# ALPHAONE: Reasoning Models Thinking Slow and Fast at Test Time

**Junyu Zhang**[I†] **Runpei Dong**[I†] **Han Wang**[I] **Xuying Ning**[I]
**Haoran Geng**[B] **Peihao Li**[B] **Xialin He**[I] **Yutong Bai**[B]
**Jitendra Malik**[B] **Saurabh Gupta**[I] **Huan Zhang**[I]

[I]University of Illinois Urbana-Champaign [B]UC Berkeley
[†]**Equal contributions.** **Correspondence:** {junyuz6, runpeid2, huanz}@illinois.edu

## Abstract

This paper presents ALPHAONE ($\alpha$1), a universal framework for modulating reasoning progress in large reasoning models (LRMs) at test time. $\alpha$1 first introduces $\alpha$ moment, which represents the scaled thinking phase with a universal parameter $\alpha$. Within this scaled pre-$\alpha$ moment phase, it dynamically schedules slow thinking transitions by modeling the insertion of reasoning transition tokens as a Bernoulli stochastic process. After the $\alpha$ moment, $\alpha$1 deterministically terminates slow thinking with the end-of-thinking token, thereby fostering fast reasoning and efficient answer generation. This approach unifies and generalizes existing monotonic scaling methods by enabling flexible and dense slow-to-fast reasoning modulation. Extensive empirical studies on various challenging benchmarks across mathematical, coding, and scientific domains demonstrate $\alpha$1's superior reasoning capability and efficiency. Project page: https://alphaone-project.github.io/

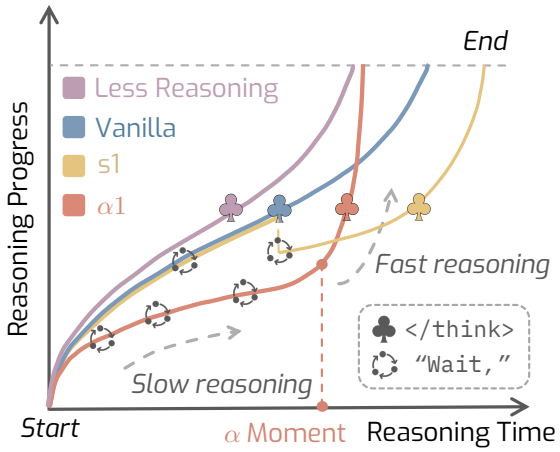



Figure 1: **Conceptual illustration** of reasoning modulation strategies. Our $\alpha$1 employs a *slow-to-fast* reasoning schedule controlled by $\alpha$. $\alpha$1 scales more efficiently than *monotonously increasing* method s1 (**yellow**) and generally outperforms *monotonously decreasing* (**purple**) approaches.

## 1 Introduction

> "The most effortful forms of slow thinking are those that require you to think fast."
>
> *Thinking, Fast and Slow (Kahneman, 2011)*

Large Reasoning Models (LRMs) such as OpenAI o1 (Jaech et al., 2024) and DeepSeek-R1 (DeepSeek-AI et al., 2025) have demonstrated unprecedented progress in approaching human-like system-2 reasoning capabilities, enabling *slow thinking*—slowing down *reasoning progress*[1] at test time—for solving complex reasoning problems that require high-order cognitive processing. These advanced models are trained to utilize slow thinking via reinforcement learning, enabling LRMs to slow down reasoning progress automatically. Is such automatic slowing down of reasoning progress determined by LRMs sufficiently reliable? According to Kahneman (2011), humans typically think fast first and activate slow thinking when running into difficulty, through a conscious control of system-1-to-2 reasoning transitioning, resulting in overall comprehensive but efficient reasoning. While similar to human systems and interesting results have been observed, a lot of works have pointed out that the LRMs themselves are prone to *overthinking* (Chen et al., 2024b; Sui et al., 2025; Pu et al., 2025; Yang et al., 2025c) or *underthinking* (Su et al., 2025; Yang et al., 2025d; Wang et al., 2025). This is because of the inability of LRMs to find the optimal human-like system-1-to-2 reasoning transitioning and limited reasoning capabilities, leading to unsatisfactory reasoning performance.

To overcome this limitation, existing works scale LRMs at test time in mainly two ways. i) *paral-*

[1]Consider reasoning progress as a metric ranging from 0 to 1, indicating the start and the end of reasoning, respectively. It increases slowly or fast at the pace of thinking. See Fig. 1 and Section 2 for a more illustrative and detailed explanation.



*Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11341–11366
November 4-9, 2025 

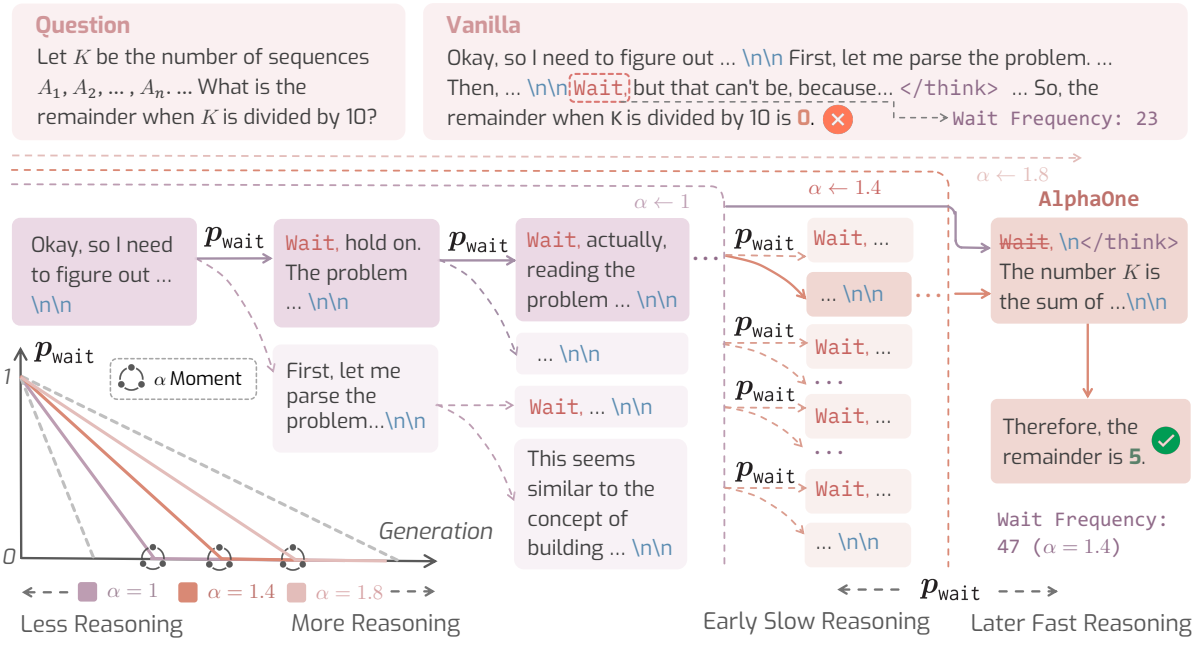



Figure 2: **Overview of ALPHAONE ($\alpha$1).** Here ◌ represents $\alpha$ moment (Section 3.1). $\alpha$1 applies dense reasoning modulation via a user-defined slow thinking scheduling in pre-$\alpha$ moment. In addition, $\alpha$1 utilizes a post-$\alpha$ moment modulation by replacing slow thinking transitioning tokens "`wait`" to "`</think>`", which fosters fast thinking. Specifically, $\alpha$ determines when the slow-to-fast reasoning transition occurs. For example, reducing $\alpha$ from 1.4 to 1.0 shifts the $\alpha$ moment earlier, resulting in shorter slow reasoning phase and accelerating the annealing of $\boldsymbol{p}_{\texttt{wait}}$.

*lel scaling*: this line of research follows a *Best of $N$* strategy and typically samples $N$ times and outputs the best answer using criteria such as self-consistency (Wang et al., 2023; Wan et al., 2024a; Zhou et al., 2025; Ma et al., 2025) and perplexity (Fang et al., 2025). ii) *sequential scaling*: this family of approaches addresses the overthinking/underthinking issues via early reasoning stopping (Fu et al., 2025; Yang et al., 2025a; Xu et al., 2025a) and promoting for reinforcing reasoning (Muennighoff et al., 2025; Wang et al., 2025), respectively. For example, Xu et al. (2025a) proposes Chain of Draft, prompting LRMs to think fast strictly within 5 words to significantly reduce overthinking. s1 (Muennighoff et al., 2025) proposes to foster reasoning continuously via appending a slow-reasoning transition token "`wait`" multiple times when LRMs are about to end. However, it is unclear if such monotonous reasoning increment or reduction is optimal, and the appropriate moment for slow thinking transitioning is still underexplored. Hence, instead of test-time scaling with an automatic slowing down by LRMs themselves or simply increasing or reducing slow thinking, we are interested in finding: *Can we modulate reasoning progress universally, and develop a better slow thinking transitioning strategy with it?*

To answer this question, we present **ALPHAONE** ($\alpha$1), which efficiently scales LRMs at test time through a *universal* reasoning progress modulation. We introduce *alpha moment*, parameterized by $\alpha \geq 0$, where the thinking process is scaled by $\alpha$ times throughout the whole generation sequence. To be specific, within a certain token length scaled by $\alpha$, we stochastically append the reasoning transition token "`wait`" after structural delimiters "`\n\n`" under $\text{Bernoulli}(\boldsymbol{p}_{\texttt{wait}})$, inspired by the observation that these two frequently co-exist (Yang et al., 2025c). Here, $\boldsymbol{p}_{\texttt{wait}}$ is scheduled to change over time to *activate* slow thinking. For example, a simple linear annealing over time indicates a slow thinking first, then fast thinking strategy.

However, we observe that amplifying slow thinking enables LRMs to sustain it automatically. Thus, when $\boldsymbol{p}_{\texttt{wait}}$ reaches 0, we replace "`wait`" with "`</think>`" to deactivate slow thinking and switch to fast reasoning. In this fashion, $\alpha$1 unifies prior methods like s1 (Muennighoff et al., 2025), where $\alpha$1 reduces to s1 if $\boldsymbol{p}_{\texttt{wait}}$ is 1 or 0 at the end of a reasoning segment within a certain reasoning token length. However, different from these works that only explore sparse slow reasoning modulation, $\alpha$1 modulates reasoning continuously, supporting both *sparse* and *dense* modulation strategies.

