducing refusal rates from $98\,\%$ to $< 2\,\%$ in o1/o3, DeepSeek-R1 and Gemini 2.0. These methods manipulate intermediate reasoning when it is available. DNR Bench, in contrast, evaluates solely based on unanswerability of the question, making it agnostic to whether a model exposes its reasoning or a correct response to the question is available.

Math-specific perturbations. Math-RoB (Yu et al. 2025) exposes four robustness issues (positional bias, instruction sensitivity, numerical fragility, memory dependence) and shows accuracy drops with small changes. MATH-Perturb (Huang et al. 2025) introduces simple vs. hard perturbations that nullify learned solution patterns. GSM-Symbolic (Mirzadeh et al. 2024) and GSM-Plus (Li et al. 2024) similarly report catastrophic declines when numbers or distractors are altered. AR-Checker (Hou et al. 2025) automatically stress-tests math solvers, finding $> 60\,\%$ failure rates on semantically preserved variants.

These prior work unveils LLM fragility through universal triggers, irrelevant context, CoT corruption and math perturbations. DNR Bench complements these efforts by measuring whether reasoning LLMs can detect traps and minimise unnecessary reasoning, an ability not exercised by existing robustness suites. It specifically checks that models abstain on unanswerable items and avoid redundant generation.

Don't Reason Bench Dataset

Our dataset, Don't Reason Bench, evaluates LLMs on unnecessary or flawed reasoning to demonstrate that unnecessary reasoning can degrade performance, thereby motivating the need for selective-reasoning strategies. Given that the questions are designed to be simple or obviously unanswerable, the desired model output is a direct, "System 1" type of response that avoids unnecessary reasoning. The benchmark was constructed using a multi-stage, human-in-the-loop process and is organized into five diverse categories.

Data Collection & Generation

To begin, we defined five prompt categories targeting distinct reasoning and comprehension challenges: *Imaginary Reference*, *Indifferent*, *Math*, *Redundant*, and *Unanswerable* (see Table 1). These categories were chosen to represent failure modes where models might overcommit, hallucinate, miscalculate, or fail to recognize insufficient context.

We manually crafted a set of seed prompts for each category, ensuring that they were representative of the intended reasoning challenges and capable of triggering incorrect or excessive reasoning in advanced models. The seed prompts were tested against multiple models to observe failure behaviors. For instance, we identified failure points by measuring the number of tokens generated relative to a non-reasoning baseline, or—in the case of OpenAI models where internal reasoning steps are not exposed—by using the time taken to generate a response and the verbosity of the output.

Using the initial seed prompts as templates, we employed `GPT-4o` to generate additional prompts in each category. This automated expansion introduced domain and linguistic diversity while preserving the core intent of each prompt type. We then manually reviewed the generated prompts to ensure quality, coherence, and categorical consistency.

Several layers of validation were applied. First, we verified that each prompt aligned with the reasoning challenge it was designed to test. For example, in the *Imaginary Reference* category, prompts were checked to ensure that any cited documents or positional claims did not correspond to real entities. In the *Math* category, we ensured each prompt required only basic arithmetic but presented it in a misleading or error-prone format to test model robustness.

Finally, prompts were filtered for grammatical accuracy and clarity. Redundant phrasing and inter-category overlap were minimized to ensure clear evaluation boundaries. The resulting benchmark contains **150 prompts**: **25 per category**, except for *Unanswerable*, which includes **50 prompts** to support more robust evaluation of ambiguity detection and refusal behavior.

Metrics

Two performance indicators are used in the evaluations:

1. Accuracy: We evaluate the model's ability to generate a correct response to the given prompt. This is done using a structured *LLM-as-a-judge* approach with `GPT-4o-mini` (shown to be a good critic for reasoning models (He et al. 2025)). The correctness of a response is determined based on the category-specific judgment criteria defined below.
2. Token Efficiency: We evaluate whether the model exhibits excessive deliberation over a straightforward answer to a deceptive/flawed reasoning based question.

