Thinking Past the Answer: Evaluating Harmful Overthinking in Large Reasoning Models

Simone Caldarella[1,*] **Davide Talon**[3]
Rahaf Aljundi[2] **Elisa Ricci**[1,3] **Massimiliano Mancini**[1]

[1]University of Trento
[2]Toyota Motor Europe
[3]Fondazione Bruno Kessler

Abstract

Large Reasoning Models (LRMs) improve performance by generating explicit intermediate reasoning traces through increased test-time compute, yet the assumption that longer reasoning is consistently beneficial remains under-examined. While recent evidence shows that additional reasoning can lead models to overthink, we ask: "*Once a model has reached the correct answer, does further reasoning refine the solution, or deviate from it?*" To study the dynamics after correctness, we introduce a prefix-level trajectory evaluation protocol grounded in reasoning sufficiency, defining the minimum reasoning budget required for a model to first generate the correct answer. This allows us to disentangle *verbose* overthinking, where additional reasoning is redundant but harmless, from *harmful* overthinking, where continued reasoning destabilizes an already-correct trajectory. Starting from multimodal benchmarks, we find that many instances considered reasoning-intensive require surprisingly little reasoning. Moreover, stopping at the first correct prefix improves accuracy over standard reasoning up to 21%, revealing that current models are limited not only by their ability to reason, but also by their inability to stop at the right time. Furthermore, while common efficiency strategies like early stopping substantially reduce verbose overthinking (up to 50%), they fail to mitigate harmful overthinking. Failure analysis reveals that correctness deviations are mainly driven by logical drift and visual reinterpretation. Finally, we show that our findings generalize to language-only reasoning benchmarks, highlighting harmful overthinking as a broader reliability risk. Code available at https://simonecaldarella.github.io/thinking-past-the-answer.

1 Introduction

Large Reasoning Models (LRMs), such as OpenAI's o1 [11] and DeepSeek's R1 [9], have shown that allocating additional computation at test time can substantially improve performance on challenging tasks.[2] This paradigm, referred to as *test-time scaling* [24], improves performance by allowing models to produce longer and more deliberative reasoning traces, with gains observed in mathematical problems [10, 5], code generation [3, 16], and multimodal reasoning [19, 31]. However, emerging evidence suggests that more reasoning is not always better: LRMs often exhibit systematic *overthinking*, generating reasoning traces substantially longer than necessary to solve a problem [28, 18, 4].

Prior work has largely treated overthinking as an efficiency problem, aiming to reduce reasoning cost while preserving the accuracy of full-length chains of thought (CoT) [26, 44, 17, 15, 39, 33]. This

*Correspondence to: simone.caldarella@unitn.it

[2]Throughout this paper, we use the term *large reasoning models* to refer jointly to language-only and multimodal models trained to generate explicit intermediate reasoning traces.

Preprint.

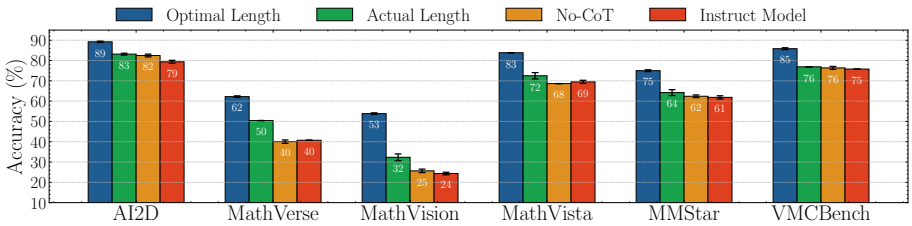

Figure 1: Performance averaged on LRMs. *Actual Length* is the model's default behavior, *No-CoT* disables intermediate reasoning, and *Instruct Model* is the pre-reasoning instruction-tuned model. Finally, *Optimal Length* stops at the first correct prefix. The gap between *Actual Length* and *Optimal Length* shows that models often reason past correctness, making additional reasoning harmful.

perspective has also been studied mostly in language-only settings [6, 34], leaving limited insight into multimodal LRMs, where continued reasoning can introduce visual misreadings or unsupported reinterpretations of the input. In this paper, we argue that this view is incomplete: overthinking is also a reliability problem. A model may reach the correct answer early, continue reasoning, and later revise, contradict, or overwrite that correct solution.

We study this phenomenon through the lens of *reasoning sufficiency*. For a given model and question, we define the question's difficulty as the minimum reasoning budget required for the model to produce the correct answer. This differs from prior work that proxies difficulty using the average length of model-generated traces [28, 26, 15], since trace length can itself be inflated by overthinking. Our formulation isolates the computation minimally required for correctness and separates two forms of overthinking: *verbose overthinking*, where the model reasons beyond the sufficient budget while preserving the correct answer, and *harmful overthinking*, where additional reasoning causes a trajectory that has already reached the correct answer to end with an incorrect final prediction. Under this view, test-time scaling is not monotonically beneficial; additional computation can destabilize an already-correct solution.

To measure these effects, we introduce a *prefix-level* trajectory evaluation protocol. Given a reasoning trace, we evaluate prefix-level performance by forcing the model to produce an answer from that partial trace. This lets us identify when the correct answer first becomes recoverable and whether continued reasoning preserves or loses correctness. Using this protocol, we find that overthinking is substantial and systematic across multimodal benchmarks. Many questions commonly viewed as reasoning-intensive can be solved with surprisingly few reasoning steps, yet models often continue far beyond the sufficient point. As shown in Fig. 1, *Optimal Length*, which stops at the first correct prefix, outperforms the model's default *Actual Length* behavior by nearly 10% on average; this gain exceeds the benefit from reasoning-oriented post-training over the corresponding instruct model. These results suggest that current LRMs are limited not only by whether they can reason, but also by whether they can stop reasoning at the right time.

We further show that harmful overthinking is not tied to a particular answer format or modality. Both multiple-choice and free-form questions exhibit harmful overthinking, with surprisingly stronger effects in the latter settings, where the less constrained output space makes it easier to drift away from a previously correct answer. Language-only experiments show that the same phenomenon also affects unimodal LRMs. Moreover, simply shortening traces is insufficient: early stopping, *i.e.*, terminating the reasoning trace earlier, reduces average reasoning length, but fails to mitigate harmful overthinking. Finally, an analysis of $4{,}842$ harmful traces reveals that correctness deviations are dominated by visual and logical errors, while calculation errors account for only a small fraction.

In summary, **the contributions of this paper** are as follows:

① We formalize overthinking via the minimum sufficient reasoning budget, disentangling verbose overthinking from harmful overthinking.
② We introduce a prefix-level evaluation protocol that measures reasoning sufficiency and correctness instability along model trajectories.
③ We quantify harmful overthinking across multimodal and language-only benchmarks, showing that LRMs often drift from early correct answers to incorrect final predictions.
④ We categorize the sources of harmful overthinking and show that correctness deviations are driven mainly by logical and visual errors rather than arithmetic mistakes.

2 Formalizing Overthinking via Reasoning Sufficiency

In this section, we formalize overthinking through the lens of reasoning sufficiency. We first define question difficulty as the minimum reasoning budget required for a model to reach a correct answer. We then use this notion to distinguish *verbose* overthinking from *harmful* overthinking.

Setting and Notation. Let (x, y) be a sample with input $x \in \mathcal{X}$ (potentially multimodal) and ground-truth answer $y \in \mathcal{Y}$. We consider a large reasoning model as a generative framework $\mathcal{F} : \mathcal{X} \to \mathcal{T}$ that, given x, produces a reasoning trace $t = \mathcal{F}(x) \in \mathcal{T}$ that includes the predicted answer, where $\mathcal{T}$ denotes the space of possible traces. For consistent evaluation in cases where answer formatting is not followed, we rely on a fixed answer extraction protocol where a language model $\mathcal{A} : \mathcal{T} \to \mathcal{Y}$ extracts the prediction from the provided trace $\hat{y} = \mathcal{A}(t)$. The extractor is implemented as a separate model (Qwen3-4B [40]) that operates solely on the generated reasoning trace.

2.1 Problem Difficulty

What does it mean for a problem to be difficult? Prior work often characterizes difficulty [26, 24, 15] using aggregate proxies such as *pass@k* or average chain-of-thought length [26]. These proxies are confounded by decoding policy, sampling strategy, and verbosity, and therefore do not isolate the computation actually required for correctness. We instead define the empirical difficulty of an instance *wrt.* a model as the minimum reasoning budget (*i.e.*, shortest CoT) sufficient for the model to obtain the correct answer. This separates required reasoning from redundant/harmful continuation.

