Wait, Do We Really Need to "Wait"? Towards Training-Free Efficient Reasoning in R1-style Models

Anonymous EMNLP submission

Abstract

Recent advances in large reasoning models have enabled complex, step-by-step reasoning but often introduce significant *overthinking*, resulting in verbose and redundant outputs that hinder efficiency. In this study, we examine whether explicit self-reflection, signaled by tokens such as "*Wait*" and "*Hmm*", is necessary for advanced reasoning. We propose NOWAIT, a simple yet effective approach that disables explicit self-reflection by suppressing these tokens during inference. Extensive experiments on ten benchmarks across textual, visual, and video reasoning tasks show that NOWAIT reduces chain-of-thought trajectory length by up to 27%–51% in five R1-style model series, without compromising model utility. NOWAIT thus offers a plug-and-play solution for efficient and utility-preserving multimodal reasoning.

1 Introduction

Recent advancements in large reasoning models (LRMs), exemplified by DeepSeek-R1 (DeepSeek-AI et al., 2025), have shown that complex reasoning abilities can be effectively elicited through simple rule-based reinforcement learning (Qwen, 2025a,b; Abdin et al., 2025; Xia et al., 2025). These models produce explicit, step-by-step reasoning through long chain-of-thought (CoT) trajectories (Yang et al., 2025a; Ma et al., 2025a) before arriving at final answers. This capability is believed to be accompanied by the emergence of the "*Aha Moment*" phenomenon (Chen et al., 2025c; Yang et al., 2025b), in which the model begins to rethink problems and self-reflect on its reasoning trajectory with anthropomorphic expressions such as "*Wait*", "*Hmm*", or "*Alternatively*". This was firstly achieved on R1-style language reasoning models and has been extended to vision-language models (VLMs), enabling multimodal reasoning on images (Zhang et al., 2025b; Shen et al., 2025a; Huang et al., 2025; Zhou et al., 2025) and videos (Feng et al., 2025; Qwen, 2024; Du et al., 2025).

Despite the effectiveness of long CoT reasoning with self-reflection, the overthinking problem has emerged (Chen et al., 2024a; Cuadron et al., 2025; Chen et al., 2025b; Wu et al., 2025; Sui et al., 2025). It is characterized by excessively verbose reasoning and redundant thought steps, often extending over thousands of tokens, resulting in significant computational overhead and high reasoning latency. Such inefficiencies hinder the practical deployment of R1-style reasoning models in applications with limited computational resources.

Although numerous efforts have been devoted to efficient reasoning, many existing approaches require additional training, either through reinforcement learning (RL) with length-based rewards (Aggarwal and Welleck, 2025; Liao et al., 2025; Luo et al., 2025) or fine-tuning on variable-length CoT trajectories (Ma et al., 2025b; Munkhbat et al., 2025). On the other hand, several training-free approaches have been proposed to mitigate overthinking by reducing token usage during inference. However, they often compromise the overall model utility (Ma et al., 2025a) or have only demonstrated effectiveness on distilled reasoning models (Yang et al., 2025c,a; Xu et al., 2025).

In this study, we investigate the impact of excessive self-reflection during the reasoning process and question whether explicit self-reflection, signaled by "*Wait*"-like tokens, is really necessary for advanced reasoning. To this end, we propose NOWAIT, a simple yet effective training-free approach that disables explicit self-reflection in R1-style reasoning models, significantly reducing token usage while maintaining overall model utility. As illustrated in Figure 1, we directly intervene in the inference process by identifying specific keyword tokens (e.g., "*Wait*" and "*Hmm*") that indicate explicit self-reflection and suppressing their generation. Specifically, we achieve this by proactively

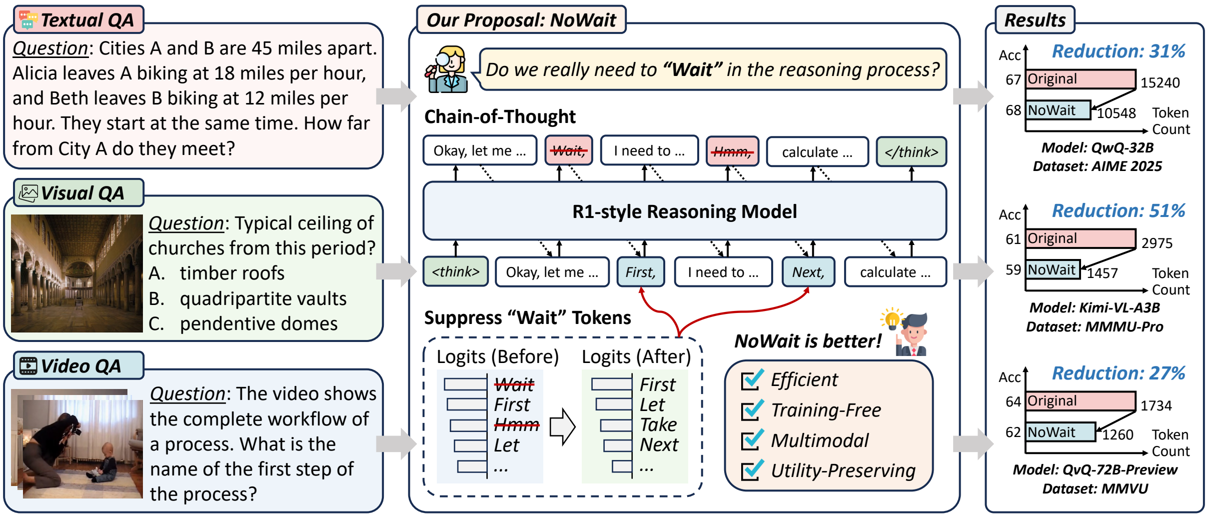


Figure 1: **Illustrative pipeline for NoWait.** We introduce NoWait, a simple yet effective approach that suppresses the generation of reflection keywords (e.g., "*Wait*" and "*Hmm*") during inference. NoWait reduces chain-of-thought trajectory length by up to 27%-51% across textual, visual, and video reasoning tasks.

adjusting the logits of these tokens to negative values during decoding, thereby steering the model toward selecting alternative tokens to continue the reasoning process.

Comprehensive experiments show that NoWait achieves strong performance on ten benchmarks spanning ❶ ***textual reasoning*** (AMC 2023 (AI-MO, 2024), AIME 2024, AIME 2025 (MAA Committees), GQPA-D (Rein et al., 2024)), ❷ ***visual reasoning*** (MMMU (Yue et al., 2024a), MMMU-Pro (Yue et al., 2024b), MathVista (Lu et al., 2024), EMMA-mini (Hao et al., 2025)), and ❸ ***video reasoning*** (MMVU (Zhao et al., 2025), VSI-Bench (Yang et al., 2024)). When integrated into five R1-style model series, including QwQ (Qwen, 2025b), Phi4 (Abdin et al., 2025), Qwen3 (Qwen, 2025a), Kimi-VL (Du et al., 2025), QvQ (Qwen, 2024), NoWait **reduces CoT trajectory length by up to 27%-51%** across different modalities. NoWait serves as a plug-and-play solution for improving reasoning efficiency while preserving overall model utility.

2 Preliminaries

Reasoning Model Generation Patterns. Reasoning models structure their output using thinking delimiters (i.e., <*think*> and <*think*>), dividing the response into two main components: the CoT trajectory detailing the reasoning process and the final answer summarizing overall thoughts.

Within the generated CoTs, models employ complex reasoning strategies, such as forward thinking, backtracking, and self-reflection. Notably, large reasoning models often continue to reason even after obtaining an initial result, performing additional validation steps. Accordingly, we define each segment of reasoning as a *thinking chunk*. Each thinking chunk is associated with an intermediate answer r. Formally, a thinking chunk can be represented as a pair $(chunk_i, r_i)$, where $chunk_i$ is the reasoning text and r_i is the intermediate answer from $chunk_i$ derived from $chunk_i$. Thus, a complete CoT can be structured as follows:

$$CoT = \{(chunk_i, a_i)\}_{i=1}^{n} . \qquad (1)$$

The final response is the combination of the CoT trajectory and a concise reasoning summary:

$$Response = (CoT, summary) . \qquad (2)$$

Self-Reflection within Reasoning Models. As stated above, a single CoT can contain multiple reasoning chunks. The transitions between these chunks are often marked by specific keywords, such as *Wait*, *Alternatively*, or *Hmm*. Models may switch their reasoning approaches in subsequent steps, often to verify previous results or explore alternative paths. However, this mechanism can sometimes lead to unproductive overthinking, causing models to repeatedly enter new reasoning steps and engage in seemingly endless validation loops. In this study, we further investigate this self-reflection phenomenon in the context of multimodal reasoning models. We introduce a simple yet effective method, which enables models to achieve more concise reasoning.