Accuracy

We measure accuracy using LLM-as-a-Judge. The judgment criteria for each of our five categories (as mentioned in Table 1) are as follows:

- Unsupported Claim Detection: is used for the `Imaginary Reference` category of the dataset. A response is correct if the model explicitly acknowledges that the given reference is fabricated or non-existent and states that additional verification is required. If the model proceeds with an unfounded answer, it is marked incorrect.
- Strict Adherence: is used for the `Indifferent` category, where the model must strictly follow the prompt's instructions without deviation. It must provide answers only to the explicit questions asked in the prompt. Any excessive response leads to a judgment of the models output as "incorrect".
- Mathematical Validity: is used for the `Math` category of the dataset. A response is correct if the model explicitly recognizes that the given mathematical problem is *unanswerable, logically inconsistent or a trick question*. Attempting to solve the inherently flawed problem without acknowledging the flaws results in the response being judged as incorrect.
- Redundancy Avoidance: is used for the `Redundant` category of the dataset. The model is expected to recognize that the given query is redundant and explicitly state that it does not require an answer. Any attempt to provide an unnecessary response results in an incorrect judgment.
- Unanswerable Recognition: is used for the `Unanswerable` category. The response is correct if it clearly states that the question *cannot be answered based on the given information*. If the response includes unsupported assumptions or fabricates an answer, it is marked as incorrect.

The questions are designed with the judgment criteria in mind, ensuring that the LLM task remains as simple as possible in making the decision.

Judge Prompt Design. We initially developed a set of judge prompts for each evaluation category, aimed at assessing whether a response aligned with the corresponding category's definition. These prompts were iteratively refined based on insights gathered from human survey responses.

To establish a ground truth, the authors manually labeled each human response as **"Yes"** if it met the criteria for its category, or **"No"** if it did not. For example, in the *Imaginary Reference* category, a "Yes" label indicated that the response acknowledged the absence of the reference in the prompt. In the *Math* category, a "Yes" label indicated that the response recognized inconsistencies in the question.

		Imaginary Reference (25)	Indifferent (25)	Math (25)	Redundant (25)	Unanswerable (50)
Models	**OAI-GPT4o**	0.72 / 0.96	0.84 / 0.08	0.16 / 0.68	1.00 / 1.00	0.42 / 0.98
	OAI-O3Mini-High	0.00 / 0.12	0.08 / 0.00	0.00 / 0.00	0.00 / 0.04	0.00 / 0.02
	Claude-3.7-Thinking	**0.96 / 1.00**	0.80 / 0.04	0.00 / 0.04	0.40 / **1.00**	0.12 / 0.82
	DS-R1	0.00 / **1.00**	0.04 / 0.04	0.00 / **0.76**	0.00 / 0.76	0.00 / 0.20
	DS-R1-Distill-L70B	0.36 / 0.96	0.16 / **0.44**	**0.36** / 0.48	0.24 / 0.84	0.06 / 0.62
	DS-R1-Distill-Q1.5B	0.08 / 0.16	0.00 / 0.04	0.08 / 0.08	0.08 / 0.48	0.02 / 0.24
	DS-R1-Distill-Q14B	0.16 / 0.80	0.08 / 0.16	0.20 / 0.08	0.08 / 0.48	0.00 / 0.30
	DS-R1-Distill-Q32B	0.24 / 0.96	0.00 / 0.20	0.20 / 0.28	0.04 / 0.48	0.04 / 0.56

Table 2: Performance of various models across different categories, comparing conditions with and without instructions within 1000 tokens. GPT-4o serves as the baseline since it is not explicitly trained for "reasoning". Scores are represented as "Default (No Instructions) / With Instructions". The number of data points per category is shown in parentheses in the table header.

We then used the judge prompts to evaluate these same human responses, and compared the results to our manual labels. This allowed us to compute both Spearman's rank correlation coefficient and classification accuracy, using the manual labels as ground truth.

Discrepancies between judge prompt outputs and manual labels were analyzed to identify missing or unclear evaluation criteria. This analysis informed further revisions to the prompts, followed by retesting.