Formally, we consider t as a sequence of N utterances, $t = (u_1, \ldots, u_N)$, where each u_i represents a semantically coherent reasoning step. We denote by $t_{\leq i} = (u_1, \ldots, u_i)$ the prefix up to step i, with $t_{\leq 0} = \emptyset$ corresponding to no intermediate reasoning, *i.e.*, the model can already answer without any reasoning. Each prefix induces a prediction $\hat{y}_i = \mathcal{A}(t_{\leq i})$. We define the *first correct index* as:

$$\tau_y(x; \mathcal{F}) = \underset{i \in \{0,\ldots,N\}}{\arg\min}\ b_i \quad \text{s.t.} \quad \mathcal{A}(t_{\leq i}) = y, \tag{1}$$

where b_i is the computational budget associated with prefix $t_{\leq i}, i = 0, \ldots, N$, and the empirical difficulty of the instance is $\hat{\kappa}(x, y; \mathcal{F}) = b_{\tau_y(x;\mathcal{F})}$. If no prefix yields the correct answer, we set $\tau_y = \infty$[3] and leave $\hat{\kappa}$ undefined for that trajectory. We emphasize that $\hat{\kappa}(x, y; \mathcal{F})$ is not an intrinsic property of the instance alone, but a *model-dependent* difficulty of the sample. This definition, invariant to overall length, captures the minimal computation required for the model to reach a correct answer: once a correct prefix has been reached, extending the reasoning does not change the difficulty. In practice, $\hat{\kappa}$ provides an *empirical* lower bound on the compute required to form the correct answer.

On Tokens *vs.* Utterances. We instantiate the budget b_i as the number of utterances in $t_{\leq i}$. Unlike token count, utterance-level budgets are less sensitive to formatting and verbosity, and better align with semantically coherent reasoning steps. In practice, we instantiate the reasoning steps by splitting traces at explicit delimiters (line breaks), which LRMs tend to use naturally. We use the generic b_i to make clear that the definition is not tied to a particular notion of budget and the same definitions can be applied to token-level steps. Appendix B.4 analyzes statistics on utterances and tokens.

2.2 Disentangling Overthinking

Verbose vs. Harmful. Given the first correct index τ_y, we define *overthinking* as any continuation beyond the first correct prefix. That is, all steps $j > \tau_y$ correspond to computation that is not necessary to first obtain the correct answer. Then, by comparing the trace $t_{\leq \tau_y}$ with the full model one $t_{\leq N}$, we distinguish two cases:

① *Verbose* overthinking corresponds to wasted computation: once the model reaches a correct intermediate state, further reasoning does not change the outcome,

$$\mathcal{A}(t_{\leq \tau_y}) = y \ \ \wedge \ \ \mathcal{A}(t_{\leq N}) = y. \tag{2}$$

Here, additional reasoning is redundant. The model has already solved the problem, but continues to generate unnecessary steps without affecting the final prediction.

[3] Mathematicians may forgive us. In practice, when a trace does not reach the correct solution, we set the optimal length equal to the maximum length.

② *Harmful* overthinking, in contrast, reflects a failure of the reasoning process itself: after reaching a correct answer, additional computation causes the model to deviate from correctness,

$$\mathcal{A}(t_{\leq \tau_y}) = y \ \ \wedge \ \ \mathcal{A}(t_{\leq N}) \neq y. \tag{3}$$

In this case, the model initially reaches the correct solution, but subsequent reasoning introduces errors that override it, making the model reply incorrectly. Rather than refining the answer, additional computation destabilizes an otherwise correct trajectory. Crucially, in Sec. 3.2 we will show that reducing verbose overthinking does not reduce harmful overthinking, highlighting their orthogonality.

Harmful Overthinking as Trajectory Instability. The definition ② treats harmful overthinking as a binary event: after first reaching a correct answer, the model terminates with an incorrect one. To analyze this behavior along the trajectory, we define the correctness state of each prefix as

$$z_i = \mathbf{1}[\mathcal{A}(t_{\leq i}) = y]. \tag{4}$$

Under monotonic reasoning, correctness would be absorbing: once $z_i = 1$, all later states would remain correct. Harmful overthinking corresponds to a violation of this monotonicity.

We therefore define the event-level harmful overthinking indicator as

$$h(x; \mathcal{F}) = \mathbb{1}\left[\tau_y < \infty \ \wedge \ z_N = 0\right]. \tag{5}$$

Thus, h captures whether the model reaches a correct prefix but loses correctness by termination. For a dataset $\mathcal{D}$, we report the harmful overthinking rate as the average of this indicator:

$$H(\mathcal{D}; \mathcal{F}) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} h(x; \mathcal{F}). \tag{6}$$

Sec. 3 further analyzes reasoning trajectory through the probability of remaining correct after τ_y.

3 Overthinking in Large Reasoning Models

We now define the main experimental protocol and investigate how reasoning unfolds in practice, relative to the minimum reasoning budget. Our analysis is guided by three questions: (i) how much reasoning is actually required to solve benchmark questions, (ii) what happens when models reason beyond this point, and (iii) whether reducing reasoning length mitigates potential failures.

We begin by examining how correct solutions first emerge along the reasoning trajectory, with a focus on the challenging multimodal setting. Building on this perspective, we then study harmful overthinking, focusing on how additional reasoning can affect correctness. We further analyze how this phenomenon depends on the answer format, contrasting multiple-choice and free-form generation. To better understand these effects, we adopt a prefix-level trajectory view and study correctness transitions across reasoning steps, revealing the underlying dynamics of reasoning. Finally, we evaluate whether reducing verbosity is sufficient to improve reliability, and assess the generality of these behaviors by extending the analysis to language-only models.

3.1 Experimental Setting

Models and Benchmarks. Building on prior work on overthinking [39, 15], we analyze recent LRMs for multimodal reasoning: MM-Eureka [22], R1-VL [45], ThinkLite-VL [36], and VL-Rethinker [30]. We evaluate these models on a diverse set of multimodal benchmarks spanning diagram understanding, visual grounding, mathematical reasoning, and multiple-choice vision-language QA: AI2D [12], MathVista [19], MathVision [31], MathVerse [47], MMStar [2], and VMCBench [50]. For language-only reasoning instead, we consider Qwen3 [40] and InternS1 [1] on AIME2025 [49] and GPQA [25].

Reasoning Strategies. We evaluate four strategies spanning lower and upper bounds on reasoning performance. Instruct Model is the base *Instruction-Tuned* model before reasoning-oriented post-training [48]. *No-CoT* forces the reasoning model to answer immediately, without intermediate reasoning. *Actual Length* is the model's default unconstrained CoT behavior. *Optimal Length* is an oracle strategy that stops at the first correct prefix $t_{\leq \tau_y}$. Since identifying this prefix requires ground-truth access, it is not deployable; rather, it quantifies the gain achievable by eliminating harmful overthinking.

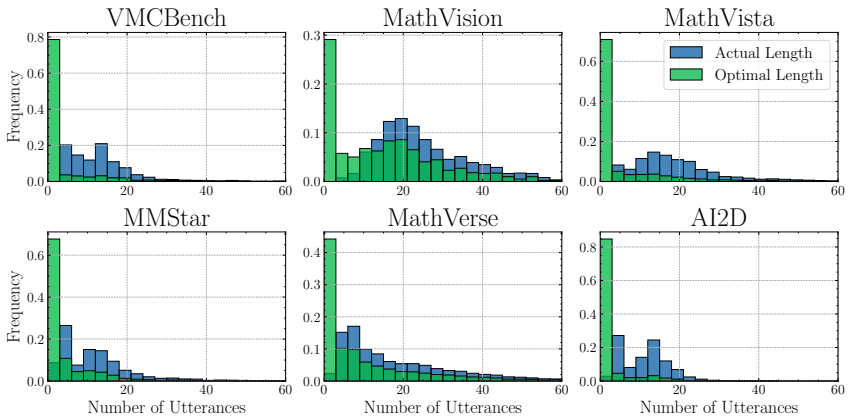

Figure 2: Average number of utterances across five multimodal models under Actual Length and Optimal Length. Even on benchmarks typically considered challenging (*e.g.*, Mathvision [31]) most solvable instances require little to no intermediate reasoning.

Prefix-Level Trajectory Evaluation. Inspired by previous work [8, 24], we probe intermediate reasoning states by evaluating every utterance-level prefix $t_{\leq i}$ of a generated trace, including the empty prefix $t_{\leq 0}$. As reasoning models usually emit answers only at termination, we append a fixed termination template to each prefix[4], thus forcing the model to provide an answer at intermediate steps. This intervention lets us track correctness across the reasoning trajectory by testing whether each *partial* trace is sufficient to generate a correct answer. See Appendix A and C.1 for more details.

3.2 Results

How much reasoning is actually required? We first examine where the optimal stopping point occurs along the reasoning trajectory. Fig. 2 compares, for solved instances, the model's actual reasoning length with the optimal length required to first reach the correct answer. Across benchmarks, optimal lengths are concentrated near the beginning of the trajectory, often at zero utterances, indicating the model can answer correctly without generating an explicit chain of thought. This is also confirmed by the performance that *No-CoT* achieves across benchmarks (see Fig. 1). On the contrary, actual traces extend substantially further. Notably, even on more challenging datasets such as MathVision and MathVerse, where traces are longer than on AI2D or VMCBench, the optimal length remains far below the model's default reasoning length.

> TAKEAWAY A. Reasoning length is a poor proxy for difficulty: LRMs often solve the problem early, then keep generating long traces that are not required for correctness.