3 NOWAIT is Better

In this section, we propose NOWAIT, a simple yet effective method, that surprisingly improves the reasoning efficiency while maintaining acceptable model utility.

3.1 Method

Unlike methods that attempt to alter the underlying reasoning process, NOWAIT functions as an inference-time intervention. It *directly prevents* the model from generating the specific tokens associated with self-reflection. Our method involves three main stages:

Initialize Reflection Keywords List. We begin by identifying initial reflection keywords, such as *"Wait"*, *"Alternatively"*, and *"Hmm"*. To establish this list empirically, we conduct 32 independent runs of the QwQ-32B model (Qwen, 2025b) on the AIME 2025 benchmark. Using "\n\n" as delimiters, we identify the 15 most frequent monolingual words as our identified keywords $K = \{k_i\}$. All identified keywords are listed in Table 1.

Specific Token-Level Keyword List. Secondly, for each target model α, we expand the initial keyword list K into a specific token-level list, K_α. We achieve this by iterating through the overall vocabulary V_α and identifying all variant tokens whose textual representation contains any keyword from K as a substring. Specifically, we define that, $is\_substr(x, y) = True$ when x is the substring of y. This process can be formulated as follows:

$$K_\alpha = \{v \in V_\alpha | \exists k_s \in K, s.t. is\_substr(k_s, v)\}$$

We further manually filter keywords that are not reasonable (i.e., "Ohio" for "oh") to keep the robustness. For instance, the variants of "wait" include " wait", "Wait", " Wait", ".wait" and "WAIT".

Suppressing Keywords Generation. During the inference, we leverage a logit processor to prohibit models from generating keywords. For any keyword $v \in K_\alpha$, its corresponding logit is set to a large negative value. This effectively makes these reflection-associated tokens, ensuring they are highly unlikely to be sampled by models.

By surgically preventing the generation of these targeted reflection-associated tokens, NOWAIT aims to streamline the LRM's reasoning pathways. This targeted intervention is designed to enhance inference efficiency, reducing both latency and token costs, without requiring any modification to the model's underlying architecture or weights.

Table 1: Keyword List for Suppressing.

Keyword List for Suppressing
"wait", "alternatively", "hmm", "but", "however", "alternative", "another", "check", "double-check", "oh", "maybe", "verify", "other", "again", "now", "ah", "any"

3.2 Experimental Setup

Model & Benchmark. To comprehensively evaluate the effectiveness of NOWAIT, we conduct experiments on the open-source models across different modalities and parameter scales.

For the textual reasoning task, we assess reinforcement learning (RL) based models, including QwQ-32B (Qwen, 2025b), Phi4-Reasoning-Plus (Abdin et al., 2025), and Qwen3-32B (Qwen, 2025a) on math reasoning benchmarks, AIME 2024, AIME 2025 (MAA Committees), and AMC 2023 (AI-MO, 2024) as well as a professor-level benchmark, GPQA-Diamond (Rein et al., 2024).

For the visual reasoning task, our experiments cover the state-of-the-art RL-based vision reasoning models, Kimi-VL-A3B-Thinking (Du et al., 2025) and QvQ-72B-Preview (Qwen, 2024) and evaluate on MMMU-Pro (Yue et al., 2024b), MMMU (Yue et al., 2024a), MathVista (Lu et al., 2024) and EMMA-mini (Hao et al., 2025)

For the video reasoning task, we select QvQ-72B-Preview and evaluate on VSI-Bench (Yang et al., 2024) and MMVU (Zhao et al., 2025).

Metrics. The goal of NOWAIT is to preserve the model's reasoning accuracy while substantially diminishing the number of generated tokens during inference. Performance is assessed using two key metrics: ❶ ***Accuracy*** **(ACC)**: This measures the correctness of the model's final output. ❷ ***Generation Length*** **(LEN)** quantifies the average number of tokens generated by the model per problem instance, calculated over n independent runs.

Experiment Details. For each evaluated benchmark, we conduct five independent runs. Except for the Qwen3 series, we infer without chat templates on open-ended problems and leverage the same prompt template for multiple-choice problems (see Appendix D). Because of the different

Table 2: **Experiment results of method NoWait for RL-based models.**

Strategy	AMC 2023 ACC↑	AMC 2023 LEN↓	AIME 2024 ACC↑	AIME 2024 LEN↓	AIME 2025 ACC↑	AIME 2025 LEN↓	GQPA-D ACC↑	GQPA-D LEN↓
Textual Reasoning Task								
QwQ-32B								
Original	91.25	7542	73.33	14142	66.67	15240	62.12	6960
NoThink	72.50	4265	46.67	7980	40.00	8167	52.02	6556
NoWait	95.50 +4.25	5267 -30%	71.33 -2.00	11907 -16%	68.00 +1.33	10548 -31%	63.63 +1.51	6522 -6%
Phi4-Resoning-Plus								
Original	90.00	6366	70.00	15161	59.33	16257	61.61	5516
NoThink	80.83	3805	34.67	6200	31.33	5549	31.82	2776
NoWait	96.00 +6.00	4524 -28%	69.33 -0.67	11185 -26%	62.67 +3.34	12490 -23%	56.57 -5.04	4758 -14%
Qwen3-32B								
Original	97.50	6424	81.33	12720	66.67	14987	69.19	5613
NoThink	59.50	1240	25.33	2511	20.00	2165	50.50	605
NoWait	96.67 -0.83	5560 -13%	83.33 +2.00	10732 -16%	64.44 -2.67	12930 -14%	63.13 -6.06	4788 -15%

thinking patterns, we apply chat templates for the Qwen3 model inference. In baseline and NoWait experiments, we set a maximum token limit of 32,768 tokens per instance. If a model's generation reaches this limit before finishing CoT generation, that instance is considered incorrect, and the generation length is 32,768 tokens. If not, we will extract the final answer from the generated CoT and judge the correctness. This policy ensures that models failing to complete their response within the budget are appropriately penalized in *Accuracy* metric. For NoThink strategy (Ma et al., 2025a), we set a token budget of 10,000. Details can be found in subsection C.1.

3.3 LRMs can be Efficient without "Wait"

Table 2 presents a comprehensive quantitative overview of our NoWait's performance on various textual reasoning tasks, evaluated across different LRMs with diverse model structures and parameter scales. Our method NoWait consistently and significantly reduces the output length while maintaining or even improving reasoning accuracy across diverse model architectures.

Model Architectures Generalization. Notably, when integrated with QwQ-32B, NoWait improves accuracy on AMC 2023 by 4.25 percentage points, while reducing output length to just 70% of the baseline. With another model architecture, Phi4-Reasoning-Plus, our method achieves an even greater improvement of 6.00 percentage points, alongside a 28% reduction in token generation. Additionally, Qwen3-32B also benefits from our approach, reducing output length by 13% with only a marginal decrease in reasoning accuracy. These results demonstrate that our method NoWait consistently enhances efficiency across diverse model architectures. This consistency suggests a fundamental similarity in the reasoning patterns and redundancy present in different models, underscoring the broad applicability of our approach.

Reasoning Difficulty Analysis. We tested our method on mathematical reasoning benchmarks spanning various difficulty levels (AMC 2023 $<$ AIME 2024 $<$ AIME 2025). The experimental statistics demonstrated strong generalization across these levels: All tested models achieved comparable reductions in token usage regardless of task difficulty. Crucially, NoWait enabled models to maintain or even improve performance on more challenging tasks. For instance, QwQ-32B achieved a 1.33% point increase on the challenging AIME 2025 benchmark, while reducing token usage by 31%, which is comparable to its performance on the college-level AMC 2023. Qwen3-32B consistently reduced output length by 14% to 16% across all three math benchmarks, while Phi4-Reasoning-Plus showed similar gains and reductions from 23% to 26%. On the non-mathematical GPQA-Diamond task, models showed a slight per-

Table 3: **Experiment results of method NoWait for RL-based models.**

Strategy	MMMU-Pro		MMMU		MathVista		EMMA-mini	
	ACC↑	LEN↓	ACC↑	LEN↓	ACC↑	LEN↓	ACC↑	LEN↓
Kimi-VL-A3B-Thinking								
NoWait	58.73 -2.54	1457 -51%	55.20 -1.80	1746 -40%	69.40 -2.10	1045 -43%	27.50 -7.25	2269 -60%
Baseline	61.27	2975	57.00	2929	71.50	1822	34.75	5734
QvQ-72B-Preview								
NoWait	63.79 -1.98	1659 -21%	66.74 -0.11	1571 -21%	70.92 -2.62	939 -30%	28.00 -4.00	1554 -26%
Baseline	65.77	2094	66.85	1977	73.54	1338	32.00	2097

Table 4: **Experiment Results on Video Reasoning Tasks.** We use QvQ-72B-Preview for experiments.