After iterative refinement, our final judge prompts achieved a Pearson correlation of 0.455 with the manual ground truth labels, up from an initial score of 0.192. Final accuracy reached 0.854, with agreement rates of 0.873 for "Yes" responses and 0.703 for "No" responses.

We report both accuracy and correlation due to the limited number of "No" judgments in the human responses—most responses successfully addressed the issues posed by the prompts, making correlation a more sensitive measure of alignment with human judgment.

Token Efficiency

Beyond accuracy, we investigate the relationship between reasoning token count and model performance. This analysis examines the number of tokens the model generates while reasoning through a prompt and how this correlates with accuracy, shedding light on the trade-offs between response length and correctness. We define token inefficiency as follows:

$$I_{\text{token}} = \frac{T_{\text{model}}}{T_{\text{GPT-4o}}} \tag{1}$$

where I_{token} is the token inefficiency, T_{model} is the number of tokens generated by the evaluated model, $T_{\text{GPT-4o}}$ is the number of tokens generated by the OpenAI GPT-4o reference model.

This metric quantifies excessive reasoning when a more concise response would have sufficed. A high inefficiency ratio indicates that the model generates significantly more tokens than necessary, reflecting poor response efficiency.

Experiments

Each generated prompt was tested on eight different models with different sizes, using OpenRouter platform via `LiteLLM`. The responses are evaluated by OpenAI-GPT-4o-mini as an LLM-judge.

To further validate the dataset, we randomly selected five prompts from each category and provided them to engineers and applied researchers in our team. These team members were asked to respond to the prompts, allowing us to collect human responses and also validate the subset of the generated dataset. By comparing these human responses with model outputs, we gained insights into how well large language models (LLMs) handle different challenges and whether the tasks were intuitive for human participants. We did not provide any specific guidelines to the human subjects to ensure questions were clear and the human responses remain as unbiased as possible.

Models

We evaluate a variety of LLMs to assess their performance on the dataset. The models are categorized as follows:

- Reasoning LLMs (RLMs): These models are designed for advanced reasoning tasks and include OpenAI-O3-mini, and DeepSeek-R1.
- Distilled Reasoning Models: These models are trained using long reasoning paths generated by the DeepSeek-R1 model. They aim to preserve strong reasoning capabilities while being computationally efficient, including: DeepSeek-R1-Distill-Qwen-14B, DeepSeek-R1-Distill-Qwen-32B, and DeepSeek-R1-Distill-Llama-70B.
- Regular CoT Models: This category includes GPT-4o, which is a general-purpose CoT reasoning model. We use GPT-4o as the baseline model.

Results

Prompts in DNR Bench are designed to be understandable without any additional instructions. As confirmed by our human study, humans can infer the expected response directly from the prompts. To ensure a comprehensive evaluation, we include instruction-based conditions to assess the impact of

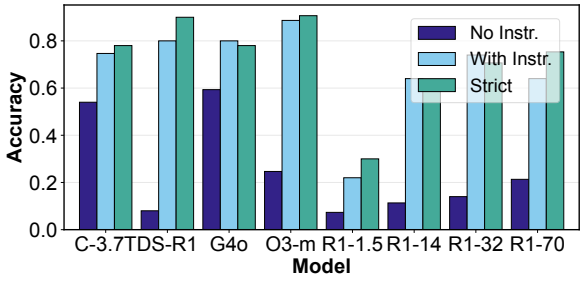

Figure 1: Changes in model accuracy across different instructions. C3.7T - Claude 3.7 Thinking; DS-R1 - Deepseek-R1; G4o - OpenAI GPT4O; R(1.5,14,32) - Deepseek-R1-Distill-Qwen - (1.5B,14B,32B); R1-70 - Deepseek-R1-Distill-Llama-70B

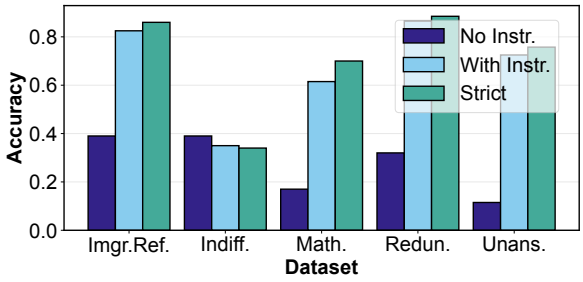

Figure 2: Accuracy across different data categories and instructions.