Reasoning beyond optimal. We next quantify the effect of reasoning beyond the first correct step $t_{\leq \tau_y}$. From Tab. 1, *Optimal Length* consistently outperforms *Actual Length* across all models and benchmarks (*e.g.*, +23.3% of R1-VL on MathVision and +7.8% of VL-Rethinker on AI2D). The largest gaps occur on harder, lower-accuracy benchmarks such as MathVision and MathVerse. This gap is not merely an efficiency loss. In many cases, the model has already reached the correct answer, but later reasoning causes it to deviate from the correct answer. Together with Fig. 1, these results show that allocating the right amount of reasoning is often more important than simply enabling reasoning: the gap between *Optimal Length* and *Actual Length* is larger than the gain from reasoning-oriented post-training itself. See Appendix B.5 for analysis of verbose overthinking.

> TAKEAWAY B. Current LRMs do not merely over-generate reasoning; instead, they frequently reason past correct intermediate states, making optimal stopping substantially more valuable than additional reasoning.

Multiple-choice *vs.* Free-form. We next ask whether harmful overthinking depends on the answer format. Fig. 3 compares multiple-choice (MC) and free-form (FF) questions, aggregated across

[4] "`Oh, I suddenly got the answer to the whole problem. <answer> \n\n ### Final Answer: [boxed{.`"

Table 1: Main multimodal results. We report accuracy (acc ↑), average utterance length (len ↓), and harmful-overthinking rate (H ↓). *No-CoT* is a zero-reasoning diagnostic; Bolding highlights the best nontrivial reasoning strategy. The gap between *Actual* and *Optimal* shows that LRMs often reason past correctness and degrade final performance.

Model	Strategy	VMCBench			MathVision			Mathvista			MMStar			MathVerse			AI2D		
		acc↑	*len*↓	H↓	*acc*↑	*len*↓	H↓	*acc*↑	*len*↓	H↓	*acc*↑	*len*↓	H↓	*acc*↑	*len*↓	H↓	*acc*↑	*len*↓	H↓
DualMind-VLM	No-CoT	79.8	0.0	0.0	24.0	0.0	0.0	69.5	0.0	0.0	62.3	0.0	0.0	40.1	0.0	0.0	82.9	0.0	0.0
	Actual	80.9	5.6	4.4	26.3	18.0	21.1	74.7	11.2	6.6	64.4	6.1	7.1	49.7	19.5	11.3	83.3	3.7	3.9
	Optimal	**85.3**	**1.8**	**0.0**	**47.4**	**11.6**	**0.0**	**81.3**	**3.9**	**0.0**	**71.5**	**2.2**	**0.0**	**60.9**	**11.3**	**0.0**	**87.2**	**0.6**	**0.0**
MM-Eureka	No-CoT	75.8	0.0	0.0	25.3	0.0	0.0	67.9	0.0	0.0	60.3	0.0	0.0	38.6	0.0	0.0	82.1	0.0	0.0
	Actual	76.4	19.6	9.6	32.9	34.0	13.5	72.8	20.0	9.6	64.0	13.1	7.6	48.2	11.9	11.3	82.8	8.6	5.0
	Optimal	**86.0**	**6.2**	**0.0**	**46.4**	**20.1**	**0.0**	**82.4**	**7.5**	**0.0**	**71.6**	**4.6**	**0.0**	**59.5**	**6.8**	**0.0**	**87.8**	**1.5**	**0.0**
ThinkLite-VL	No-CoT	75.7	0.0	0.0	21.4	0.0	0.0	65.5	0.0	0.0	64.1	0.0	0.0	40.1	0.0	0.0	82.7	0.0	0.0
	Actual	75.1	11.8	9.2	28.3	28.0	26.3	70.4	19.3	11.0	65.6	10.9	11.1	49.7	23.0	13.4	83.2	9.9	6.0
	Optimal	**84.3**	**3.2**	**0.0**	**54.6**	**14.9**	**0.0**	**81.4**	**6.3**	**0.0**	**76.7**	**3.6**	**0.0**	**63.0**	**12.2**	**0.0**	**89.2**	**1.5**	**0.0**
VL-Rethinker	No-CoT	77.2	0.0	0.0	28.0	0.0	0.0	70.5	0.0	0.0	62.7	0.0	0.0	40.5	0.0	0.0	82.4	0.0	0.0
	Actual	79.2	19.8	7.8	33.9	36.3	24.7	73.0	26.2	11.9	63.0	17.2	13.7	51.2	29.9	12.1	83.6	15.0	7.1
	Optimal	**87.0**	**4.4**	**0.0**	**58.6**	**19.5**	**0.0**	**84.9**	**6.1**	**0.0**	**76.7**	**5.1**	**0.0**	**63.3**	**15.2**	**0.0**	**90.7**	**1.9**	**0.0**
R1-VL	No-CoT	70.1	0.0	0.0	26.0	0.0	0.0	52.9	0.0	0.0	54.5	0.0	0.0	26.2	0.0	0.0	79.6	0.0	0.0
	Actual	71.1	19.8	8.8	26.6	45.0	23.4	62.2	38.9	14.2	58.5	15.4	12.3	52.8	46.4	15.7	80.3	11.8	7.8
	Optimal	**79.9**	**10.3**	**0.0**	**50.0**	**25.1**	**0.0**	**76.4**	**14.1**	**0.0**	**70.8**	**5.4**	**0.0**	**68.5**	**25.0**	**0.0**	**88.1**	**2.0**	**0.0**

benchmarks. Harmful overthinking is substantially higher in free-form (.11 for MC *vs.* .24 for FF), suggesting that earlier correctness and later deviations are not byproducts of a restricted answer space, but rather the opposite. If correctness deviations were primarily random answer fluctuations, one would expect multiple-choice tasks to exhibit higher, or at least comparable, earlier correctness and later answer instability. Surprisingly, we observe the opposite pattern. This suggest that (i) earlier correct answers are not byproduct of randomness and (ii) correctness is less stable when the setting involves verification (MC) rather than exploration (FF), making correctness in FF setting more vulnerable to unsupported revisions, reinterpretations, and reasoning drift.

> TAKEAWAY C. Free-form generation exposes harmful overthinking more sharply: without a fixed answer set, the unconstrained reasoning is more likely to deviate from correctness.

Reasoning Dynamics. The previous results measure whether harmful overthinking occurs. We now examine how it occurs by tracking correctness along the reasoning trajectory. At each prefix $t_{\leq i}$, the model is either correct or incorrect, inducing a binary state $z_i = \mathbf{1}[\mathcal{A}(t_{\leq i}) = y]$. If reasoning were monotonic, then reaching a correct state would be absorbing: once $z_i = 1$, subsequent prefixes would remain correct. Instead, Fig. 4 shows that correctness is unstable under continued generation. After the first correct prefix τ_y, the probability of remaining correct drops rapidly as additional reasoning steps are generated, plateauing around 0.2 after roughly 100 intermediate steps. This confirms the trajectory-instability view: reasoning does not simply accumulate evidence toward correctness, but can move the model both toward and away from the correct answer.

> TAKEAWAY D. Reasoning trajectories are non-monotonic: after reaching correctness, the probability of staying correct drops rapidly as LRMs keep reasoning.

Can reducing verbosity mitigate harmful overthinking? A natural hypothesis is that harmful overthinking is simply a consequence of verbosity: if a model reasons less, it should have fewer opportunities to leave a correct trajectory. We test this hypothesis with two forms of adaptive inference. First, we consider a training-free early-stopping baseline, applied to each model, inspired by prior work [8]: after each prefix, we extract the current answer and stop when the prediction remains unchanged for K consecutive steps (*i.e.*, Stopping@K). The setting $K = \infty$ recovers the model's default behavior, *i.e.*, *Actual Length*. Second,

Table 2: Early stopping and efficient reasoning reduce verbosity, but do not consistently reduce harmful overthinking.

Setting	Acc ↑	Len ↓	H ↓
Stopping@∞	**66.5**	17.2	**10.1**
Stopping@5	64.3	8.9	18.4
Stopping@2	63.2	**5.1**	14.1
VL-Rethinker	**66.6**	23.4	11.1
DualMind-VLM	66.3	**11.2**	**8.6**

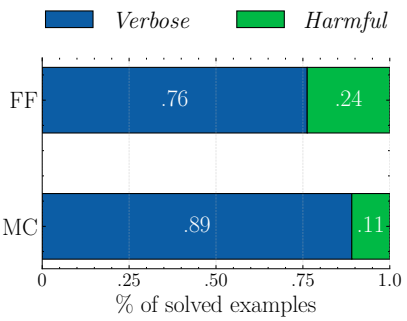

Figure 3: Distribution of overthinking types across response formats. Bars show the percentage of solved samples exhibiting verbose versus harmful overthinking for multiple-choice (MC) and free-form (FF) settings.

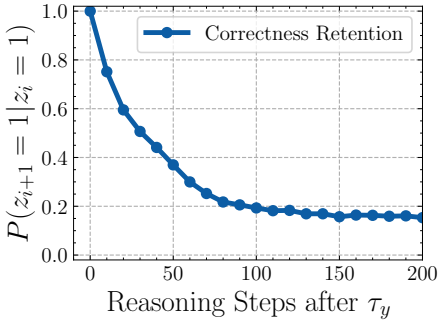

Figure 4: Correctness stability. After first reaching a correct answer at τ_y, the probability of remaining correct decreases sharply with additional reasoning, revealing diminishing reasoning value.

we compare VL-Rethinker [30], trained to encourage thinking, against DualMind-VLM [15], which is explicitly trained to select whether to use reasoning or not.