Strategy	MMVU		VSI-Bench	
	ACC↑	LEN↓	ACC↑	LEN↓
Baseline	64.10	1734	22.51	1280
NoWait	62.20	1260	22.57	1020
Performance	-1.90	-27%	+0.06	-20%

Table 5: **Comparison Experiments across Multiple Efficient Reasoning Methods.** We use QwQ-32B-Preview for experiments.

Strategy	AIME 2024		AMC 2023	
	ACC↑	LEN↓	ACC↑	LEN↓
Baseline	42.00	8979	82.50	4143
Token-Budget	46.67	8734	82.50	3636
O1-Pruner	33.33	4289	77.50	2399
NoWait	42.00	5764	86.00	3396

formance decrease compared to the math reasoning benchmarks, but still maintained efficiency, with an overall 11.67% reduction in token usage.

These consistent efficiency gains and stable performance across diverse models and varying task difficulties suggest that, despite their scale, large reasoning models exhibit inherent redundancy in their output generation processes. NoWait effectively prunes this redundancy, demonstrating that substantial efficiency improvements can be achieved simply by generating more concise outputs, without the need for more complex or explicit "waiting" mechanisms.

3.4 Comparison Analysis

Comparison Experiment. We further compare with existing efficient reasoning techniques, including prompt-based training-free technique, Token-Budget (Han et al., 2024), and training-based technique, O1-Pruner (Luo et al., 2025), using QwQ-32B-Preview (Qwen, 2025b) on AIME 2024 and AMC 2023. All inference is conducted without chat templates to ensure fairness.

NoWait exhibits more significant generation length curtailment compared to Token-Budget. Although Token-Budget shows promising results on base models, such as GPT-4o, its effectiveness does not generalize to current LRMs(Deepseek-R1 (DeepSeek-AI et al., 2025), QwQ-32B (Qwen, 2025b)). These reasoning models are less sensitive to the prompt design, resulting in less efficiency. O1-Pruner, while effective at reducing token usage, incurs severe performance degradation on QwQ-32B-Preview. In contrast, NoWait does not require additional training or data, but instead guides models to strike an effective balance between output length and reasoning accuracy, achieving an inherent trade-off.

LRM Cannot Skip Thinking. As shown in Table 2, Qwen3-32B, a model specifically trained for non-thinking patterns, exhibits notable reductions in token usage. However, for other models (QwQ-32B and Phi4-Reasoning-Plus) without non-thinking pattern training, the prompt-based method, NoThinking (Ma et al., 2025a), fails to fully suppress the generation of reasoning steps. While NoThinking does reduce the generation length, the evaluated model can still generate lengthy text, which results in seriously compromising accuracy. This failure indicates that the presence of explicit "thinking" tokens is not as critical to reasoning performance as previously assumed. Our proposed

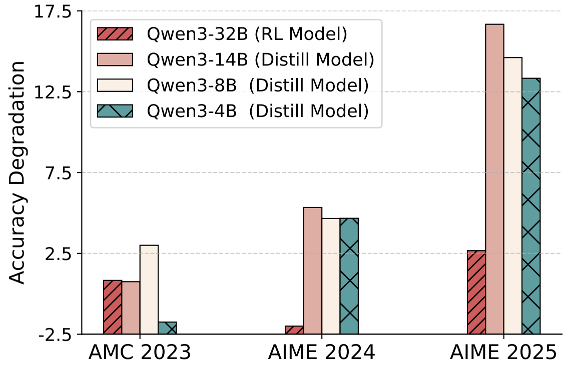


Figure 2: **Accuracy Degradation across Qwen3 Seires Models on Math Reasoning Benchmarks.**

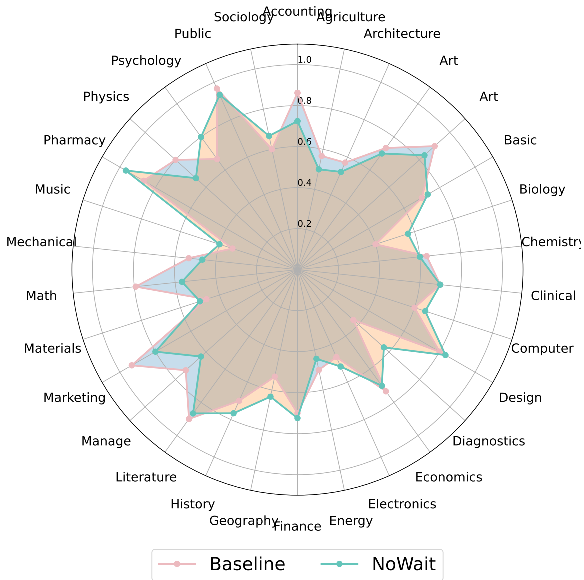


Figure 3: **Accuracy Radar Map on MMMU for QvQ-72B-Preview.**

NoWait operates on a similar premise by targeting key reasoning-related tokens, but achieves much greater efficiency improvements with minimal impact on accuracy.

3.5 Efficient Multimodal Reasoning

In this study, we first propose efficient multimodal reasoning and evaluate our method on vision reasoning models using both image and video reasoning benchmarks. As shown in Table 2, vision reasoning models exhibit more exciting outcomes.

Severe Verbosity on Multimodal Reasoning. Although Kimi-VL-A3B-Thinking generates an average of only 2,000 tokens across four image reasoning benchmarks - significantly fewer than that in math reasoning tasks - our method NoWait further reduces the generation length by an average of 49%, with only a modest overall accuracy drop of 3.42 percentage points. A similar trend is observed with QvQ-72B-Preview, which achieves up to a 30% reduction in token usage, accompanied by only a slight decrease in accuracy (ranging from 0.11% to 4.00%). For video reasoning tasks, QvQ-72B-Preview also demonstrates substantial reductions in output length while maintaining comparable accuracy. Similar to textual reasoning tasks, these results reveal the same challenging problems that a significant portion of generated tokens are either redundant or contribute little to the final reasoning. Existing multimodal reasoning models still suffer from severe overthinking.

Reinforcement Learning is Less Efficient. We further evaluate various RL-based reasoning models across various benchmarks and modalities. While a generation of intellectual reasoning models confirms the effectiveness of the RL algorithm in advanced reasoning capabilities, the efficiency of the optimal policy derived from the RL algorithm is still disappointing. The model learns a reasoning policy from training and begins to spontaneously reflect reasoning processes during inference. However, the RL algorithm fails to effectively teach these models when reflection is truly necessary. As a result, these models often adopt a lower threshold for self-reflection, leading to unnecessary verification steps and less efficient reasoning. Our method suppresses the generation of reflection keywords, raising the threshold of self-reflection, and making it more efficient and necessary.

4 Discussion

In this section, we first discuss the effectiveness of our method NoWait by case study (described in subsection 4.1) and the robustness of the model while applying NoWait. Additionally, we conduct an empirical experiment to analyze the difference between RL-based models and distill models based on NoWait in subsection 4.2.

4.1 Why does NoWait Work?

As we discussed in Table 3.4, thinking tokens (“<think>” and “<\think>”) as well as keywords (i.e., “wait”, “alternatively”, “hmm”) cannot completely trigger models’ actions. The thinking tokens failed to directly prevent the model from thinking, and banning keywords will not completely remove self-reflection in CoTs. Our method raises

Question

How does the energy of the pendulum changes after adding a new item in the experiment without consideration of friction?

Ground Truth Answer: D.