**Takeaways** We present some insightful findings from evaluating three different $\alpha$1 LRMs, ranging from 1.5B to 32B across six reasoning benchmarks, including math, code generation, and scientific problem reasoning: i) *Slow thinking first*, then fast thinking, leads to better LRM reasoning. Surprisingly, this differs from humans who commonly think fast, followed by slow thinking (Kahneman, 2011), emphasizing the requirement of dedicated test-time scaling strategies for LRMs. ii) Slow thinking can bring *efficient* test-time scaling. While slow thinking slows down reasoning, the overall token length is significantly reduced with $\alpha$1, inducing more informative reasoning progress brought by slow thinking. iii) Slow thinking transitioning in *high frequency* is helpful. Interestingly, we find that $\alpha$1 appending "`wait`" significantly more (*e.g.*, over $2\times$ more than s1) achieves much better results.

## 2 Background & Problem Statement

**Revisiting Reasoning Models** Following the success of OpenAI's o1 model (Jaech et al., 2024), modern LRMs solve complex reasoning problems via a *thinking-then-answering* paradigm (DeepSeek-AI et al., 2025; Qwen Team, 2025; Huang et al., 2024b). Generally, a special end-of-thinking token "`</think>`" is generated as a *end-of-thinking moment*, transitioning from the thinking phase to the answering phase. During the thinking process, LRMs automatically transit between slow thinking and fast thinking, utilizing self-reflection as chain of thoughts (Wei et al., 2022).

**Slow Thinking Transitioning** To leverage human-like system-2 slow thinking that helps solve complex reasoning problems, o1-style LRMs automatically transit between fast thinking and slow thinking. To be specific, during the thinking process, LRMs frequently generate slow thinking transitioning tokens such as "`wait`", "`hmm`", and "`alternatively`", *etc*. Once these tokens are generated, LRMs slow down reasoning, where previous reasoning chains are self-reflected and corrected immediately. Hence, reasoning following the transitioning token can be viewed as slow thinking, while the rest is generally fast thinking.

**Reasoning Progress** Let the overall answer sequence generation process be a *reasoning progress* $\mathcal{P} \in [0, 1]$, where 0 and 1 indicate the start and the end of reasoning, respectively. Notably, reasoning progress represents the overall problem-solving progress instead of the number of generated tokens, where a reasoning progress closer to 1 represents the reasoning chain is more informative. For example, the reasoning progress can be closer to 1 while generating fewer tokens, indicating more efficient reasoning. However, it is intractable to measure the exact progress obtained. Hence, we define the reasoning progress following a *reasoning velocity* assumption. Given the total time $t = T > 0$ spent on generating the whole sequence, the *reasoning velocity* at timestep $t$, $\mathcal{V}_t$ is defined as $\frac{d\mathcal{P}}{dt}$, where $dt$ is the infinitesimal of time. We assume:

**Assumption 1.** *The reasoning velocity of slow thinking is smaller than that of fast thinking.*

See Fig. 1, different reasoning strategies result in different reasoning progress achieved over time.

### 2.1 Reasoning Progress Modulation: A Universal View of Test-Time Scaling

There are mainly two components that must be modulated: **i) Thinking phase budget.** As discussed before, o1-like LRMs follow a "think-then-answer" paradigm. Therefore, modulating reasoning via scaling up or down the thinking phase budget is required. **ii) Slow thinking scheduling.** Within the thinking phase, the transition to slow thinking should also be modulated, thus increasing or reducing slow thinking according to a certain plan specified by users (*e.g.*, slow thinking first, and then fast thinking). With user-defined scheduling, the modulation of slow thinking transitions vary arbitrarily, ranging from sparse modulation—where little is adjusted—to dense modulation, where adjustments are frequent and extensive.

Based on the above analysis, we establish a unified perspective on test-time scaling and identify key limitations in existing approaches—namely, their failure to consider both reasoning schedule and overall thinking budget jointly. For instance, s1 modulates reasoning by sparsely increasing slow thinking (*i.e.*, adding two `wait` tokens), but overlooks broader thinking budget adjustments (Muennighoff et al., 2025). Conversely, Chain-of-Draft (CoD) reduces the thinking budget while neglecting the scheduling of slow thinking (Xu et al., 2025a). As a result, while LRMs are indirectly guided to reason more or less—sometimes achieving deeper reasoning or pruning unproductive thoughts—**we instead aim to explicitly and universally modulate the reasoning process by jointly considering both components**, as introduced next.

## 3 ALPHAONE

We introduce **ALPHAONE** ($\alpha 1$), a universal reasoning progress modulation framework for test-time scaling of LRMs, which is illustrated in Fig. 2. In the following, we first introduce $\alpha$ moment in Section 3.1, a moment that the thinking phase budget is scaled at least $\alpha\times$. In Section 3.2 and Section 3.3, we detail how we modulate slow thinking scheduling pre-$\alpha$ moment and modulating fast thinking encouragement post-$\alpha$ moment, respectively.