Tab. 2 shows that both approaches substantially reduce reasoning length. Early stopping with smaller patience values cuts the average length from 17.2 utterances at $K = \infty$ to 8.9 for $K = 5$ and 5.1 for $K = 2$. Similarly, DualMind-VLM produces much shorter traces than VL-Rethinker (11.2 vs. 23.4 utterances) while maintaining comparable accuracy. However, this reduction in verbosity does not translate into a corresponding reduction in harmful overthinking. In fact, early stopping increases the harmful overthinking rate from 10.1 at $K = \infty$ to 18.4 at $K = 5$ and 14.1 at $K = 2$, while DualMind-VLM still exhibits non-negligible harmful transitions despite its shorter traces. These results show that verbose and harmful overthinking are distinct failure modes. Reducing the amount of generated reasoning can remove wasted computation, but it does not necessarily make the remaining trajectory more stable. In some cases, aggressive stopping may even truncate useful recovery dynamics while leaving correctness deviations unresolved.

> TAKEAWAY E. Efficiency-oriented methods address verbose reasoning, but not correctness instability. *Harmful* overthinking must therefore be measured separately from *verbose* one.

Language-Only Reasoning. Finally, we verify that the pattern is not specific to multimodal reasoning. Tab. 3 shows the same qualitative behavior for language-only LRMs on GPQA and AIME2025: default reasoning improves over *No-CoT*, but *Optimal Length* yields much larger gains. For Qwen3, optimal stopping improves accuracy from 55.8 to 77.9 on GPQA and from 58.3 to 91.7 on AIME2025. Similarly, InternS1 improves from 64.4 to 84.7 on GPQA and from 38.9 to 72.2 on AIME2025. These gains coincide with large reductions in reasoning length. For example, Qwen3 on AIME2025 drops from 372.5 to 29.9 utterances under *Optimal Length*. Thus, our experiments show that harmful overthinking is not an artifact of visual grounding but reflects a broader instability of the reasoning process. See Appendix B.6 for more results on the language-only setup.

Table 3: Language-only reasoning results. We report accuracy (acc ↑), average utterance length (len ↓), and harmful-overthinking rate (H ↓). *No-CoT* is a zero-reasoning diagnostic; bolding highlights the best nontrivial reasoning strategy. The pattern also holds for language-only models.

Model	Strategy	GPQA			AIME2025		
		acc↑	*len*↓	H ↓	*acc*↑	*len*↓	H ↓
Qwen3	No-CoT	37.0	0.0	0.0	25.0	0.0	0.0
	Actual	55.8	125.5	22.1	58.3	372.5	33.3
	Optimal	**77.9**	**28.9**	**0.0**	**91.7**	**29.9**	**0.0**
InternS1	No-CoT	37.3	0.0	0.0	11.1	0.0	0.0
	Actual	64.4	177.1	20.3	38.9	514.2	33.3
	Optimal	**84.7**	**30.9**	**0.0**	**72.2**	**144.9**	**0.0**

> TAKEAWAY F. Harmful overthinking is not merely a byproduct of visual drift or instability in multimodal reasoning: similar patterns also appear in language-only models, even on math-heavy complex benchmarks.

Table 4: Failure-mode distribution by model and benchmark. Each triplet reports the percentage of valid harmful overthinking traces assigned to visual, calculation, or logical errors (highest in bold).

Model	VMCBench			MathVision			MathVista			MMStar			MathVerse			AI2D		
	V	*C*	*L*	*V*	*C*	*L*	*V*	*C*	*L*	*V*	*C*	*L*	*V*	*C*	*L*	*V*	*C*	*L*
DualMind-VLM	**50.0**	16.7	33.3	**45.6**	14.0	40.4	38.7	16.1	**45.2**	**46.5**	9.9	43.6	27.9	20.6	**51.5**	**53.3**	1.7	45.0
MM-Eureka	45.0	7.5	**47.5**	**50.0**	10.5	39.5	45.6	7.6	**46.8**	**46.8**	9.0	44.1	28.4	14.3	**57.3**	49.0	0.6	**50.3**
ThinkLite-VL	**47.4**	5.3	**47.4**	**73.8**	4.8	21.4	**59.1**	6.8	34.1	**63.8**	2.9	33.3	38.4	12.4	**49.2**	**64.5**	0.0	35.5
VL-Rethinker	**46.7**	8.0	45.3	41.5	9.2	**49.2**	39.8	14.2	**46.0**	**50.5**	3.5	46.0	25.3	16.1	**58.6**	45.3	0.9	**53.7**
R1-VL	**46.9**	12.5	40.6	37.2	9.3	**53.5**	33.3	19.0	**47.6**	**52.2**	3.3	44.4	18.0	11.1	**70.9**	**51.7**	0.7	47.7

4 Why Does Reasoning Become Harmful?

The previous section shows that harmful overthinking is a systematic failure mode. *But what causes a model to transition from a correct answer to an incorrect one?* In the following, we consider the multimodal setting and categorize the type of errors arising when models reason beyond the optimal.

Taxonomy. For each harmful overthinking trajectory, we now identify the last correct prefix $i^\star = \max\{i < N : \mathcal{A}(t_{\leq i}) = y\}$ and compare the reasoning state at $t_{\leq i^\star}$ with the final trace $t_{\leq N}$. This isolates the segment of reasoning that turns a correct trajectory into an incorrect one. We identify three main failure modes:

① *Visual Error.* The model introduces an error by misreading, inventing, or over-interpreting visual evidence. This includes incorrect object recognition, counts, spatial relations, labels, diagram structure, or geometric interpretation.

② *Calculation error.* The model perceives and approaches the problem correctly, but introduces an arithmetic, algebraic, unit-conversion, formula-selection, or numerical-computation error.

③ *Logical error.* The model changes its answer due to a non-visual and non-numerical reasoning failure. This includes unsupported conclusions, contradictions, irrelevant detours, answer-option mismatches, or answer revisions that are not justified by new visual or computational evidence.

Evaluation Protocol. We perform this analysis on harmful overthinking cases from the same multimodal reasoning models considered in Sec. 3, as well as the same benchmarks. For each harmful trajectory, we construct a pair consisting of the last correct prefix and the final incorrect trace. We then use an external judge model (Qwen3.6-35B) to label the dominant failure mode for each harmful trajectory and provide an *evidence* of the error from the original trace. Additional implementation details, including prompt templates and parsing rules, are provided in Appendix C.2.

4.1 Results

Quantitative. Table 4 reports the failure-mode decomposition for harmful overthinking across models and benchmarks. Calculation errors are rarely dominant: they are never the largest failure mode for any model–benchmark pair, and often remain below 10%. Instead, harmful overthinking is primarily driven by logical drift and visual reinterpretation. Logical errors are especially prominent on MathVerse and MathVista: on MathVerse, they exceed 50% for four out of five models and reach 70.9% for R1-VL; on MathVista, they are the leading failure mode for four out of five models. Visual errors dominate more strongly on visually grounded benchmarks such as MathVision, MMStar, and AI2D. For example, ThinkLite-VL reaches 73.8% visual errors on MathVision and 64.5% on AI2D, while visual errors are also the leading category for most models on MMStar. Thus, the table suggests two recurring mechanisms behind correctness deviation: logical drift on more abstract reasoning benchmarks, and visual reinterpretation on perception-heavy ones.

> TAKEAWAY G. Correctness deviations are mainly driven by logical drift and visual reinterpretation rather than arithmetic mistakes.

Qualitative. We show representative examples of each failure mode in Fig. 5. In the visual-error case, the model first reaches the correct count, $\hat{y}_{i^*} = 6$, but later changes its answer to $\hat{y}_{t_{\leq N}} = 5$ after introducing the false observation that "on the right side, there are 2 bricks missing." The subsequent arithmetic is consistent, but the visual premise is wrong. In the calculation-error case, the model first gives the correct answer, $\hat{y}_{i^*} = 65°$, but later outputs $\hat{y}_{t_{\leq N}} = 61°$. The added reasoning contains a

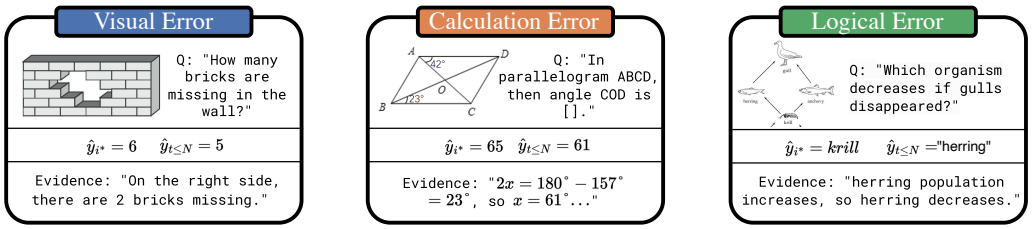

Figure 5: Representative correctness deviations. Each example shows a trajectory that first reaches the correct answer at τ_y, but later changes to an incorrect final answer t_N through perception, calculation, or logical error. Below an evidence, representing the mistaken step of the reasoning model.

direct numerical error: $2x = 180° - 157° = 23°, \quad$ so $x = 61°$. Here the failure is not perceptual, but arithmetic introduced during the continuation. In the logical-error case, the model first correctly answers that krill would decrease if gulls disappeared, but later changes the answer to herring. This contradicts its own explanation, which states that herring would increase. The final answer is therefore unsupported by the model's causal reasoning. The resulting picture is that harmful overthinking is not a single error type, but different failure modes contribute to corrupting an already-correct trajectory.