Thinking Length: 2119 Tokens

So I'm watching this video where there's a woman conducting some kind of experiment ...
She starts by explaining what she's going to do ...
At one point, she steps away from the table briefly and then returns to continue ...
Now, at the end of the video, there's a question about how the energy of the pendulum...
A. Larger B. Smaller, then larger C. Smaller D. Keep same E. Larger, then smaller ...
First, I should recall some basic physics about
So, if the woman adds a new item to the experiment without considering friction ...
Possible scenarios could be:
1. Adding mass to the pendulum: ... 2. Changing the length of the pendulum: ... 3. Adding an obstacle or another object that the pendulum interacts with: ...
But without knowing exactly what she added to the experiment, it's a bit tricky to be precise.
Let me think differently. Suppose she added something that doesn't directly ...
However, if she added mass to the pendulum ...
Wait a minute, maybe I should consider the principle of conservation of energy. ...
But option A suggests that ... Option B says smaller ... Option C says smaller ... Option D says keep same ... Option E says larger ... Given that the question specifies... D. Keep same.
However, if adding mass to the pendulum changes its energy ...
But I think the key here is
I'll go with that. **Final Answer** \[\boxed{D} \]

Thinking Length: 445 Tokens, NoWait !

So I'm watching this video where there's a woman conducting some kind of experiment ...
She *starts* by explaining what she's going to do ...
After that, she introduces a new element into the experiment. She adds something to the setup ...
Throughout the video, she's very animated and engaged, using her hands to show ...
At one point, she steps away from the table to retrieve something, and then returns to continue ...
As the video progresses, she continues to interact with the apparatus ...
Towards the end of the video, she summarizes what she's demonstrated ...
Overall, the video is a well-structured and informative demonstration of pendulum ...

Final Answer: \[\boxed{D} \]

Figure 4: **One Case Study From QvQ-72B-Preview on MMVU.** NoWait CoT is more straightforward than the original CoT, without unnecessary self-reflection and verbosity.

the threshold of self-reflection by suppressing keyword generation, substantially pruning less effective reflection steps.

Concise and Straightforward Reasoning. For an example from the MMVU benchmark, by carefully evaluating and comparing the original QvQ-72B's CoT (see Figure 7 and Figure 8) with the NoWait approach's QvQ-72B's CoT (see Figure 9), we can find that the **NoWait CoT** is more efficient primarily due to its concise reasoning structure, reduced self-reflection, and streamlined logical progression; it swiftly summarizes observations, confidently connects actions to outcomes without frequent hesitation, and promptly reaches the conclusion. In contrast, the **Baseline CoT** exhibits prolonged deliberation, redundant explorations of multiple hypothetical scenarios, repeated self-doubt, and extensive revisiting of basic principles, all of which slow down and complicate the reasoning process, ultimately making it less efficient.

Different from the original reasoning policy, NoWait guides models to link observations to conclusions without unnecessary speculation, making the reasoning more concise and straightforward.

More Efficient Self-Reflection Mechanism. NoWait does not prohibit models from self-reflection. Even if we identify a relatively comprehensive keyword list associated with self-reflection, the model can still conduct the double-checking operations. Specifically, the model will generate non-English transition terms, including Chinese, French, and other languages, or potentially structure its reasoning process.

By closely evaluating and comparing the original Qwen3-32B CoT and the Qwen3-32B CoT employing the NoWait strategy for an example from the AMC2023 benchmark, we observe that although the **NoWait CoT** doesn't litter its exposition with "hmm" or "let me think," it practices efficient self-reflection by embedding three targeted checkpoints—first, discarding the extraneous root immediately after factoring; second, briefly cross-validating the same result via a compact exponent-based substitution; and third, verifying both original equations numerically at the end—instead of the **Original CoT**'s repeated, explicit pausing and full re-derivation, thus maintaining clarity while ensuring correctness with minimal overhead.

A Closer Look at RL Models Performance. For textual reasoning tasks, our evaluation primarily includes math reasoning problems. As we discussed in subsection 3.3, NoWait demonstrates consistent experimental outcomes across different math benchmarks. For multimodal reasoning tasks, Figure 3 showcases the accuracy of the QvQ-72B-Preview across these various fields for Baseline and NoWait strategies on the MMMU benchmark. A crucial observation highlights remarkably small accuracy divergence between the baseline and NoWait strategies across almost all tested disciplines. Despite the potential intervention introduced by NoWait, the performance remains consistently close to the baseline across a wide range of academic and professional subjects. This minimal degradation across diverse domains strongly indicates the high robustness of the QvQ-72B-Preview when using the NoWait strategy, demonstrating its ability to maintain performance effectively without significant drops across varied areas.

4.2 Distill Models Cannot Reasoning without "Wait"

Recent studies (Yue et al., 2025) underscore the distinct difference between RL-based reasoning models and distill reasoning models in terms of their reasoning capabilities and robustness. To explore this difference further under challenging reasoning tasks, we evaluate NoWait on Qwen3 series (Qwen, 2025a), including the RL-based model, Qwen3-32B, and three distill models: Qwen3-14B, Qwen3-8B, and Qwen3-4B.

Figure 2 illustrates the accuracy degradation observed for each model across three math reasoning benchmarks: AMC 2023, AIME 2024, and AIME 2025, which represent increasing levels of difficulty (AMC 2023 < AIME 2024 < AIME 2025). A distinct accuracy degradation trend is evident between the RL model and the distill models. Qwen3-32B, which is post-trained with RL algorithms, exhibits consistently low accuracy degradation, showing fluctuations generally within the range of 1.5% to 3% across all three benchmarks.

In stark contrast, the distill models (Qwen3-14B, Qwen3-8B, and Qwen3-4B) show significantly higher and more varied degradation, particularly as the reasoning difficulty increases. On the relatively easier AMC 2023 benchmark, the degradation is slight, with Qwen3-4B even showing a slight performance improvement (-2.5%), while Qwen3-14B and Qwen3-8B experience moderate degradation (1.5% and 3% respectively). However, on the more challenging AIME 2024 and especially AIME 2025, the degradation for distill models escalates dramatically. On math reasoning benchmark AIME 2025, degradation reaches approximately 17% for Qwen3-14B, around 14% for Qwen3-8B, and about 13% for Qwen3-4B.

The sharp performance drop in distill models, unlike the stable RL model, shows their higher sensitivity to reflection keywords. Since SFT injects new knowledge directly, the CoT structure is crucial for the advanced reasoning. Simply removing these keywords disrupts this structure, preventing distill models from fully demonstrating their reasoning abilities, especially on harder tasks where effective self-checking is needed even without explicit reflection keywords.

5 Conclusion

This work demonstrates that explicit self-reflection, signaled by tokens such as "*Wait*" and "*Hmm*", is not essential for advanced reasoning in R1-style models. By suppressing these tokens during inference, the proposed NoWait approach effectively reduces overthinking and shortens chain-of-thought trajectories without compromising overall model utility. Extensive experiments across diverse models and benchmarks in textual, visual, and video reasoning tasks demonstrate that NoWait serves as an efficient and utility-preserving solution for multimodal reasoning, offering new insights for the lightweight deployment of large reasoning models.

Limitation

In this paper, we introduce the NoWait method to address the overthinking phenomenon. We have conducted experiments across various benchmarks and with a range of models, which yield positive results and demonstrate the effectiveness of our approach. Although we have shown the feasibility of NoWait on multiple and diverse benchmarks, we acknowledge that the benchmarks used in our experiments may still have certain limitations.

References

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, Piero Kauffmann, Yash Lara, Caio César Teodoro Mendes, Arindam Mitra, Besmira Nushi, Dimitris Papailiopoulos, Olli Saarikivi, Shital Shah, Vaishnavi Shrivastava, and 4 others. 2025. Phi-4-reasoning technical report. *Preprint*, arXiv:2504.21318.

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *Preprint*, arXiv:2503.04697.

AI-MO. 2024. AMC 2023. `https://huggingface.co/datasets/AI-MO/aimo-validation-amc`. Accessed: 2024-05-20.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *Preprint*, arXiv:2407.21787.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025a. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *Preprint*, arXiv:2503.09567.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025b. Do not think that much for 2+3=? on the overthinking of o1-like llms. *Preprint*, arXiv:2412.21187.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024a. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024b. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, and 1 others. 2025c. An empirical study on eliciting and improving r1-like reasoning models. *arXiv preprint arXiv:2503.04548*.

Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, and 1 others. 2025. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao ..., and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, Congcong Wang, Dehao Zhang, Dikang Du, Dongliang Wang, Enming Yuan, Enzhe Lu, Fang Li, Flood Sung, Guangda Wei, Guokun Lai, and 72 others. 2025. Kimi-VL technical report. *Preprint*, arXiv:2504.07491.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. 2025. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*.

Google. 2025. Gemini 2.5 pro. `https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-pro`.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2024. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*.