### 3.1 $\alpha$ Moment for Universal Modulation

To modulate the thinking phase budget, we scale the thinking phase by at least $\alpha\times$, where $\alpha > 1$ is a universal modulating parameter. Formally, given the average thinking phase token length $\overline{N}_{\text{think}} > 0$ generated by an LRM, we scale the thinking phase token length to $\alpha\overline{N}_{\text{think}}$, where the moment when the generated token length reaches $\alpha\overline{N}_{\text{think}}$ is dubbed as "$\alpha$ *moment*". In addition to scaling the thinking phase, we modulate the thinking phase via slow thinking scheduling before the $\alpha$ moment, thus achieving both controllable and scalable thinking. Note that $\alpha$ moment does not represent the new thinking phase transitioning moment, because the thinking phase typically continues after $\alpha$ moment, which we will elaborate later.

### 3.2 Pre-$\alpha$ Moment Modulation

Following previous works (Yang et al., 2025c; Muennighoff et al., 2025), we activate slow thinking before $\alpha$ moment via appending "`wait`" after a frequently co-generated structural delimiters "`\n\n`". Moreover, the activation of slow thinking is conducted following a user-specified scheduling plan, such as slow thinking, then fast thinking.

**Stochastic Reasoning Transitioning** $\alpha 1$ achieves such scheduling by modeling the activation of slow thinking as a Bernoulli stochastic process. Specifically, $\alpha 1$ appends "`wait`" following $\text{Bernoulli}(\boldsymbol{p}_{\texttt{wait}})$. Let $t = 0, 1, \ldots, T_m$ be the timestamps of generated tokens before $\alpha$ moment, where $T_m = \alpha\overline{N}_{\text{think}}$ represents the timestamp of $\alpha$ moment. $\boldsymbol{p}_{\texttt{wait}}$ is determined by a user-specified scheduling function $\mathcal{S}(t)$,

$$\boldsymbol{p}_{\texttt{wait}} := \mathcal{S}(t), t = 0, 1, \ldots, T_m. \quad (1)$$

$\mathcal{S}(t)$ can be an arbitrary function, such as linear annealing. $\alpha 1$ adopts linear annealing, which we find the most effective and efficient (See Section 4.3.1).

### 3.3 Post-$\alpha$ Moment Modulation

While an LRM significantly increases slow thinking through pre-$\alpha$ modulation, this extended thinking phase often exhibits *slow thinking inertia*, making it difficult to transition back to fast thinking. Notably, without post-$\alpha$ moment modulation, the LRM substantially reduces the likelihood of generating "`</think>`". Furthermore, inserting a few "`</think>`" tokens does not effectively overcome the inertia, failing to fully restore fast thinking.

**Deterministic Reasoning Termination** After the $\alpha$ moment, we guide $\alpha 1$ to transition into fast reasoning by disabling further slow thinking. Specifically, any generated slow reasoning transition token "`wait`" is replaced with "`</think>`" to explicitly mark the end of the thinking phase, reinforcing a shift to fast thinking before entering the answering phase. This deterministic termination strategy allows $\alpha 1$ to conclude reasoning naturally and consistently, enabling more efficient test-time scaling.

## 4 Experiments

### 4.1 Experimental Setup

**Benchmarks** To comprehensively evaluate the reasoning capability of LRMs, we conduct systematic evaluations on six benchmarks covering three reasoning categories: i) mathematical reasoning, including AIME 2024 (AIME24) (Mathematical Association of America, 2024), , AMC23 (AI-MO, 2024), and Minerva-Math (Minerva) (Lewkowycz et al., 2022); ii) code generation, including LiveCodeBench (LiveCode) (Jain et al., 2025); iii) scientific problems, including OlympiadBench (Olympiad) (He et al., 2024). We report the problem-solving accuracy by average Pass@1 (%), and the average number of generated tokens.

**Base Models** We use three o1-like open-source LRMs as the base model, including DeepSeek R1 distilled DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI et al., 2025), as well as a recently larger LRM Qwen QwQ 32B (Qwen Team, 2025).

**Implementations** Without additional specifications, we use a temperature of 0.6, top-$p$ of 0.95, and the maximum token length is set to 8192. We set $\alpha$ as 1.4, and we obtain the average thinking phase token length generated by an LRM on any benchmark by randomly sampling 10 test questions and averaging the generated token length before benchmarking. See more details in Section A.