5 Related Work

Test-Time Scaling and Reasoning. Recent reasoning models derive much of their performance from *test-time scaling*: allocating more inference-time compute via longer chains of thought or larger reasoning budgets often improves accuracy [24, 1, 42, 9]. Similar trends hold in multimodal settings, where structured deliberative traces further boost performance [23, 46, 30, 37]. This line of work largely focuses on average gains from increased compute. In contrast, we study when additional reasoning is unnecessary or harmful, and when longer traces degrade rather than improve predictions.

Adaptive Thinking and Early Exit. Recent work shows that reasoning models often continue generating after reaching a correct solution, and may even revise correct intermediate states into incorrect answers [4]. Early-exit methods stop generation using intermediate predictions, confidence, or learned signals [43, 41, 7, 8], while adaptive-thinking methods allocate variable reasoning budgets across examples using proxies such as response length or confidence [26, 44, 17, 29, 15, 39, 33]. Both primarily target unnecessary computation. Our perspective is complementary: we separate *verbose* overthinking, which is wasteful but harmless, from *harmful* overthinking, which degrades correctness, showing that efficiency alone does not address reasoning failures. Closest to our work, [38] shows that longer CoTs do not consistently improve performance; we extend this analysis to state-of-the-art reasoning models across language and multimodal benchmarks.

Reasoning Compression and No-Thinking Settings. A related line of work questions how much explicit reasoning is required. Prior studies show that reasoning traces can often be compressed, and in some cases removed entirely without loss in performance [13, 20, 14, 35]. Our findings align with this view: the key issue is not whether models can reason longer, but whether additional reasoning is useful, redundant, or harmful.

6 Conclusion

Test-time scaling rests on a simple premise: think longer, and performance should improve. Our results show that this premise is incomplete, offering insights on the overlooked problem of harmful overthinking. Across multimodal and language-only benchmarks, LRMs often reach the correct answer before termination, continue generating, and then leave the correct trajectory. We show that optimal stopping yields large gains, many solvable instances require little or no explicit reasoning, and shorter traces fail to reduce harmful transitions. Failure analysis shows that these errors rarely stem from arithmetic limitations; they more often arise from logical drift or visual reinterpretation. We believe these experimental results can stimulate future work on LRMs, focusing not only on making models reason more, but also on helping them *understand when reasoning is sufficient*.

Acknowledgments and Disclosure of Funding

The authors acknowledge the CINECA award under the ISCRA initiative for the availability of high performance computing resources and support. This work was supported by the EU Horizon ELIAS (No. 101120237), ELLIOT (No. 101214398), and TURING (No. 101215032) projects.

References

[1] Lei Bai, Zhongrui Cai, Yuhang Cao, Maosong Cao, Weihan Cao, Chiyu Chen, Haojiong Chen, Kai Chen, Pengcheng Chen, Ying Chen, et al. Intern-s1: A scientific multimodal foundation model. *arXiv:2508.15763*, 2025.

[2] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *NeurIPS*, 2024.

[3] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv:2107.03374*, 2021.

[4] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *ICML*, 2025.

[5] 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:2110.14168*, 2021.

[6] Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv:2502.08235*, 2025.

[7] Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models. *NeurIPS*, 2025.

[8] Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Yonghao Zhuang, Yian Ma, Aurick Qiao, Tajana Rosing, Ion Stoica, et al. Efficiently scaling llm reasoning with certaindex. *NeurIPS*, 2025.

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

[10] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

[11] 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:2412.16720*, 2024.

[12] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016.

[13] Juncai Li, Ru Li, Yuxiang Zhou, Boxiang Ma, and Jeff Z Pan. Chain of thought compression: A theoritical analysis. *arXiv preprint arXiv:2601.21576*, 2026.

[14] Ming Li, Jike Zhong, Shitian Zhao, Yuxiang Lai, Haoquan Zhang, Wang Bill Zhu, and Kaipeng Zhang. To think or not to think: A study of thinking in rule-based visual reinforcement fine-tuning. In *NeurIPS*, 2025.

[15] Chenyu Lin, Cheng Chi, Jinlin Wu, Sharon Li, and Kaiyang Zhou. Learning to think fast and slow for visual language models. *arXiv:2511.16670*, 2025.

[16] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *NeurIPS*, 2023.

[17] Wanlong Liu, Junxiao Xu, Fei Yu, Yukang Lin, Ke Ji, Wenyu Chen, Yan Xu, Yasheng Wang, Lifeng Shang, and Benyou Wang. Qfft, question-free fine-tuning for adaptive reasoning. *NeurIPS*, 2025.

[18] Yue Liu, Jiaying Wu, Yufei He, Ruihan Gong, Jun Xia, Liang Li, Hongcheng Gao, Hongyu Chen, Baolong Bi, Jiaheng Zhang, et al. Efficient inference for large reasoning models: A survey. *arXiv:2503.23077*, 2025.

[19] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *ICLR*, 2023.

[20] Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning models can be effective without thinking. *arXiv:2504.09858*, 2025.

[21] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 1975.

[22] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv:2503.07365*, 2025.

[23] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, Ping Luo, Yu Qiao, Qiaosheng Zhang, and Wenqi Shao. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv:2503.07365*, 2025.

[24] 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 *EMNLP*, 2025.

[25] David Rein, Betty Hou, Amos Stock, William Liu, Ayan Mandlekar, Arian Ghodsi, Dara Bahri, Fan Zhou, Akshay Mehra, Eunice Yiu, et al. Gpqa: A graduate-level google-proof q&a benchmark. *COLM*, 2023.

[26] Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. In *EMNLP*, pages 2322–2331, 2025.

[27] Charles Spearman. The proof and measurement of association between two things. 1961.

[28] Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv:2503.16419*, 2025.

[29] Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in llms. In *Findings-ACL 2025*, 2025.

[30] Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *NeurIPS*, 2025.

[31] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *NeurIPS*, 2024.

[32] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv:2409.12191*, 2024.

[33] Xinglin Wang, Shaoxiong Feng, Yiwei Li, Peiwen Yuan, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. Make every penny count: Difficulty-adaptive self-consistency for cost-efficient reasoning. In *Findings-NAACL*, 2025.

[34] Xinpeng Wang, Nitish Joshi, Barbara Plank, Rico Angell, and He He. Is it thinking or cheating? detecting implicit reward hacking by measuring reasoning effort. In *ICLR*, 2026.

[35] Xinpeng Wang, Nitish Joshi, Barbara Plank, Rico Angell, and He He. Is it thinking or cheating? detecting implicit reward hacking by measuring reasoning effort. In *ICLR*, 2026.

[36] Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. In *NeurIPS*, 2025.

[37] Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. *NeurIPS*, 2025.

[38] Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms. *ICLR*, 2026.

[39] Wenyi Xiao and Leilei Gan. Fast-slow thinking GRPO for large vision-language model reasoning. In *NeurIPS*, 2025.

[40] 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:2505.09388*, 2025.

[41] Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. *ICLR*, 2026.

[42] Shiming Yang, Yuxuan Tong, Xinyao Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning. In *ICML*, 2025.

[43] Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. *COLM*, 2025.

[44] Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. In *EMNLP*, 2025.

[45] Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. In *ICCV*, 2025.

[46] Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *ICCV*, 2025.

[47] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *ECCV*, 2024.

[48] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Guoyin Wang, et al. Instruction tuning for large language models: A survey. *ACM*, 2026.

[49] Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025. https://huggingface.co/datasets/math-ai/aime25, 2025.

[50] Yuhui Zhang, Yuchang Su, Yiming Liu, Xiaohan Wang, James Burgess, Elaine Sui, Chenyu Wang, Josiah Aklilu, Alejandro Lozano, Anjiang Wei, et al. Automated generation of challenging multiple-choice questions for vision language model evaluation. In *CVPR*, 2025.

Supplementary Material Overview

This appendix is organized in four macro blocks complementing the discussion in the main paper. First, Appendix A provides robustness analyses for the proposed difficulty-based reasoning budget, testing the sensitivity of the first correct index estimation to sampling seeds, termination prompts, and answer extraction models. Second, in Appendix B we provide additional quantitative results, including token-level budget statistics, verbose overthinking analysis, and language-only evaluations. Third, Appendix C reports implementation and reproducibility details, such as prompt templates, parsing rules, compute accounting, and the failure-analysis categorization. Finally, Appendix D discusses the limitations and possible future work.

A Robustness Study for Difficulty-Based Reasoning Budgets

The prefix-level trajectory evaluation protocol estimates an example's difficulty by identifying the earliest point in a model's reasoning trace from which the correct answer can be recovered. In this section, we test whether that estimate is robust to procedural choices. In particular, we measure sensitivity to three factors: the sampling seed used to generate the trace, the termination prompt used for prefix-level probing, and the answer-extraction model used to parse the answer.

Model and benchmark. We run the robustness study with VL-Rethinker on MathVision. For each condition, the model first generates a full reasoning trace for every benchmark example. We use three random seeds to measure sensitivity to stochastic generation. Raw generations are saved before answer extraction so that answer parsing can be repeated independently with different extraction models.