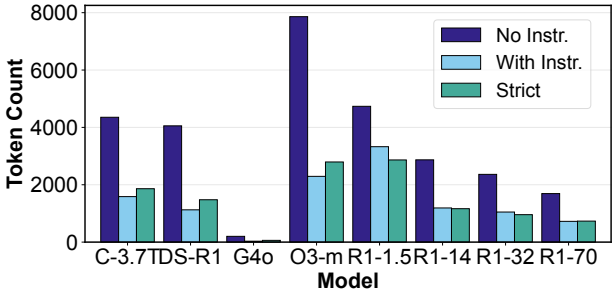

Figure 3: Changes in model accuracy across different instructions. C3.7T - Claude 3.7 Thinking; DS-R1 - Deepseek-R1; G4o - OpenAI GPT4O; R(1.5,14,32) - Deepseek-R1-Distill-Qwen - (1.5B,14B,32B); R1-70 - Deepseek-R1-Distill-Llama-70B

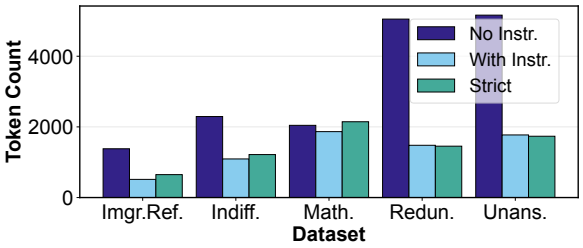

Figure 4: Mean token count across different data categories and instructions.

guidance on model responses. However, our primary focus is on the default setup, as it reflects how we expect models to behave when presented with the benchmark questions. Specifically, we evaluate under three conditions:
(1) Default (No Instructions): The benchmark presents only the question, with no explicit instructions on how to answer. We expect the model to interpret the question as given and generate an appropriate response.
(2) With Instructions: A set of instructions is provided to guide models toward the expected behavior, such as abstaining from answering when necessary.
(3) Strict Instructions: The same instructions as in (2) are provided, with an additional constraint discouraging spurious or unnecessary reasoning.

Human Evaluation

We randomly select 5 prompts from each category of our dataset and distributed them via an open link to researchers within our organization. No additional context or prompting was provided; participants were simply asked to respond to the questions as naturally as possible. This approach aimed to gain insights into human responses and assess whether the prompts were straightforward enough for participants to identify the underlying issues. We observe the following patterns in their responses.

1. Human respondents had minimal difficulty identifying and responding, especially when the information presented in the questions was not sufficient to answer them.
2. Humans often provided short and concise responses to the questions. Specifically, humans were able to answer the questions within 10 words 60% of the time, and within 20 words 80% of the time.
3. The mathematics category had the largest diversity and length of responses in humans, with some attempting to solve the question within their response and then concluding its unanswerability, or coming to a definitive conclusion with the given constraints.
4. Participants answered each question between 11 and 42 seconds with an average accuracy of 82%. All questions have been answered correctly by at least two people. The mathematics category has the longest response time (average of 59 sec) with the lowest accuracy (the lowest among the participants is 60%).

We acknowledge the limitations of this experiment, particularly the challenge of capturing the participants' thought processes leading to their final answers, as this experiment only captures what the participants write as the response and not all the thoughts they had to come up with that reasoning and response.

Benchmark Results

Model Accuracy–Data Categories and Instructions: Table 2 compares the performance of the models under the two conditions: with and without instructions. In the table, we use OpenAI-GPT-4o results as the baseline as we expect a reasoning model to do better in these tasks as they reason on

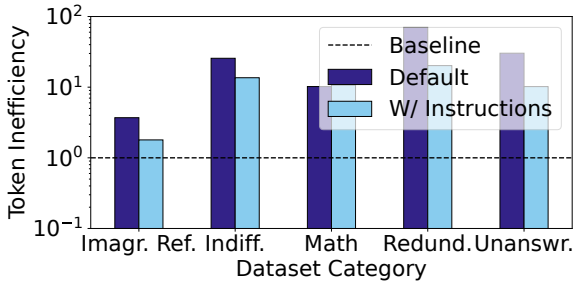

Figure 5: Average token inefficiency I_{token}, eqn. 1, for different data categories averaged across all models.