Table 1: **Systematic comparison of reasoning results** on mathematical, coding, and science reasoning benchmarks with DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-Distill-Qwen-7B, and Qwen QwQ 32B. Additional results for other models are provided in Section B. P@1: Pass@1 (%); #Tk: number of generated tokens; $\overline{\Delta}_{\text{P@1}}$ (%): average Pass@1 result boost over the base model. *For a fair comparison, s1 (Muennighoff et al., 2025) directly applies budget forcing at test-time without supervised fine-tuning, which is same as CoD and our $\alpha$1 that are training-free.

| Method | MATHEMATICAL | | | | | | | | CODING | | SCIENCE | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AIME24 | | AMC23 | | Minerva | | MATH500 | | LiveCode | | Olympiad | | |
| | P@1 | #Tk | P@1 | #Tk | P@1 | #Tk | P@1 | #Tk | P@1 | #Tk | P@1 | #Tk | $\overline{\Delta}_{\text{P@1}}$ |
| ***DeepSeek-R1-Distill-Qwen-1.5B*** | | | | | | | | | | | | | |
| BASE | 23.3 | 7280 | 57.5 | 5339 | 32.0 | 4935 | 79.2 | 3773 | 17.8 | 6990 | 38.8 | 5999 | N/A |
| s1* | 26.7$_{+3.4}$ | 7798 | 57.5$_{+0.0}$ | 6418 | 31.6$_{-0.4}$ | 5826 | 78.2$_{-1.0}$ | 4733 | 17.0$_{-0.8}$ | 7025 | 38.5$_{-0.3}$ | 6673 | +0.15 |
| CoD | **30.0**$_{+6.7}$ | 6994 | 65.0$_{+7.5}$ | 5415 | 29.0$_{-3.0}$ | **4005** | **81.4**$_{+2.2}$ | **3136** | 20.3$_{+2.5}$ | 6657 | 40.6$_{+1.8}$ | 5651 | +2.95 |
| **$\alpha$1 (Ours)** | **30.0**$_{+6.7}$ | **5916** | **70.0**$_{+12.5}$ | **4952** | **34.2**$_{+2.2}$ | 4586 | 81.0$_{+1.8}$ | 3852 | **24.8**$_{+7.0}$ | **5426** | **45.5**$_{+6.7}$ | **4944** | **+6.15** |
| ***DeepSeek-R1-Distill-Qwen-7B*** | | | | | | | | | | | | | |
| BASE | 46.7 | 6648 | 82.5 | 4624 | 40.4 | 4191 | 87.6 | 3239 | 43.5 | 5885 | 50.4 | 5385 | N/A |
| s1* | 46.7$_{+0.0}$ | 7295 | 80.0$_{-2.5}$ | 5673 | 42.3$_{+1.9}$ | 6510 | **92.8**$_{+5.2}$ | 5848 | 44.0$_{+0.5}$ | 5979 | 54.2$_{+3.8}$ | 6007 | +1.48 |
| CoD | 43.3$_{-3.4}$ | **6078** | 87.5$_{+5.0}$ | **3594** | **43.4**$_{+3.0}$ | **2142** | 88.8$_{+1.2}$ | **2094** | 45.0$_{+1.5}$ | 5593 | 53.5$_{+3.1}$ | **4520** | +1.73 |
| **$\alpha$1 (Ours)** | **50.0**$_{+3.3}$ | 6827 | **90.0**$_{+7.5}$ | 4397 | 42.3$_{+1.9}$ | 4124 | 91.2$_{+3.6}$ | 4337 | **49.8**$_{+6.3}$ | **5067** | **55.7**$_{+5.3}$ | 4883 | **+4.65** |
| ***Qwen QwQ-32B*** | | | | | | | | | | | | | |
| BASE | 40.0 | 4058 | 77.5 | 2901 | **47.8** | 2199 | 90.2 | 1951 | 67.0 | 5092 | 53.6 | 3230 | N/A |
| s1* | 43.3$_{+3.3}$ | 4221 | 77.5$_{+0.0}$ | 3068 | 46.7$_{-1.1}$ | 2433 | **90.8**$_{+0.6}$ | 2218 | 66.5$_{-0.5}$ | 5260 | 55.1$_{+1.5}$ | 3454 | +0.63 |
| CoD | 46.7$_{+6.7}$ | 3959 | 80.0$_{+2.5}$ | 2400 | 47.4$_{-0.4}$ | 1464 | 90.6$_{+0.4}$ | **1421** | 66.8$_{-0.2}$ | **4984** | **57.2**$_{+3.6}$ | 2844 | +2.10 |
| **$\alpha$1 (Ours)** | **53.3**$_{+13.3}$ | **3141** | **87.5**$_{+10.0}$ | **2286** | 46.0$_{-1.8}$ | **1441** | 89.4$_{-0.8}$ | 1668 | **75.8**$_{+8.8}$ | 5824 | 56.1$_{+2.5}$ | **2504** | **+5.33** |

**Baselines** We compare our $\alpha$1 against the vanilla LRM and two training-free, test-time scaling baselines. i) BASE: The original LRM that transitions between slow and fast thinking ***automatically***, without any external modulation. ii) S1 (Muennighoff et al., 2025): A baseline that enforces a ***monotonically increasing*** slow thinking pattern by appending approximately two "`wait`" tokens near the end of the reasoning phase to prolong slow thinking. For a fair comparison, we apply S1 at test time without supervised fine-tuning used in its original implementation. iii) CHAIN OF DRAFT (COD) (Xu et al., 2025a): A baseline that enforces a ***monotonically decreasing*** slow thinking pattern by prompting the model to constrain each slow thinking step to no more than five words, thereby sharply reducing the thinking budget.