Termination prompt variants. The difficulty pipeline probes partial reasoning traces by appending a termination prompt that asks the model to stop deliberating and provide a final answer. We compare two variants, shown in Fig. 14: the default prompt used in our pipeline and a reworded version with the same intent and similar length. This tests whether the estimated difficulty is sensitive to a specific stop-and-answer phrase rather than reflecting the content of the reasoning trace.

Answer extraction variants. Because benchmark accuracy is computed from parsed final answers, we also vary the answer-extraction model $\mathcal{A}$. We compare `Qwen/Qwen3-4B-Instruct-2507` and `Qwen/Qwen3.5-4B` and report the answer extraction prompt in Fig. 15. These models are used only after generation: first to parse the full CoT outputs, and later to parse the intermediate answers. This separation avoids loading the extractor during expensive VL-Rethinker generation runs and isolates parser-induced variance from reasoning-model variance.

Experimental design. The study uses three seeds, two termination prompts, and two answer extractors. For each condition, we generate raw traces, apply the corresponding answer extractor, run prefix-level difficulty probing, and evaluate correctness at each probed prefix.

Correlation analysis. To assess robustness, we compute pairwise agreement between conditions over the vector of first-correct budgets $\{b_{\tau_y}\}$ using Spearman Correlation [27]. High agreement across seeds indicates that the difficulty estimate is not dominated by sampling noise. High agreement across termination prompts indicates that the probing method is not overly sensitive to the exact stop-and-answer phrasing. High agreement across answer extractors indicates that the signal is not primarily an artifact of the parser. We also report the correlation of the different runs on the answer extracted at the actual length $\{z_N\}$ based on the Matthews Correlation Coefficient (MCC) [21]. High agreement scores indicate that the considered conditions tend to have the same final answer.

Interpretation. The robustness results in Fig. 6 show that the difficulty-based budget estimate is highly stable across the considered procedural variations. The Spearman correlations for the estimated optimal budget remain consistently high across all comparison groups, indicating that examples are ranked similarly by difficulty even when changing the seed, termination prompt, or answer extractor. Varying only the random seed yields high agreement, while changing the termination prompt introduces the largest drop.

A similarly stable pattern is observed for final-answer correctness at the actual reasoning length. MCC values remain close to one across all conditions, meaning that the same examples tend to be classified as correct or incorrect at the end of the full trace. The slightly lower agreement when

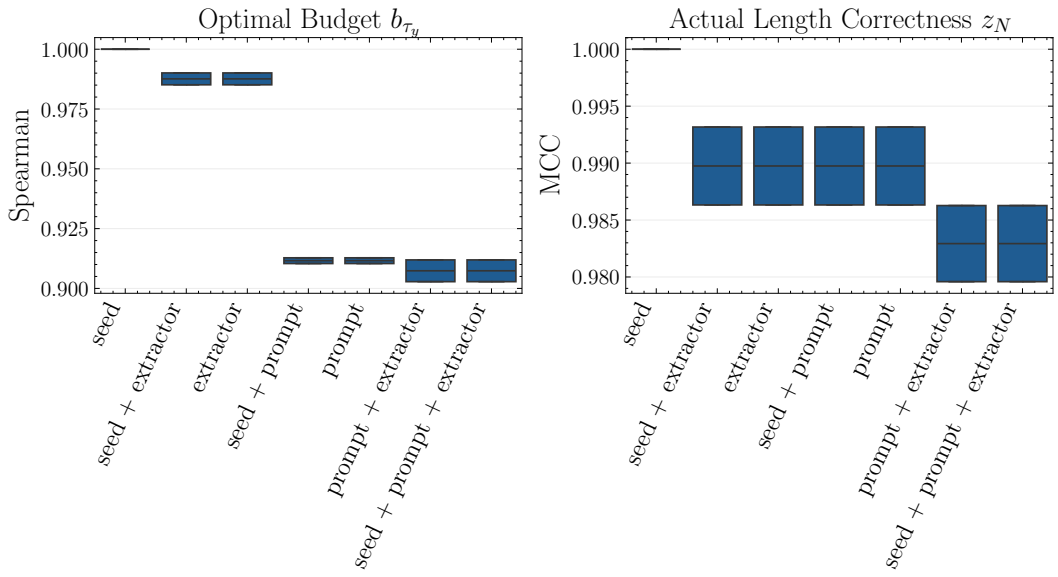

Figure 6: Robustness of the difficulty analysis across controlled sources of variation. The left panel reports Spearman correlation of the estimated optimal budget b_{τ_y}. The right panel reports the Matthews Correlation Coefficient for correctness of the final predicted answer z_N. High optimal-budget correlations indicate that examples are ranked similarly by difficulty across conditions, while high final-correctness MCC indicates that the same examples tend to be correct or incorrect at the actual reasoning length. The comparison groups show impact of joint variation of procedural factors.

both the answer extractor and the termination prompt change indicates that final correctness is more sensitive to parser choice and termination wording, but the effect remains small overall.

Overall, these results support the reliability of the prefix-level trajectory protocol. The estimated first-correct budgets are not artifacts of a particular sampling seed, stop-and-answer prompt, or extraction model. Instead, the high correlations suggest that the measured reasoning sufficiency signal is largely tied to the underlying reasoning trajectory.

B Additional Analysis

We provide additional analyses that complement the main results and further characterize harmful overthinking and the minimum reasoning budget for a model to answer a question.

B.1 Correlation Between Optimal Length and No-CoT Among Models

Fig. 7 studies whether estimated reasoning requirements are consistent across different LRMs. Spearman's correlation on *Optimal Length* is moderately high. This suggests that, despite model-specific differences in how long models reason, they often agree on the number of reasoning steps required to solve a problem. This supports our central claim that reasoning length is a poor proxy for benchmark difficulty: many examples that elicit long traces are nevertheless perceived by several models as solvable with little or no explicit reasoning.

B.2 Optimal Length vs. Test-Time Scaling

Fig. 8 contrasts *Optimal Length* with conventional test-time scaling. The test-time scaling curve, represented by *Actual Length*, improves as additional samples or longer computations are allocated, but remains below the oracle *Optimal Length* strategy, which stops each trajectory at its first correct prefix. This comparison shows that the limitation is not only whether the model can produce the correct answer at some point, but also whether it can preserve that answer until termination. *Pass@K* provides an intermediate diagnostic: the correct answer is often present in the trajectory before the considered average length, but not always at the final utterance, corroborating the findings in Sec. 3.

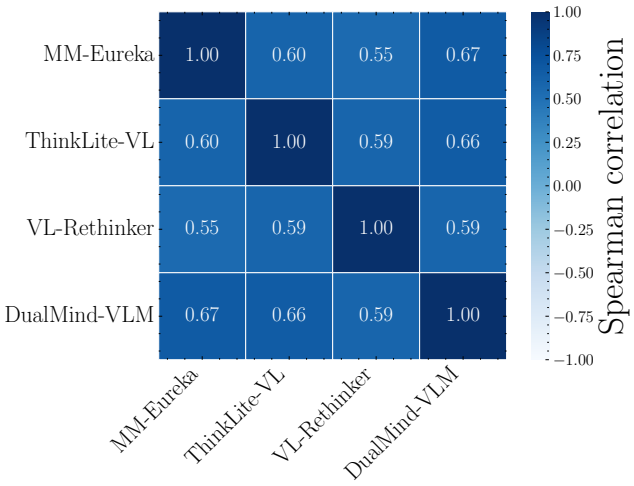

Figure 7: Cross-model Spearman Correlation in estimated *Optimal Length*. Correlation on the exact optimal length is moderately high, indicating that different LRMs share a notion problem difficulty.

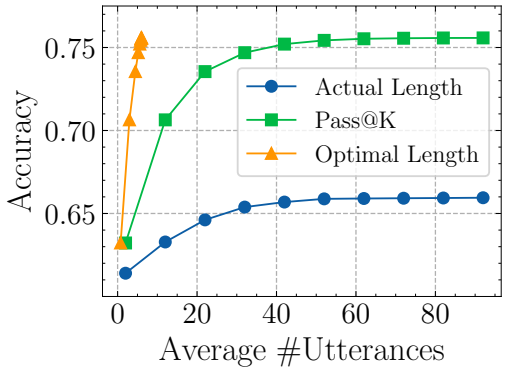

Figure 8: *Optimal Length* scaling compared with standard test-time (*Actual Length*) scaling and *Pass@K*. Increasing test-time compute improves performance, but remains below Optimal Length, which stops each trajectory at the first correct prefix. The gap shows that models often already contain the correct answer before termination, but fail to stop before later reasoning deviates from correctness.

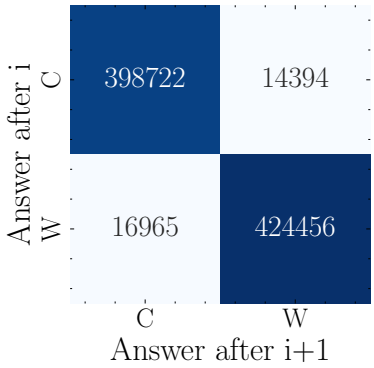

Figure 9: Prefix-level correctness transitions. Rows indicate whether the answer after prefix i is correct or wrong, and columns indicate the answer after prefix $i+1$. The off-diagonal correct-to-wrong mass measures correctness deviations, showing that reasoning is not monotonic once a model has reached the correct answer.