Yunzhuo Hao, Jiawei Gu, Huichen Will Wang, Linjie Li, Zhengyuan Yang, Lijuan Wang, and Yu Cheng. 2025. Can mllms reason in multimodality? emma: An enhanced multimodal reasoning benchmark. *Preprint*, arXiv:2501.05444.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, Christof Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. 2025. Reward-guided speculative decoding for efficient llm reasoning. *Preprint*, arXiv:2501.19324.

Kevin Lin, Charlie Snell, Yu Wang, Charles Packer, Sarah Wooders, Ion Stoica, and Joseph E. Gonzalez. 2025. Sleep-time compute: Beyond inference scaling at test-time. *Preprint*, arXiv:2504.13171.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *Preprint*, arXiv:2503.20783.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *Preprint*, arXiv:2501.12570.

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

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025b. Cot-valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*.

MAA Committees. Aime problems and solutions. `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`. Accessed: 2024-05-20.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. 2025. Self-training elicits concise reasoning in large language models. *Preprint*, arXiv:2502.20122.

OpenAI. 2024a. Chatgpt. `https://openai.com/o1/`.

OpenAI. 2024b. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Qwen. 2024. Qvq: To see the world with wisdom.

Qwen. 2025a. Qwen3.

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

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

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, and 1 others. 2025a. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, and Shiguo Lian. 2025b. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *Preprint*, arXiv:2503.04472.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Hanjie Chen, Xia Hu, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024. Fast best-of-n decoding via speculative rejection. *Preprint*, arXiv:2410.20290.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2025. When more is less: Understanding chain-of-thought length in llms. *Preprint*, arXiv:2502.07266.

Bingquan Xia, Bowen Shen, Cici, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, Liang Zhao, Peidian Li, Peng Wang, Shihua Yu, Shimao Chen, Weikun Wang, Wenhan Ma, and Zihao Yue Xiangwei Deng and... 2025. Mimo: Unlocking the reasoning potential of language model – from pretraining to posttraining. *Preprint*, arXiv:2505.07608.

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

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Zheng Lin, Li Cao, and Weiping Wang. 2025a. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.

Jihan Yang, Shusheng Yang, Anjali Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 2024. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. *arXiv preprint arXiv:2412.14171*.

Shu Yang, Junchao Wu, Xin Chen, Yunze Xiao, Xinyi Yang, Derek F. Wong, and Di Wang. 2025b. Understanding aha moments: from external observations to internal mechanisms. *Preprint*, arXiv:2504.02956.

Wang Yang, Xiang Yue, Vipin Chaudhary, and Xiaotian Han. 2025c. Speculative thinking: Enhancing small-model reasoning with large model guidance at inference time. *arXiv preprint arXiv:2504.12329*.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. *Preprint*, arXiv:2407.06023.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024a. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. 2024b. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Preprint*, arXiv:2504.13837.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025a. Reasoning models know when they're right: Probing hidden states for self-verification. *Preprint*, arXiv:2504.05419.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. 2025b. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *arXiv preprint arXiv:2503.12937*.

Yilun Zhao, Lujing Xie, Haowei Zhang, Guo Gan, Yitao Long, Zhiyuan Hu, Tongyan Hu, Weiyuan Chen, Chuhan Li, Junyang Song, Zhijian Xu, Chengye Wang, Weifeng Pan, Ziyao Shangguan, Xiangru Tang, Zhenwen Liang, Yixin Liu, Chen Zhao, and Arman Cohan. 2025. Mmvu: Measuring expert-level multi-discipline video understanding. *Preprint*, arXiv:2501.12380.

Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. 2025. R1-zero's" aha moment" in visual reasoning on a 2b non-sft model. *arXiv preprint arXiv:2503.05132*.

A Related Works

B Related Work

Large Reasoning Model The pursuit of advanced reasoning capabilities in Large Language Models (LLMs) (OpenAI, 2024b) has spurred significant research, particularly focusing on strategies that scale computation (Chen et al., 2024b; Snell et al., 2024) or refine the generation process during inference.

Prior studies apply fundamental techniques like Chain-of-Thought (CoT) prompting (Wei et al., 2023), guiding the model to think step by step, or integrate Process Reward Models(PRMs), external verifiers and search-guided decoding (Brown et al., 2024) to aggregate multiple reasoning paths and enhance final answer accuracy. These efforts have culminated in a new generation of powerful Large Reasoning Models (LRMs), such as ChatGPT-O1 (OpenAI, 2024a), Deepseek-R1 (DeepSeek-AI et al., 2025), QwQ (Qwen, 2025b), Gemini2.5 (Google, 2025), which enable to spontaneous generation of extensive CoT sequences involving forward thinking, backtracking, and verification steps.

Within the open-source domain, models derive reasoning abilities from diverse training paradigms, primarily through reinforcement learning (RL) (DeepSeek-AI et al., 2025; Ramesh et al., 2024; Muennighoff et al., 2025) on reasoning tasks or distillation (DeepSeek-AI et al., 2025; Yu et al., 2024) on high-quality CoT data produced from RL-based models. Recent works (Yue et al., 2025) have analyzed the difference between the two types of models. In this study, we include RL-based models for further exploration, underscoring the defects of RL-triggered reasoning capabilities.

Efficient Reasoning While elaborating reasoning processes like long CoT demonstrates enhanced performance on reasoning tasks, the associated verbosity presents a significant efficiency challenge (Chen et al., 2024a). The generation of extensive intermediate steps substantially increase inference latency and computational cost, hindering practical deployment in real-world applications. Consequently, a considerable body of work explores methods for efficient reasoning, aiming to reduce the length of reasoning traces without compromising accuracy.

Some techniques continue to train models for CoT optimization (Aggarwal and Welleck, 2025; Luo et al., 2025; Shen et al., 2025b), such as applying RL with length-based reward design (Sun et al., 2024; Liao et al., 2025; Luo et al., 2025; Aggarwal and Welleck, 2025), or fine-tuning with variable-length CoT data (Han et al., 2024; Yu et al., 2024; Munkhbat et al., 2025). Other methods conform training-free strategy, applying dynamic reasoning paradigms during inference (Yang et al., 2025a; Zhang et al., 2025a; Wu et al., 2025; Lin et al., 2025) or leveraging prompts to guide efficient reasoning (Cheng and Van Durme, 2024; Xu et al., 2025; Han et al., 2024; Ma et al., 2025a). While existing studies are effective in cutting down the token usage for reasoning, our study provides a new insight to rethink the internal mechanism of efficient reasoning and propose efficient multimodal reasoning.

Self-Reflection & Overthinking Parallel to enhancing reasoning capabilities and efficiency, recent studies analyze the intricacies of the generated thought processes. Within these generated sequences, an interesting phenomenon occurs - moments marked by keywords like "wait" and "hmm", which we term *Aha Moment* (DeepSeek-AI et al., 2025; Liu et al., 2025). These moments seemingly indicate a capability for self-reflection (Chen et al., 2025a), allowing models to reassess their reasoning path and verify their CoT before concluding.

Prior studies (Yang et al., 2025b; Zhang et al., 2025a) have begun to characterize these moments and probe the latent states to explore the potential mechanisms behind such spontaneous self-reflection. However, the frequent occurrence of these keywords can also lead to significant *Overthinking* (Chen et al., 2024a; Sui et al., 2025), where the model continues reflecting even after reaching correct intermediate or final conclusions. Building on the initial characterizations from previous work, our study takes a further step to evaluate the functional effectiveness of these spontaneously generated Aha Moments, directly addressing whether they are essential contributors to the reasoning outcomes or potentially represent a form of inefficient behavioral mimicry.

C Baseline Implementation Details

Our experiments include three baselines, NoThinking (Ma et al., 2025a), TokenBudget (Han et al., 2024), and O1-Pruner (Luo et al., 2025). In this section, we introduce the implementation details on these techniques.

C.1 NoThinking

The core idea of NoThinking is to leverage prompts, guide reasoning models to skip the reasoning processes and directly generating final response. For models that have not been post-trained for non-reasoning mode, such as QwQ-32B and Phi4-Reasoning-Plus, we apply the prompt template as follows:

Prompt Template for NoThinking

{Question}
<think>
Okay, I think I have finished thinking.
<\think>

We then adopt a budget forcing technique specifically for NoThinking. Different from the token budget we apply for normal inference and NOWAIT, we set the token budget to 10,000 and forced models to generate *Final Answer* when the model reaches the token budget.