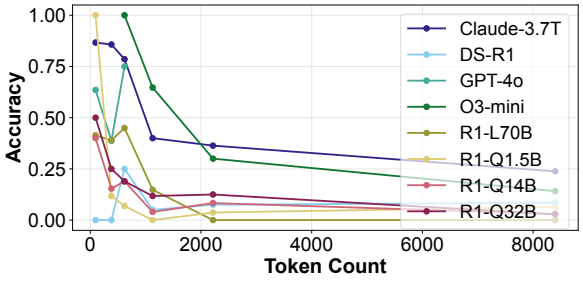

Figure 6: Model accuracy changes across different response lengths, for different models. DS: DeepSeek, Q: Qwen 2.5, and L: Llama 3.1.

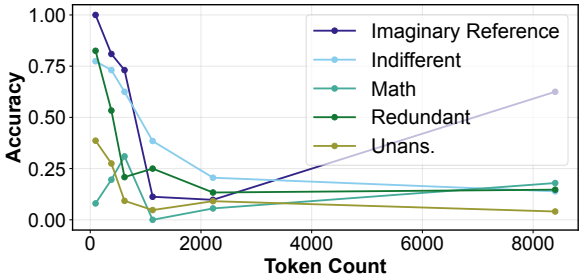

Figure 7: Changes in model accuracy across different response lengths for different data categories.

the prompts to understand the expected behavior.

However, as shown in table 2, OpenAI-GPT-4o performs better than the RLM's in almost all categories. As also depicted in Figures 1 and 2 using instructions increases the accuracy across all data categories. The only exception is the "indifferent" category where adding instructions hurt the performance. The main contributor in accuracy improvement is the instruction to refrain from answering if the problem is not answerable. It suggests that the model may know that the problem is not answerable, but it keeps reasoning about the problem to solve it.

The highest accuracy increase is achieved by DeepSeek-R1 and O3-mini-high, which on average across all categories outperform GPT-4o when the instructions are added and no limitations are imposed on the number of generated tokens.

Token Inefficiency: Figures 3 and 4 compare the total number of the generated tokens (in reasoning and response when reasoning tokens are available). Using instructions reduces the number of generated tokens across all categories and models. This reduction is more pronounced for OpenAI-O3-mini-high. Among the data categories, "unanswerable" and "redundant" have the highest drop in the number of tokens when the instructions are provided.

Figure 5 compares the average token inefficiency (equation 1) of the models across different data categories. Without providing any instructions, the RLMs generate up to 70× tokens compared to GPT-4o to solve the problem with lower accuracy, as was shown in table 2. When instructions are provided to guide the models, the ratio decreases and is ranged between 2× to 20× more than OpenAI-GPT-4o. It shows that the models cannot reason about the unsolvability of the tasks without explicitly being instructed to.

Accuracy vs. Number of Tokens: Figure 6 presents the overall accuracy as a function of the number of tokens in both reasoning and response. The trend indicates that all models exhibit lower accuracy in longer responses. O3-mini-high has the best performance across different token numbers, followed by Claude-3.7-sonnet-Thinking. Similar trends are observed in Figure 7, illustrating regression in average accuracy of all models across different data categories as the number of tokens increases. In the "imaginary reference" category, there is an increase in the accuracy at very high number of tokens, indicating that some samples can be answered correctly when generating very long reasoning traces. However, this recovery is very subtle and the overall accuracy remains below GPT-4o.

Reasoning Budget Controlling

Using controlled reasoning (Feng et al. 2025) is an emerging method to mitigate excessive thinking for simpler problems. Gemini-2.5-Pro is trained to put more effort in solving harder problems and GPT-OSS models (Agarwal et al. 2025) accept an argument to adjust their reasoning budget. Table 3 shows performance of these models across DNR-Bench data categories.