B.3 Transition Matrix of Trajectories

The transition matrix in Fig. 9 highlights the non-monotonic nature of reasoning trajectories moving from $t_{\leq i}$ to $t_{\leq i+1}$. If correctness were absorbing, then once a prefix was correct, later prefixes would almost always remain correct. Instead, a non-trivial number of trajectories transition from correct to wrong, showing that additional reasoning can mislead a correct intermediate solution. This is precisely the harmful-overthinking phenomenon studied in the main paper. The matrix also shows that trajectories are more likely to remain wrong than correct, further emphasizing the instability of reasoning once models leave the correct state.

B.4 Utterances and Tokens

Our main analysis uses utterances rather than raw tokens as the unit of reasoning budget. An utterance is a semantically coherent logical step in the generated trace, obtained by splitting the trace along

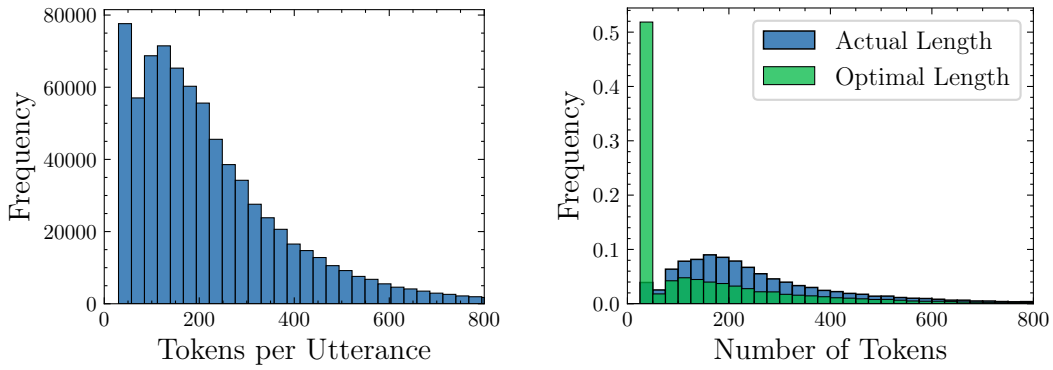

Figure 10: Token-level statistics for utterance-based reasoning budgets. Left: distribution of the number of tokens per utterance, showing that most reasoning steps are short but that occasional long utterances create a heavy tail. Right: token-budget distributions under *Actual Length* and *Optimal Length*; actual traces consume substantially more tokens than the first-correct prefixes, confirming that the utterance-level overthinking effect also appears at the token level.

explicit line-break delimiters ("\n\n" and \n) that LRMs naturally use when producing multi-step reasoning. This choice makes the budget less sensitive to formatting artifacts, local verbosity, and tokenizer-specific conventions. For example, two models may express the same intermediate step with different numbers of tokens, while both still represent a single reasoning transition in the trajectory.

Fig. 10 reports the relationship between utterance-level and token-level budgets. The left panel shows the distribution of tokens per utterance. Most utterances are short, but the distribution has a long tail, indicating that token count can be strongly affected by unusually verbose individual steps. The right panel compares token budgets under actual length and optimal length. The same qualitative pattern observed with utterances also appears at the token level: actual traces allocate substantially more computation than is required to first reach the correct answer. Thus, our conclusions are not an artifact of measuring compute in utterances. Utterances provide a cleaner trajectory step abstraction, while token statistics confirm that the gap between actual and sufficient reasoning remains visible under a lower-level compute measure.

B.5 On Verbose Overthinking

The main paper separates harmful overthinking from verbose overthinking. Harmful overthinking concerns correctness loss: the model reaches a correct prefix but terminates with an incorrect answer. Verbose overthinking concerns wasted computation: the model has already reached a correct answer and continues reasoning without changing the final outcome. In this Section we quantify the latter.

For each trajectory that reaches a correct prefix, we define the wasted budget as the number of utterances generated after the first correct prefix:

$$w(x; F) = N - \tau_y(x; F),$$

where N is the actual trace length and $\tau_y(x; F)$ is the first correct prefix. Large values indicate that the model solved the problem early but continued to spend inference compute.

Fig. 11 reports average wasted budget across multimodal benchmarks and models. The figure shows substantial variation across models. DualMind-VLM, which is trained to decide whether to reason fast or slow, exhibits the smallest wasted budget, averaging roughly 5 unnecessary utterances. In contrast, R1-VL produces the largest wasted budget, averaging roughly 18 unnecessary utterances. However, a lower wasted budget should not be interpreted as eliminating harmful overthinking: as shown in the main results, models with shorter traces can still deviate from correct trajectories.

B.6 Overthinking in Language Reasoning Models

The main paper shows that harmful overthinking is not restricted to multimodal reasoning. Fig. 12 visualizes the same effect for language-only models by comparing actual and optimal utterance

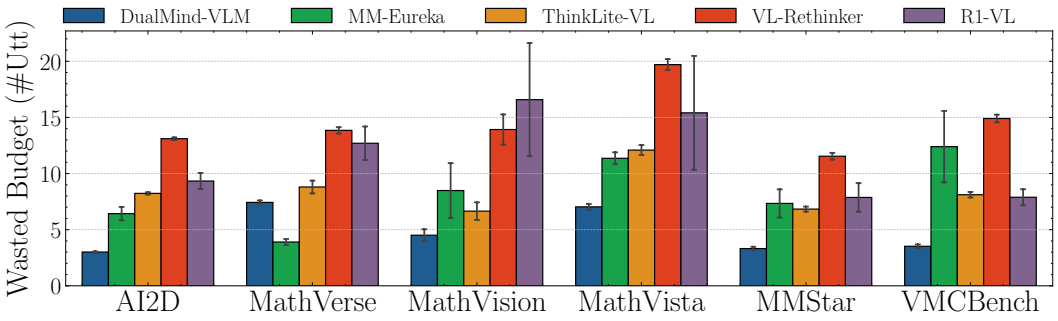

Figure 11: Average wasted budget in number of utterances per model per benchmark. DualMind-VLM [15], a model trained to predict input difficulty and use budget accordingly, achieves the lower wasted budget, with an average of 5 wasted utterances. R1-VL [46], whose base model is Qwen2VL [32], is the least "optimized" model, having an average wasted budget equal to 15 utterances, while having lower base performance than all the other models 1.

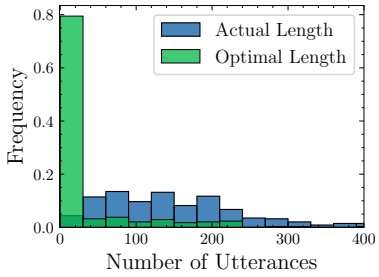

Figure 12: Actual *vs.*optimal reasoning length for language-only LRMs across all models and benchmarks. Actual traces are substantially longer than the first-correct prefixes, showing that language-only models also reason far beyond the point at which the correct answer first becomes recoverable.

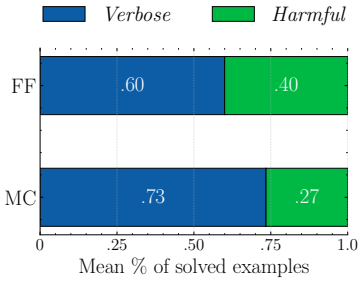

Figure 13: Harmful and verbose overthinking by answer format in language-only benchmarks. Free-form tasks exhibit higher harmful-overthinking rates than multiple-choice tasks, confirming the trend shown in the multimodal setting.

lengths on language benchmarks. Actual traces are extremely long, especially on mathematical reasoning tasks, whereas optimal prefixes are much shorter. This mirrors the multimodal setting: models often reach a correct solution far before their natural stopping point.

Fig. 13 reports harmful overthinking by answer format for language-only benchmarks. The effect is again stronger in free-form settings than in multiple-choice settings. This is consistent with the multimodal results: when the output space is unconstrained, the model must preserve and express the correct answer throughout the remainder of the trace, making it more vulnerable to later revisions and contradictions.

C Additional Details

Here, we provide the procedural details needed to reproduce our prefix-level evaluation and failure analysis. We describe the prefix-level probing setup, the taxonomy for harmful-overthinking cases, and the implementation details.

C.1 Prefix-Level Evaluation

Algorithm 1 summarizes the prefix-level trajectory protocol to estimate the difficulty of a sample for a given model. For each input, the model first generates a full reasoning trace. The trace is then split into utterances, and every prefix, including the empty prefix, is evaluated by appending a termination template and extracting a final answer. The returned difficulty is the first utterance index that yields

a correct answer. If no prefix yields the correct answer, the instance is treated as unsolved for that trajectory.

Early termination prompts

```
P1 = "Oh, I suddenly got the answer to the whole problem.
<answer> ### **Final Answer**: \[ boxed{"

P2 = "I got it now. I can now give the final response.
<answer> ### ***Final Answer**: \[ boxed{"
```

Figure 14: Termination prompts used for prefix-level probing. Each prompt is appended to a partial reasoning trace to force the model to stop deliberating and produce a final answer. The two variants preserve the same function while changing surface wording, allowing us to test whether estimated difficulty is sensitive to the exact probing phrase.