## 4.2 Main Results

Table 1 shows the systematic comparison results of our $\alpha$1 and baseline methods, and we observe: i) $\alpha$1 consistently yields a higher problem-solving accuracy than all baseline methods across all models and benchmarks. Notably, compared to the base model, $\alpha$1 improves the 1.5B LRM by a clear margin of **+6.15%**, while reducing nearly **14%** token length. This demonstrates both the effectiveness and efficiency of $\alpha$1. ii) Compared to baseline test-time scaling methods, including s1 and CoD, $\alpha$1 still achieves significantly better results. Specifically, the average accuracy boost over all benchmarks and models of $\alpha$1 is **+3.12%** and **+4.62%** higher than CoD and s1, respectively. iii) Surprisingly, we observe that while $\alpha$1 modulates reasoning densely without restrictions on reducing the thinking budget (instead, we use $\alpha > 1$ that increases the thinking budget), the average thinking phase token length generated by $\alpha$1 is only about **+4.4%** higher than the monotonically decreasing baseline CoD (4231 *vs.* 4053), which is about **+21.0%** more efficient than the monotonically increasing baseline s1 (4231 *vs.* 5357). This indicates that $\alpha$1 achieves more efficient reasoning than baselines, which we provide analysis later.

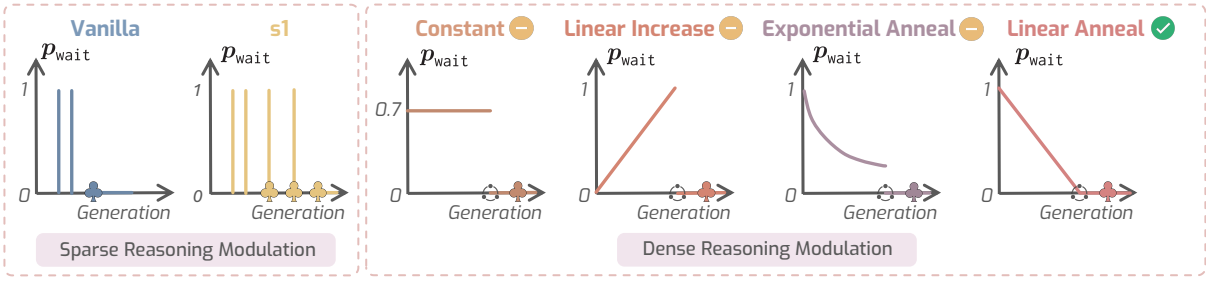
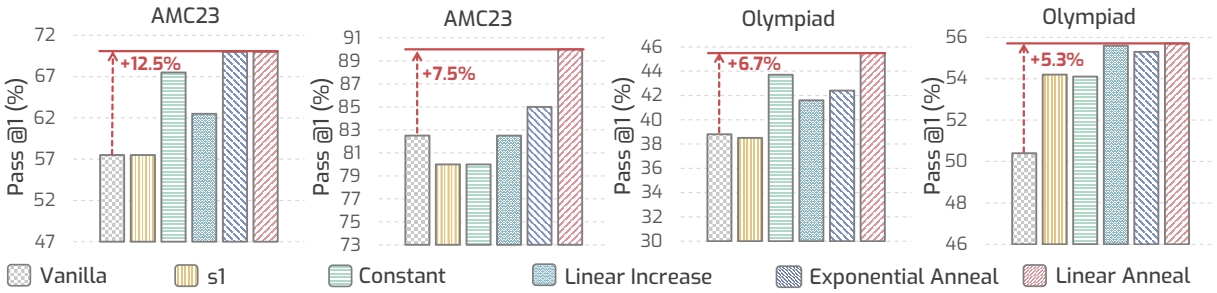



Figure 3: **Visualization of different scheduling strategies.** We detail the functions in Section 4.3.1. Here ♲ represents $\alpha$ moment, which we elaborate in Section 3.1, and ♣ denotes the end of the thinking phase.




(a) DeepSeek-R1-Distill-Qwen-1.5B (b) DeepSeek-R1-Distill-Qwen-7B (c) DeepSeek-R1-Distill-Qwen-1.5B (d) DeepSeek-R1-Distill-Qwen-7B

Figure 4: **Ablation study of different scheduling strategies** on (a-b) AMC23 and (c-d) OlympaidBench.

### 4.3 Analytic Results

In this section, we analyze $\alpha 1$ by systematically addressing the following five questions:

#### 4.3.1 What scheduling strategy is better?

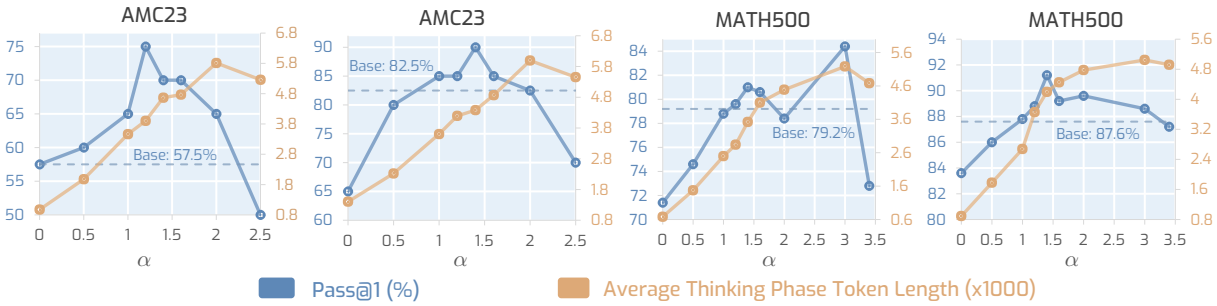
As shown in Fig. 3, we study four variants of scheduling strategies for $\mathcal{S}(t)$ defined in Eq. (1), where $T_m = \alpha \overline{N}_{\text{think}}$ represents the timestamp of $\alpha$ moment:

- **Constant:** $\mathcal{S}(t) := \boldsymbol{p}_{\text{constant}}$, where $\boldsymbol{p}_{\text{constant}} \in [0, 1]$ is a constant probability. This represents a consistently more slow thinking strategy, and the increase is large when $\boldsymbol{p}_{\text{constant}}$ is larger. Note that when $\boldsymbol{p}_{\text{constant}} = 0$ and $\alpha = 1$, it degenerates to vanilla reasoning models; and when $\boldsymbol{p}_{\text{constant}} < 0.1$ and $\alpha > 1$, it degenerates to s1-like model, where only about two "`wait`" are appended.
- **Linear increase:** $\mathcal{S}(t) := \frac{1}{T_m} t$, where $t = \{0, 1, \ldots, T_m\}$ and $\frac{1}{T_m} > 0$ indicates the increasing coefficient. This scheduling function indicates a fast-to-slow thinking strategy.
- **Exponential anneal:** $\mathcal{S}(t) := \exp(-\gamma t)$, where $t = \{0, 1, \ldots, T_m\}$ and $\gamma > 0$ is a hyperparameter that controls annealing speed (here we use $\gamma = 0.3$). This scheduling function indicates a slow-to-fast thinking strategy.
- **Linear anneal:** $\mathcal{S}(t) := -\frac{1}{T_m} t + 1$, where $-\frac{1}{T_m} < 0$ indicates the annealing coefficient. Its modulation is similar to exponential anneal scheduling.

Fig. 4 shows the results of $\alpha 1$ using these four different scheduling strategies. We observe: i) Linear anneal consistently yields the highest reasoning accuracy, indicating that the *slow thinking first, then fast thinking* is a better slow thinking scheduling strategy. ii) Similar to linear anneal, exponential anneal also follows an annealing slow thinking scheduling, where the improvement on the 1.5B model further demonstrates the efficacy of the slow thinking, then fast thinking strategy. However, such annealing scheduling may lead to an unstable performance boost compared to linear anneal.

#### 4.3.2 Can $\alpha$-moment scale the thinking phase budget?

Fig. 5 shows the results of $\alpha 1$ with different $\alpha$-moments determined by scaling $\alpha$ from 0 to a maximum value subject to the 8192 token length budget. We observe: i) $\alpha$-moment enables a *scalable thinking phase budgeting*. By scaling up $\alpha$, the average thinking phase token length is accordingly scaled up. ii) Interestingly, while the thinking phase is scaled up, there exists a trade-off between the optimal value of $\alpha$ and the resulting reasoning accu-




(a) DeepSeek-R1-Distill-Qwen-1.5B (b) DeepSeek-R1-Distill-Qwen-7B (c) DeepSeek-R1-Distill-Qwen-1.5B (d) DeepSeek-R1-Distill-Qwen-7B

Figure 5: **Scaling property of $\alpha$.** We scale $\alpha$ from 0 to the maximum value restricted by the maximum token length, and plot the corresponding reasoning Pass@1 and average thinking phase token length on AMC23 and MATH500.

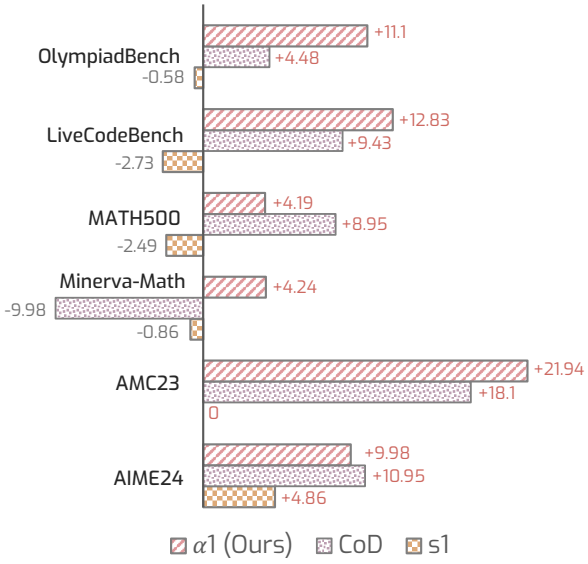



Figure 6: **Scaling efficiency analysis with REP** using Deepseek-R1-distill-Qwen-1.5B. The REP metric is introduced in Eq. (2).

racy. This indicates that monotonously increasing the thinking phase budget does not consistently bring better reasoning performance, and it is critical to find the optimal $\alpha$-moment that results in a satisfactory improvement.