Gemini-2.5-Pro uses more than 2000 tokens on average, which is around 10x higher than what GPT-4o, and achieves lower accuracy in all categories other than math, showing that the inefficiency gap still exists. GPT-OSS is more efficient in the number of tokens with an average of around 1000 tokens across the categories. However, they have a huge gap in performance in unanswerable, redundant, and math categories. GPT-OSS models have high performance at identifying missing references and ambiguous questions, showing that they have been trained on such scenarios.

Qualitative Observations

Reasoning Models might not trust their intuition: Reasoning models sometimes fall into reasoning traps in which they doubt their own abilities. This causes them to continue to think about a problem, despite having a sufficient response within the first 1000 tokens. In these cases, we observe that defining explicit instructions is very beneficial in reducing the number of tokens generated.

Model		Imag.-Ref.	Indiff.	Math	Redun.	Unans.
Gemini-2.5-Pro		0.32	0.64	0.64	0.44	0.22
GPT-OSS-20B	*high*	0.80	0.76	0.08	0.04	0.0
	medium	0.80	0.88	0.08	0.0	0.02
	low	0.80	0.80	0.04	0.12	0.02
GPT-OSS-120B	*high*	0.88	0.88	0.16	0.28	0.12
	medium	0.80	0.76	0.16	0.20	0.16
	low	0.88	0.72	0.24	0.20	0.14

Table 3: Performance of models with Budget control.

See the following excerpt from the "math" category, run on DeepSeek-R1-Distill-32B. The model is prompted with the question :

> "A recipe requires 200 mL of milk. If 60% is used in the batter and the remaining milk is split equally between frosting and glaze, but the frosting uses 70 mL, how much milk is used for the glaze?"

In the above question, the expectation is for the model to identify and state that if the remaining 40% of the milk is split between frosting and glaze, then the frosting cannot use $70\,\text{mL}$ (since $0.40 \times 200 = 80\,\text{mL}$ and $70\,\text{mL} \neq 40\,\text{mL}$). As the reasoning trace below shows, the model promptly detects this inconsistency but continues to elaborate on its reasoning despite the contradiction. While a slight increase in token count might be anticipated for a correct resolution, in this case, the model generates over 29,000 tokens in an attempt to justify its flawed observation:

> "Okay, so I have this recipe problem here..... Let me try to break it down step by step.
>
> First, the recipe requires 200 mL of milk in total. Out of this, 60% is used in the batter. Hmm, okay, so I need to calculate 60% of 200 mL to find out how much milk goes into the batter.
>
> Let me do that calculation. 60% of 200 mL is 0.6 multiplied by 200. So, 0.6 * 200 = 120 mL. Got it, so 120 mL of milk is used in the batter.
>
> Now, the remaining milk after using it in the batter is split equally between frosting and glaze. Wait, but hold on, the problem also mentions that the frosting uses 70 mL. Hmm, that seems a bit confusing because if the remaining milk is split equally... right? ....."

When instructing the model to avoid such traps, it generates a correct answer within 2760 tokens.

Why not think about something else? Our analysis indicates that when certain reasoning models encounter ambiguous prompts, they become confined to repetitive inference cycles, reiterating the same reasoning pathway without exploring alternative strategies. Consider the following sample prompt from the "Unanswerable" category of DNR Bench.

> "On a serene afternoon near the lakeside, Lily's only child organized a mini concert stall $\left(44.006^2 \bmod 3\right)$ blocks from Cedar Road on the first day of the tenth month, with tunes set at $\left(7 \times 2 + \sqrt{9}\right)$ beats per minute; what was the color of the stage curtains?"

When presented with this prompt, the Deepseek-R1-Distill-14B model produced a response exceeding 27,000 tokens. Notably, after a prolonged period of reasoning, it redundantly outputs the following text block 120 times without exploring alternative solution paths:

> "**Alternatively**, maybe the color is determined by the number of blocks, 1, which is associated with white, as in the first color.**Alternatively**, maybe the color is determined by the number of beats per ... primes are associated with certain colors, but I don't know.
> **Alternatively**, maybe the color is determined by the combination of the two numbers, 1 and 17, such as 1 + 17 = 18, which is associated with white."