C.2 Token-Budget

We apply TALE-EP strategy, a prompt-based method. This method consists of two steps:

1) Directly answering the reasoning model:

Prompt Template for TALE-EP

Task: Analyze the given question and estimate the minimum number of tokens required to generate a complete and accurate response. Please give the response by strictly following this format: [[budget]], for example, Budget: [[12]].

2) We include a token budget in the prompt to guide models thinking efficient.

C.3 O1-Pruner

O1-Pruner is an effective post-training method. We select a released model trained on QwQ-32B-Preview by O1-Pruner. This model can be accessible via Hugging Face.

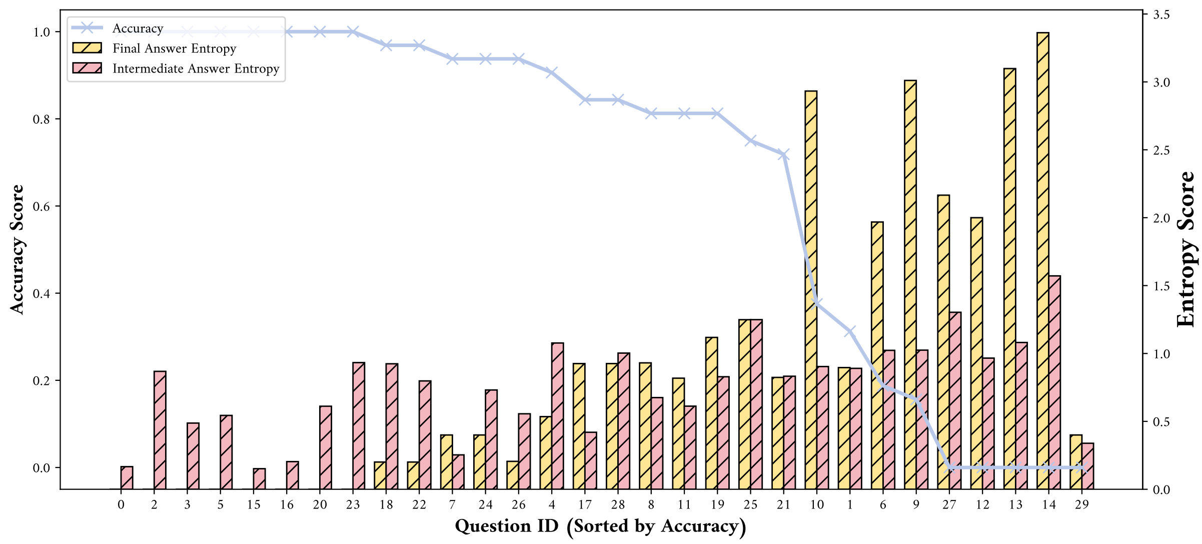


Figure 5: **Final / Intermediate answer entropy scores across AIME25.** As the accuracy degradation, the entropy scores of final answers increase gradually, while the intermediate answer entropy remains relatively lower.

Prompt method	**Content**
Vanilla CoT	Let's think step by step:
Token Budget	Let's think step by step and use less than {budget} tokens:

Table 6: Prompt Template Applied for Token Budget.

D Prompts

Prompt Template for Multiple-Choice Question

{Question}
—
Choices:
A. option A
B. option B
...
—
Choose the correct answer from the choices above.
Output format: [ANSWER: "<answer>"] If the answer is A, output [ANSWER: "A"]

E Additional Experiment: Is Self-Reflection Really Necessary?

E.1 Preliminaries

Reasoning models structure their output using thinking delimiters (i.e., <*think*> and <*think*>), dividing it into two main parts: the Chain-of-Thought (CoT), detailing the reasoning processes, and the final response by summarizing the thought process.

Within the generated CoTs, models employ complex reasoning strategies, including thinking forward, backtracking, and self-reflection. Notably, even after initially arriving at a result, LRMs frequently continue their reasoning for further validation. We define the subparagraph of reasoning or validation from its beginning up to the point where a result is generated as a "*thinking chunk*". Each thinking chunk is associated with one result r, referred to as the *intermediate answer* r. Conceptually, the thinking chunk can be represented as a pair $(chunk_i, r_i)$, where $chunk_i$ comprises the reasoning text and r_i is the intermediate answer from $chunk_i$. A complete CoT is thus formally structured as a sequence of these reasoning-answer pairs: $CoT = \{(chunk_i, a_i)\}_{i=1}^{n}$.

On the other hand, the final response is a brief summary without complex reasoning strategies such as reflection. Generally, models will provide a *final answer* in the response. The structure of final response can be denoted as $Response = (summary, ans)$.

E.2 Does Self-Reflection Really Work?

In this section, we conduct a pilot study to xxxx. There are multiple thinking chunks in the single CoT. Transition between different chunks are always marked by a series of keywords, such as *Wait* and *Alternatively*, and then reasoning models switch their reasoning strategies to double check the reasoning processes. However, this mechanism can sometimes lead to unproductive overthinking, causing models to repeatedly entering the new thinking chunk and endlessly validate the processes. Are these verbose self-reflection really effective to the advanced reasoning?

To investigate the practical effectiveness of self-reflection, we employ a two-fold analysis, assessing the association between intermediate answers and final answers.

E.3 Study Design

In this section, we conduct an empirical experiment to assess the effectiveness of self-reflection mechanism. We describe the experiment design in E.3, introduce experiment method in E.3 and metrics in E.3, and finally analyze the experiment results.

Setup. In this experiment, we first sample 32 runs on math benchmark AIME25 (30 problems), collecting the final answers across different samples and all the intermediate answers within 32×30 CoTs. Second, by quantifying and comparing the variability observed in both the intermediate answers and the overall reasoning paths, we aim to assess the nature and utility of reflection-driven exploration in reasoning models.

Intermediate Answer Data Collection. To enable a detailed analysis of the reasoning process, we precisely segment each Chain-of-Thought (CoT) into discrete thinking chunks. This process involves several steps.

The first step is to intially divide CoTs into thinking chunks. We perform an initial split of the raw CoT text into paragraph segments based on the double newline delimiter ("\n\n"). We then leverage reflection keywords (i.e., "*Wait*", "*Hmm*" and "*Alternatively*") to define the boundaries of thinking chunks. Each thinking chunk is formed by grouping one or more consecutive paragraph segments starting either at the beginning of the CoT or immediately following one of these reflection keywords, and ending just before the next occurrence of a keyword. Secondly, for each initially delineated thinking chunk, we employ GPT-4.1-mini (OpenAI, 2024b) to extract the intermediate answer. This step specifically targets identifying and extracting the result or conclusion presented at the end of the reasoning sequence within that chunk, if such a result is explicitly stated.

This structured representation facilitates the analysis of the step-by-step reasoning process and the role of intermediate results.

Evaluation & Metrics. To enhance the robustness of our experimental results, we perform $N = 32$ independent samplings for state-of-the-art model QwQ-32B (Qwen, 2025b) on math reasoning benchmark AIME25. To quantify the consistency of final answers and intermediate answers, we apply entropy score.

Specifically, for each question, we can collect two answer collections as described in E.3, including 32× final answers $A = \{ans_i\}_{i=1}^{N}$ and 32× intermediate answer collections $\{R_i|\{r_j\}_{j=1}^{M_j}\}_{i=1}^{N}$, where M is the count of intermediate answers within j_{th} CoT. Additionally, we define the calculation operation $U(\cdot)$, which is the corresponding non-duplicated collection, and $|\cdot|$ represents the count of the collection.

To quantify the consistency of the final answers, we first we calculate the entropy score $H(A)$:

$$H(A) = -\sum_{i=1}^{|U(A)|} (\frac{u_k}{N} \log(\frac{u_k}{N}))$$

where u_k is the k_{th} answer of $U(A)$.

For intermediate answer collections $C = \{R_i\}_{i=1}^{N}$, we first get the entropy score $H(R_i)$ for each sample. Then calculate the average score of 32 collections. The entropy score of intermediate answers can be defined as follow:

$$H(C) = \frac{1}{N}\sum_{i=1}^{N} H(R_i)$$

A higher H score indicates higher consistency, highlighting the variability of reasoning processes, while the lower score reflect less breaking point within reasoning processes.