C.2 Taxonomy Experiment Details

The taxonomy experiment analyzes harmful-overthinking cases, i.e., trajectories that reach a correct answer at some prefix but terminate with an incorrect final prediction. For each case, we identify the last correct prefix and compare it with the full final trace, thereby isolating the additional reasoning segment responsible for the correctness deviation. Fig. 16 summarizes the prompt configuration used to extract the category and supporting evidence.

We classify each harmful trajectory into one dominant failure mode: visual hallucination/perception error, calculation error and Logical error.

We use an external judge model, `Qwen3.6-35B`, to assign the label. The judge receives the last correct prefix, the final trace, the ground-truth metadata, and, when available, the image associated with the example. The prompt instructs the judge to compare only the reasoning added after the last correct prefix and to ignore the standardized forced-answer suffix used by the probing pipeline. The judge returns a compact JSON object containing the primary category, optional secondary categories, severity, a short explanation, evidence, and confidence. We parse only valid JSON outputs; malformed outputs are discarded or re-run under the same prompt configuration.

C.3 Implementation Details and Reproducibility

Evaluation pipeline We re-implement and re-run all benchmark evaluations from scratch using a unified LLM-based answer-extraction pipeline. Instead of relying on benchmark-specific regular expressions, we apply a fixed answer extractor $\mathcal{A}$ to each generated trace and use the extracted concise answer for verification. This design is important because reasoning models frequently deviate from requested answer formats, and prefix-level probing produces partial traces whose answers can appear in heterogeneous forms. Unless otherwise specified, we use `Qwen/Qwen3-4B-Instruct-2507` as the extractor. Appendix A repeats the difficulty-estimation analysis with `Qwen/Qwen3.5-4B` to measure sensitivity to the parser. The extractor prompt is shown in Fig. 15.

Hyperparameters and Answer Extraction. For each evaluated reasoning model, we use the reference decoding configuration recommended by the corresponding model release whenever available, including temperature, top-p, maximum generation length, and image-processing settings. *Actual Length* denotes the model's natural termination behavior under this configuration. For prefix-level difficulty estimation, we first generate the complete reasoning trace, split it into utterances, and probe nested prefixes by appending a fixed termination template that asks the model to stop and provide a final answer. The failure-mode taxonomy in Appendix C.2 is produced by a separate judge model, which compares the last correct prefix with the final incorrect trace and labels the newly introduced error as visual, calculation, or logical. The judge prompt explicitly instructs the model to ignore the artificial termination suffix used by the probing pipeline.

Compute. All experiments are run with `vLLM` for batched inference on machines equipped with four NVIDIA A100-64GB GPUs. We store raw generations before answer extraction, which allows

Answer extractor $\mathcal{A}$ prompt

```
SYSTEM: You are a helpful assistant who extracts concise
answers from text. Extract only the direct answer provided
by the model, removing explanations.

USER: Given the following reasoning trace, extract ONLY
the final answer in a concise format.

Model Answer: {model_trace}

Extract the answer (just the answer itself, no explanations):
```

Figure 15: Prompt used by the answer extractor $\mathcal{A}$. The variable `model_trace` denotes the raw generation produced by the evaluated model, either at full length or after prefix-level probing. The extractor returns only the concise final answer used for benchmark verification.

Algorithm 1 PyTorch-style code for $\hat{\kappa}(x; \mathcal{F})$

```
# x = input problem
# F = reasoning model
# A = parser mapping output to a prediction
# y = ground-truth answer
# T = fixed termination template
def difficulty(x, F, A, y, T):
    # step 1: generate full reasoning trace
    t = F.generate(x)

    # step 2: split trace into utterances
    utts = split_utterances(t)

    # step 3: evaluate each prefix, including no reasoning
    for i in range(len(utts) + 1):
        prefix = "".join(utts[:i])
        prompted = prefix + T
        o_i = F.generate_from_prefix(x, prompted)
        y_hat_i = A(o_i)

        # step 4: return first correct index
        if verify(y_hat_i == y):
            return i

    # no correct prefix found
    return None
```

parsing, verification, robustness checks, and failure analyses to be repeated without regenerating expensive model traces. We release the evaluation scripts, prompts, decoding configurations, intermediate generations, parsed predictions, and analysis code required to reproduce the reported results. On average an evaluation on a benchmark can span from 1 to 4 hours depending on the model and dataset size (around 1K samples on average in our setting).

Packages, versions, and licenses. Our implementation was developed in Python 3.10.19, distributed under the Python Software Foundation License Version 2. We used vLLM v0.20.0 for efficient large language model inference, released under the Apache License 2.0; PyTorch v2.11.0+cu130 for tensor operations and GPU-accelerated model execution, released under a BSD-style license; and Hugging Face Transformers v5.6.2 for model and tokenizer interfaces, released under the Apache License 2.0.

D Limitations and Future Work

Verifiable outputs. Our analysis is limited to settings where correctness can be automatically verified, which is necessary for estimating the first correct prefix and separating verbose from harmful

overthinking. The conclusions are therefore strongest for benchmarks with well-defined ground-truth answers, such as mathematical reasoning, visual reasoning, and scientific QA. Open-ended generation, tool use, and coding tasks may require different definitions of correctness. For example, a program can be partially correct, fail hidden tests, or improve through later debugging. Extension to execution-based or subjective evaluation settings is an important direction for future work.

Model-dependent difficulty. The difficulty we estimate is not an intrinsic property of a problem alone, but a property of how a particular model processes that problem. We view this model dependence as a feature of the formulation rather than only a limitation. The same problem may be easy for one model and difficult for another, depending on the model's training data, post-training procedure, visual grounding ability, mathematical knowledge, reasoning shortcuts, and decoding policy. Accordingly, the empirical difficulty $\hat{\kappa}(x, y; F)$ should be interpreted as a model-conditioned quantity: it measures the minimum reasoning budget required by model F, on a sampled trajectory, to recover the correct answer. This is precisely the notion we aim to capture, since overthinking is also a property of a model's own reasoning dynamics rather than of the benchmark instance in isolation.

Compute accounting. The experiments require substantial inference compute because prefix-level probing evaluates many nested prefixes for each generated trace. We report the hardware used in Appendix C.2 and save intermediate generations to avoid unnecessary regeneration.

Oracle stopping and deployability. *Optimal Length* is not a deployable inference method because it requires ground-truth access to identify the first correct prefix. It serves as an oracle measuring how much performance is lost when models continue reasoning after a correct answer has already become recoverable. Developing practical stopping policies that approximate these oracles without access to ground truth remains an important open problem. Future work will explore how to best leverage the empirical difficulty estimate $\hat{\kappa}(x, y; F)$. It provides a possible supervision signal: models could be rewarded for reaching correct answers with sufficient but non-redundant reasoning, rather than for producing longer traces. This could support explicit stopping policies, model-agnostic difficulty predictors, or training objectives that penalize reasoning beyond the first correct prefix. The taxonomy analysis also suggests targeted interventions: visual errors may require stronger grounding, calculation errors may benefit from symbolic verification, and logical drift may require consistency constraints that prevent unsupported answer revisions.

Failure-analysis judge prompt configuration

```
TAXONOMY = [
    "visual_hallucination_or_perception",
    "calculation_error",
    "logical_error",
]

PROBE_SUFFIX_INSTRUCTION = """Important probe artifact:
- Ignore forced final-answer probe suffixes that start like
  "Oh, I suddenly/finally got the answer..." and lead into "\\boxed{".
- Treat that text as evaluator scaffolding, not as reasoning produced by
  the model under test.
- Do not classify a sample as a logical error only because this standard
  probe suffix appears.
- Classify the drift using the substantive reasoning or final-answer
  change before/around that scaffold."""

COMPACT_OUTPUT_INSTRUCTION = """Output style:
- Return exactly one compact JSON object.
- No analysis, markdown, prose, or preamble.
- Use the metadata as ground truth for last/final predictions.
- Keep went_wrong to one short sentence.
- Use example for one minimal quote/paraphrase from this sample."""

PROMPT = """
You are analyzing overthinking in a nested difficulty reasoning trace.

The first trace is the LAST prefix where the model's parsed answer was
still correct. The second trace is the FINAL prefix with all retained
utterances.

Task:
1. Compare only what changed after the last-correct prefix.
2. Identify the main failure mode introduced by the final/full trace.
3. If an image is provided, decide whether the added suffix hallucinates
   or misreads visual evidence.
4. Choose the best available category even when the drift is ambiguous.
5. Ignore the standard forced final-answer probe suffix.

Allowed categories: {categories}

Return only valid JSON:
{
  "category": "one_allowed_category",
  "secondary_categories": ["zero_or_more_allowed_categories"],
  "severity": 0_to_100_integer,
  "went_wrong": "short explanation",
  "evidence": "short quote or paraphrase from the added/final trace",
  "example": "minimal quote or paraphrase illustrating the reason",
  "confidence": 0.0
}
"""
```

Figure 16: Failure-analysis judge prompt. The judge compares the final incorrect trace against the last correct prefix and labels the dominant failure mode introduced by the additional reasoning. The prompt explicitly instructs the judge to ignore the standardized forced-answer suffix used by the probing pipeline.