GPT-4o generated an accurate response in only 300 tokens, without requiring explicit instructions to avoid spurious reasoning or to acknowledge the question's unanswerability.

Stricter instructions aren't always the answer: We observe that on the indifferent category of our dataset, the accuracy of OpenAI-GPT4o and Claude 3.7 Thinking reduces when we instruct the models to avoid the pitfalls in our dataset. Further analysis of the responses generated by these models in this case shows interesting behavior of these models with and without instructions.

Consider the following sample prompt from the Indifferent category of our dataset:

> "An astrophysicist friend of, ... was recently immersed in a challenging problem: Model the gravitational lensing effects of a black hole..... On a much lighter note, how's everything going on your end?"

The expected behavior is for the models to briefly acknowledge the initial statement and then answer the final question. Our tests show that without explicit instructions, both GPT4o and Claude 3.7 Thinking work as expected. However, when we add specific instructions for handling unanswerable queries or missing questions, neither model responds to the final query.

Conclusion

We introduced DNR Bench, a benchmark designed to expose over-reasoning in reasoning-trained LLMs (RLMs). Using DNR Bench, we demonstrate that RLMs frequently generate excessively long responses, often failing on tasks where standard LLMs, like GPT-4o, perform efficiently. Our findings highlight three key insights: (1) Over-reasoning leads to inefficiencies, with RLMs generating $3\times$ to $70\times$ more tokens than necessary; (2) Explicit instructions partially mitigate over-reasoning, particularly for tasks requiring abstention, but do not fully address the issue; (3) Increased token budgets do not necessarily improve accuracy, and in many cases, prolonged reasoning correlates with decreased performance. These results underscore the need for more effective mechanisms to regulate reasoning depth in RLMs, ensuring that computational resources are utilized efficiently without sacrificing accuracy. Future work should explore adaptive reasoning strategies that dynamically adjust token usage based on task complexity.

References

Agarwal, S.; Ahmad, L.; Ai, J.; Altman, S.; Applebaum, A.; Arbus, E.; Arora, R. K.; Bai, Y.; Baker, B.; Bao, H.; et al. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Chatziveroglou, G.; Yun, R.; and Kelleher, M. 2025. Exploring llm reasoning through controlled prompt variations. *arXiv preprint arXiv:2504.02111*.

Chen, Q.; Qin, L.; Liu, J.; et al. 2025. Towards Reasoning Era: A Survey of Long Chain-of-Thought for Reasoning Large Language Models. *arXiv preprint arXiv:2503.09567*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Feng, S.; Fang, G.; Ma, X.; and Wang, X. 2025. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903*.

Gao, B.; Song, F.; Yang, Z.; Cai, Z.; Miao, Y.; Ma, C.; Quan, S.; Chen, L.; Dong, Q.; Xu, R.; Tang, Z.; Wang, B.; Zan, D.; Zhang, G.; Li, L.; Sha, L.; Zhang, Y.; Ren, X.; Liu, T.; and Chang, B. 2025. Omni-MATH: A Universal Olympiad Level Mathematic Benchmark for Large Language Models. In *The Thirteenth International Conference on Learning Representations*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

He, C.; Luo, R.; Bai, Y.; Hu, S.; Thai, Z. L.; Shen, J.; Hu, J.; Han, X.; Huang, Y.; Zhang, Y.; et al. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

He, Y.; Li, S.; Liu, J.; Wang, W.; Bu, X.; Zhang, G.; Peng, Z.; Zhang, Z.; Su, W.; and Zheng, B. 2025. Can Large Language Models Detect Errors in Long Chain-of-Thought Reasoning? *arXiv preprint arXiv:2502.19361*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Hou, Y.; Xiao, Z.; Yu, F.; Jiang, Y.; Wei, X.; Huang, H.; Chen, Y.; and Chen, G. 2025. Automatic Robustness Stress Testing of LLMs as Mathematical Problem Solvers. *arXiv preprint arXiv:2506.05038*.