E.4 Results & Analysis

Interestingly, models tend to generate consistent intermediate answers within a single Chain-of-Thought (CoT), regardless of correctness. However, across different samplings, models exhibit lower consistency and robustness, frequently generating various incorrect answers. Each sampling is independent and initiates a fresh reasoning sequence, demonstrating that LRMs can reach solutions through multiple reasoning routes or make distinct errors in different steps. Conversely, within a single CoT, repeated self-reflection yields highly consistent intermediate answers, highlighting the influence of earlier reasoning steps on subsequent processes. This contradict suggests the ineffectiveness of spontaneous self-reflection mechanisms implied in these models

E.5 Case Analysis on Attention Map

To find out why models determine to generate "Wait", in this section, we detect the attention map at the beginning of thinking chunks. We randomly select a CoT generated for AIME25, and split it into thinking chunks. We then calculate the attention score of the last token in CoT. It is clear that LRMs know how many times of self-reflection do they conduct, and the tails of thinking chunks exhibit much higher scores than those of intermediate reasoning processes. The high-score zone is overlapped with keywords position.

To further analyze whether keywords result in a high score or not, we delete all the keywords in the CoT, and recalculate the attention map. We find that there is no distinct difference before and after modification. Keywords do not act as the flag

F Benchmark & Models

F.1 Textual QA

In this paper, we evaluate a range of mathematics competition benchmarks designed to assess the mathematical reasoning abilities of models, including **AIME2024**, **AIME2025**, **AMC2023**. We have also evaluated **GPQA-Diamond** (Rein et al., 2024), a challenging benchmark spanning biology, physics, and chemistry. Here is the detailed information on these benchmarks:

- AIME2024: A benchmark derived from the 2024 American Invitational Mathematics Examination (AIME), a challenging mathematics competition aimed at high school students in the U.S., designed specifically to evaluate advanced mathematical reasoning abilities of AI models. It consists of complex problems covering algebra, geometry, combinatorics, and number theory, each requiring integer solutions ranging from 0 to 999. Models are tested on their ability to perform multi-step reasoning, provide accurate step-by-step explanations, and derive correct final answers.
- AIME2025: Like AIME2024, the AIME2025 benchmark is based on the 2025 American Invitational Mathematics Examination (AIME), an advanced and highly respected mathematics competition aimed at high school students in the United States, intended specifically for evaluating the mathematical reasoning and problem-solving capabilities of AI models.
- AMC2023: A benchmark derived from the 2023 American Mathematics Competitions (AMC), specifically designed to evaluate the mathematical reasoning abilities of AI models. It consists of 40 questions, covering various mathematical topics such as algebra, geometry, number theory, and combinatorics.
- GPQA-Diamond (Rein et al., 2024): A subset of the GPQA dataset, specifically designed to assess the reasoning capabilities of advanced AI systems and highly knowledgeable humans on extremely difficult, domain-expert-level questions in biology, physics, and chemistry. The "Diamond" subset is the hardest subset of the benchmark, intended to facilitate research on reasoning models.

We evaluated and measured these models on the above benchmarks:

- QwQ-32B (Qwen, 2025b): A large-scale language model designed to achieve robust performance across a wide range of natural language processing tasks. Developed with 32 billion parameters, Qwq32B leverages advanced architecture and training techniques to enhance understanding, generation, and reasoning in both general and specialized domains.
- Phi4-Reasoning-Plus (Abdin et al., 2025): An advanced language model specifically designed to excel in complex reasoning and problem-solving tasks across multiple domains, demonstrating strong performance in textual data.
- Qwen3-32B(Qwen, 2025a): A state-of-the-art large language model developed by Alibaba Cloud, featuring 32 billion parameters and designed to deliver high performance across a broad spectrum of language understanding and generation tasks.
- QwQ-32B-Preview (Qwen, 2025b): An experimental large language model developed by Alibaba's Qwen team, designed to advance AI reasoning capabilities. With 32.5 billion parameters and a 32,768-token context window, it is specifically tested on benchmark

AIME2024 and AMC2023 to compare with other methods.

F.2 Visual QA

Additionally, we incorporate evaluations on the multimodal benchmarks including **MMMU-Pro** (Yue et al., 2024b), **MMMU** (Yue et al., 2024a), **Math-Vista** (Lu et al., 2024) and **EMMA-mini** (Hao et al., 2025) to further explore the models' capabilities across diverse reasoning and multimodal tasks. Here is the detailed information on these benchmarks:

- MMMU (Yue et al., 2024a): A large-scale, multimodal evaluation benchmark specifically designed to test the capabilities of AI models on college-level tasks that require both advanced subject knowledge and deliberate reasoning across a broad range of academic disciplines.

- MMMU-Pro (Yue et al., 2024b): MMMU-Pro is an enhanced evaluation benchmark designed to rigorously test the true understanding and reasoning capabilities of multimodal AI models. Building on the original MMMU benchmark, it forces models to simultaneously process and integrate visual and textual information, simulating real-world scenarios that require human-like cognitive skills.

- Math-Vista (Lu et al., 2024) : A comprehensive benchmark specifically designed to evaluate and challenge the mathematical reasoning abilities of large language and multimodal models within visual contexts. MathVista requires models to perform deep, fine-grained visual understanding and complex compositional reasoning across diverse mathematical tasks.

- EMMA-mini (Hao et al., 2025): A specialized benchmark designed to rigorously assess the ability of Multimodal Large Language Models (MLLMs) to perform integrated, organic reasoning over both text and images—an essential aspect of human intelligence. Unlike existing benchmarks that often focus on text-based reasoning or superficial visual cues, EMMA-mini presents tasks spanning mathematics, physics, chemistry, and coding, all of which require genuine cross-modal reasoning that cannot be solved by independently analyzing text or images alone.

We evaluated and measured these models on the above benchmarks:

- Kimi-VL-A3B-Thinking (Du et al., 2025): An efficient open-source vision-language model (VLM) built on a Mixture-of-Experts (MoE) architecture, designed to deliver advanced multimodal reasoning, robust long-context understanding, and strong agent capabilities.

- QvQ-72B-Preview (Qwen, 2024): QVQ-72B-preview is an open-source, large-scale multimodal reasoning model built on top of Qwen2-VL-72B, achieving remarkable performance on challenging benchmarks. In this part of the experiment, the image recognition and reasoning capability of this model has been tested.

F.3 Video QA

Furthermoer, we conduct experiment on video benchmarks, whose name and details is listed as follows:

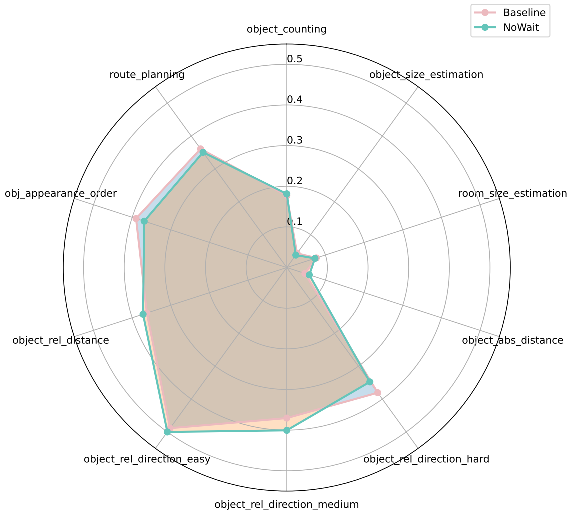
- MMVU (Zhao et al., 2025): A comprehensive dataset designed to evaluate the capabilities of AI models in understanding and reasoning over expert-level, domain-specific videos. Each example is meticulously crafted using a textbook-guided annotation process, ensuring that questions require both visual comprehension and the application of domain-specific knowledge. Unique to MMVU is the inclusion of expert-annotated reasoning rationales and relevant domain knowledge for each question, facilitating fine-grained analysis of model performance.

- VSI-Bench (Yang et al., 2024): A pioneering dataset, designed to evaluate the visual-spatial reasoning capabilities of multimodal large language models (MLLMs). It comprises many question-answer pairs derived from egocentric videos sourced from public indoor 3D scene reconstruction datasets, aiming to provide a comprehensive benchmark for testing and improving the spatial reasoning abilities of MLLMs, moving beyond traditional static image evaluations.

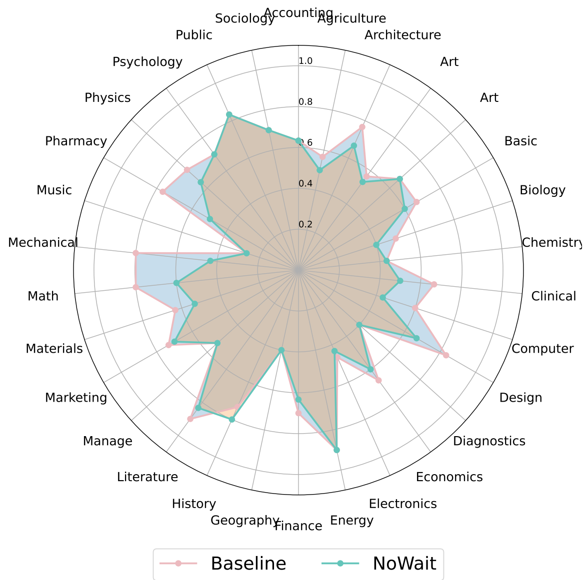
G Additional Experiment Results & Case Study

Table 7: **Complete Experiment results of NoWait on Qwen3 Series Models and Other Distill Models.**

Strategy	**AMC 2023**		**AIME 2024**		**AIME 2025**		**GQPA-D**	
	ACC↑	LEN↓	ACC↑	LEN↓	ACC↑	LEN↓	ACC↑	LEN↓
				Qwen3-32B				
NoWait	96.67	5560	83.33	10732	64.44	12930	63.13	4788
NoThink	59.50	1240	25.33	2511	20.00	2165	50.50	605
Baseline	97.50	6424	81.33	12720	66.67	14987	69.19	5613
				Qwen3-14B				
NoWait	95.50	4714	73.33	10919	61.33	12104	54.75	3889
NoThink	69.50	1749	33.33	3559	26.67	3171	38.59	1001
Baseline	96.25	6677	78.67	14217	78.00	14765	59.59	4633
				Qwen3-8B				
NoWait	94.50	5251	72.67	10963	60.00	13674	51.71	4735
NoThink	66.50	1760	28.89	3362	25.56	3719	32.93	1271
Baseline	97.50	8513	77.33	14142	74.61	16094	57.07	5904
				Qwen3-4B				
NoWait	95.50	4523	65.33	10358	56.67	12213	76.26	3178
NoThink	70.00	2236	33.33	4068	23.00	4656	27.27	1288
Baseline	93.75	8125	70.00	13488	70.00	18086	53.54	5965
				Llama-Nemontron-Nano-8B-v1				
NoWait	72.00	3690	30.67	4865	33.33	7271	42.83	3754
NoThink	39.00	2982	6.6	2185	14.67	3677	30.30	3109
Baseline	71.50	4535	39.33	7371	44.67	11798	54.1	5071
				Deepseek-R1-Distill-Qwen-7B				
NoWait	72.00	4315	26.67	7247	31.33	8236	40.91	3672
NoThink	30.00	2552	18.67	3895	10.00	2724	21.00	1112
Baseline	73.00	4796	34.67	7755	40.00	13767	49.10	3809



(a) **Accuracy Radar Map for QvQ-72B-Preview on VSI-Bench**



(b) **Accuracy Radar Map for Kimi-VL-A3B-Thinking-7B on MMMU**

Original CoT:

So I'm watching this video where there's a woman conducting some kind of experiment or demonstration. She's in a room that looks like a laboratory or a classroom with a lot of scientific equipment and wooden cabinets in the background. On the table in front of her, there's this setup that seems to be a pendulum or some sort of apparatus that can move back and forth.

She starts by explaining what she's going to do. She mentions a pendulum and talks about its motion. Then, she begins to manipulate the apparatus, pushing it back and forth. I can see that the object is moving in a way that suggests it's a pendulum, swinging from one side to the other.
As she continues, she adds another component to the setup. It looks like she's introducing a new element into the experiment, maybe to observe how it affects the pendulum's motion. She keeps adjusting things and observing the results.

At one point, she steps away from the table briefly and then returns to continue her demonstration. Throughout the video, she's engaged in showing and explaining the principles behind the pendulum's behavior.

Now, at the end of the video, there's a question about how the energy of the pendulum changes after adding a new item to the experiment, assuming no friction is present. The options are:

A. Larger B. Smaller, then larger C. Smaller D. Keep same E. Larger, then smaller
I need to choose the correct answer from these options.

First, I should recall some basic physics about pendulums and energy conservation. A pendulum swings back and forth, converting potential energy to kinetic energy and back again. If there's no friction or other dissipative forces, the total mechanical energy of the pendulum should remain constant. That means that the sum of its potential energy and kinetic energy shouldn't change over time.

So, if the woman adds a new item to the experiment without considering friction, I need to think about how this addition affects the pendulum's energy.

Possible scenarios could be:
1. Adding mass to the pendulum: If she adds mass to the pendulum, the potential energy at the highest points would increase because potential energy depends on mass and height. Since total energy is conserved, the kinetic energy at the lowest point would also have to adjust accordingly. So, the overall energy might be larger if mass is added, but I'm not entirely sure.
2. Changing the length of the pendulum: Changing the length affects the period of the pendulum but doesn't directly change its energy if the amplitude remains the same. However, if the length is changed and the amplitude is adjusted, that could affect the energy.

Figure 7: **A CoT Example from QvQ-72B-Preview on MMVU 2023.**

Original CoT(continue):

3. Adding an obstacle or another object that the pendulum interacts with: This could transfer some of the pendulum's energy to the other object, possibly reducing the pendulum's energy.

But without knowing exactly what she added to the experiment, it's a bit tricky to be precise.
Let me think differently. Suppose she added something that doesn't directly interact with the pendulum's energy, like a support or another structural component that doesn't affect the motion. In that case, the energy of the pendulum should remain the same, assuming no energy is being lost or gained from external sources.

However, if she added mass to the pendulum, say by attaching something to it, that would increase its moment of inertia, potentially affecting its motion and energy.

Wait a minute, maybe I should consider the principle of conservation of energy. If no friction is present and no external work is done on the system, the total energy should remain constant. So, unless she's adding or removing energy in some way, the energy of the pendulum should stay the same.

But option A suggests that the energy becomes larger, which might be the case if she added mass without accounting for the increased potential energy.

Option B says smaller, then larger, which doesn't make much sense immediately.

Option C says smaller, which might be if she added something that absorbed energy, like a damping mechanism.

Option D says keep same, which aligns with the conservation of energy principle if no energy is added or removed.

Option E says larger, then smaller, which also seems unclear.

Given that the question specifies "without consideration of friction," and assuming that the added item doesn't dissipate energy, the most straightforward answer would be D. Keep same.
However, if adding mass to the pendulum changes its energy, then perhaps it could be A. Larger.
But I think the key here is that if no friction is considered, and no energy is being added or removed, the energy should remain constant. Therefore, the answer should be D. Keep same.
I'll go with that.
Final Answer
\[\boxed{D} \]

Figure 8: **A CoT Example from QvQ-72B-Preview on MMVU 2023.** [2]

No Wait CoT:

So I'm watching this video where there's a woman conducting some kind of experiment or demonstration. She's in a room that looks like a laboratory or a classroom with a lot of scientific equipment and wooden cabinets in the background. On the table in front of her, there's this setup that seems to be a pendulum or some sort of apparatus that can move left and right.

She starts by explaining what she's going to do. She mentions a pendulum and talks about its motion. Then, she begins to manipulate the apparatus, pushing it and observing how it moves. She does this a few times, and it seems like she's demonstrating the natural motion of the pendulum without任何 external forces acting on it.

After that, she introduces a new element into the experiment. She adds something to the setup, possibly a magnet or另一个 pendulum, and continues to observe the motion. This time, the motion of the original pendulum is affected by this new addition, and she explains what's happening.

Throughout the video, she's very animated and engaged, using her hands to show what's going on and speaking clearly to the camera. The room is filled with various scientific instruments and models, which adds to the educational atmosphere of the setting.

At one point, she steps away from the table to retrieve something, and then returns to continue her demonstration. This suggests that she's prepared multiple parts to her experiment, each building upon the last to illustrate a particular concept or principle.

As the video progresses, she continues to interact with the apparatus, adjusting it and observing the effects. Her explanations are thorough, and she makes sure to point out key points in the motion and how they relate to the properties of pendulums.

Towards the end of the video, she summarizes what she's demonstrated and how it relates to the choices provided in the question. She emphasizes the importance of considering friction and otros factors that might affect the energy of the pendulum over time.

Overall, the video is a well-structured and informative demonstration of pendulum motion and the factors that influence its energy, particularly focusing on the changes in energy when additional elements are introduced into the system.

Final Answer

\[\boxed{D} \]

Figure 9: **A CoT Example from QvQ-72B-Preview on MMVU 2023.** [1]