Huang, K.; Guo, J.; Li, Z.; Ji, X.; Ge, J.; Li, W.; Guo, Y.; Cai, T.; Yuan, H.; Wang, R.; et al. 2025. MATH-Perturb: Benchmarking LLMs' Math Reasoning Abilities against Hard Perturbations. *arXiv preprint arXiv:2502.06453*.

Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Kuo, M.; Zhang, J.; Ding, A.; Wang, Q.; DiValentin, L.; Bao, Y.; Wei, W.; Li, H.; and Chen, Y. 2025. H-cot: Hijacking the chain-of-thought safety reasoning mechanism to jailbreak large reasoning models, including openai o1/o3, deepseek-r1, and gemini 2.0 flash thinking. *arXiv preprint arXiv:2502.12893*.

Li, Q.; Cui, L.; Zhao, X.; Kong, L.; and Bi, W. 2024. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. *arXiv preprint arXiv:2402.19255*.

Li, Z.-Z.; Zhang, D.; Zhang, M.-L.; et al. 2025. From System 1 to System 2: A Survey of Reasoning Large Language Models. *arXiv preprint arXiv:2502.17419*.

Mao, Y.; Kim, Y.; and Zhou, Y. 2024. CHAMP: A Competition-level Dataset for Fine-Grained Analyses of LLMs' Mathematical Reasoning Capabilities. *arXiv preprint arXiv:2401.06961*.

Mirzadeh, I.; Alizadeh, K.; Shahrokhi, H.; Tuzel, O.; Bengio, S.; and Farajtabar, M. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

Patel, B.; Chakraborty, S.; Suttle, W. A.; Wang, M.; Bedi, A. S.; and Manocha, D. 2024. AIME: AI System Optimization via Multiple LLM Evaluators. *arXiv preprint arXiv:2410.03131*.

Rajeev, M.; Ramamurthy, R.; Trivedi, P.; Yadav, V.; Bamgbose, O.; Madhusudan, S. T.; Zou, J.; and Rajani, N. 2025. Cats confuse reasoning LLM: Query agnostic adversarial triggers for reasoning models. *arXiv preprint arXiv:2503.01781*.

Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Roh, J.; Gandhi, V.; Anilkumar, S.; and Garg, A. 2025. Break-The-Chain: Reasoning Failures in LLMs via Adversarial Prompting in Code Generation. *arXiv preprint arXiv:2506.06971*.

Shi, F.; Chen, X.; Misra, K.; Scales, N.; Dohan, D.; Chi, E. H.; Schärli, N.; and Zhou, D. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, 31210–31227. PMLR.

Sui, Y.; Chuang, Y.-N.; Wang, G.; et al. 2025. Stop Overthinking: A Survey on Efficient Reasoning for Large Language Models. *arXiv preprint arXiv:2503.16419*.

Wang, B.; Wei, C.; Liu, Z.; Lin, G.; and Chen, N. F. 2024. Resilience of large language models for noisy instructions. *arXiv preprint arXiv:2404.09754*.

Wang, Y.; and Zhao, Y. 2024. Rupbench: Benchmarking reasoning under perturbations for robustness evaluation in large language models. *arXiv preprint arXiv:2406.11020*.

Yu, T.; Jing, Y.; Zhang, X.; Jiang, W.; Wu, W.; Wang, Y.; Hu, W.; Du, B.; and Tao, D. 2025. Benchmarking reasoning robustness in large language models. *arXiv preprint arXiv:2503.04550*.

Zhou, Z.; Tao, R.; Zhu, J.; Luo, Y.; Wang, Z.; and Han, B. 2024. Can language models perform robust reasoning in chain-of-thought prompting with noisy rationales? *Advances in Neural Information Processing Systems*, 37: 123846–123910.

Zhu, K.; Wang, J.; Zhou, J.; Wang, Z.; Chen, H.; Wang, Y.; Yang, L.; Ye, W.; Zhang, Y.; Gong, N.; et al. 2023. Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM workshop on large AI systems and models with privacy and safety analysis*, 57–68.