#### 4.3.3 Does $\alpha$1 scale more efficiently?

To quantitatively evaluate how different methods trade off reasoning efficiency and accuracy, we introduce the $\mathcal{F}_{\text{REP}}(\mathcal{A}_{\text{method}}; \mathcal{A}_{\text{base}}, T_{\text{norm}})$ (Reasoning Efficiency-Performance, REP) metric. The REP metric is defined as:

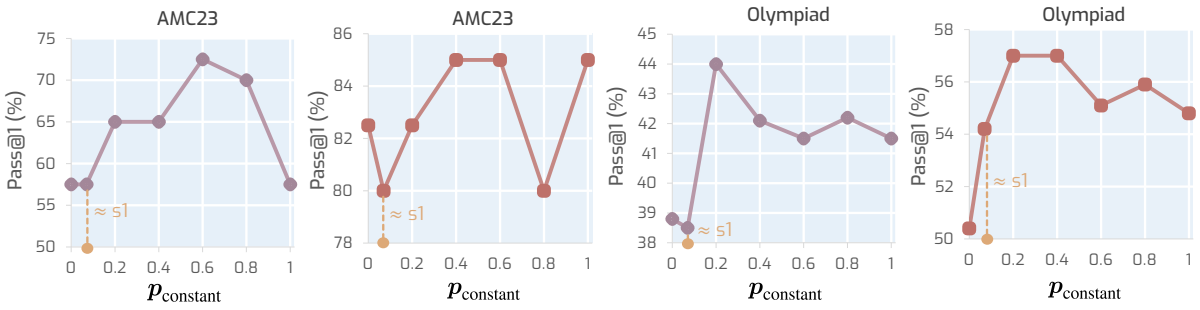
$$\mathcal{F}_{\text{REP}}(\mathcal{A}_{\text{method}}; \mathcal{A}_{\text{base}}, T_{\text{norm}}) = \frac{\mathcal{A}_{\text{method}} - \mathcal{A}_{\text{base}}}{T_{\text{norm}}} \tag{2}$$

where $\mathcal{A}_{\text{method}}$ and $\mathcal{A}_{\text{base}}$ denote the reasoning accuracy of the evaluated method and the base model, respectively. $T_{\text{norm}}$ is the normalized thinking phase token length, computed by dividing the current thinking phase token length by the maximum token length. Higher REP indicates stronger performance with better reasoning efficiency.

We report the REP of CoD, s1, and $\alpha$1 on six reasoning benchmarks with Deepseek-R1-distill-Qwen-1.5B. Fig. 6 shows that $\alpha$1 achieves higher REP on most benchmarks, indicating a more favorable balance between reasoning performance and efficiency. Notably, $\alpha$1 outperforms CoD by **+6.62** and s1 by **+11.68** on Olympiad-Bench, and exceeds CoD by **+14.22** on Minerva-Math.

#### 4.3.4 How frequent should slow thinking transitioning be?

$\alpha$1 modulate slow thinking transitioning via sampling from Bernoulli($\boldsymbol{p}_{\texttt{wait}}$), which leads to another question of how large should $\boldsymbol{p}_{\texttt{wait}}$ be that can bring a better result. To study this question, we use the constant scheduling function and scale $p_{\text{constant}}$ from 0 to 1 to increase the frequency of transitioning to slow thinking. This is because the constant scheduling is a sampling process with a certain probability, and the value of the probability determines how frequently the slow thinking transitioning token will be sampled. Fig. 7 shows the results, from which we observe: i) An extremely low or high frequency of transitioning to slow thinking brings unsatisfactory results (*e.g.*, $p_{\text{constant}} = 0.1$). Similar to the scaling of the thinking phase dedget (*e.g.*, modualting $\alpha$), the slow thinking frequency also needs to be carefully selected. ii) While an extremely dense or sparse slow thinking transitioning leads to unsatisfactory results, the reasoning performance is decent across a large range of $p_{\text{constant}}$, demonstrating that increasing slow thinking generally brings improved reasoning.




(a) DeepSeek-R1-Distill-Qwen-1.5B (b) DeepSeek-R1-Distill-Qwen-7B (c) DeepSeek-R1-Distill-Qwen-1.5B (d) DeepSeek-R1-Distill-Qwen-7B

Figure 7: **Scaling property of "`wait`" frequency** under constant scheduling on AMC23 and OlympiadBench. Increasing $p_{\text{constant}}$ leads to a higher frequency of yielding "`wait`" in the Bernoulli process $\text{Bernoulli}(\boldsymbol{p}_{\texttt{wait}})$.

Table 2: **Ablation study on post-$\alpha$ moment modulation**. Without post-$\alpha$ modulation represents our $\alpha$1 without the suppression of the slow thinking inertia after the $\alpha$ moment.

| Method | Post-$\alpha$ Moment Modulation | AIME24 | | AMC23 | |
|---|---|---|---|---|---|
| | | P@1 | #Tk | P@1 | #Tk |
| ***DeepSeek-R1-Distill-Qwen-1.5B*** | | | | | |
| BASE | N/A | 23.3 | 7280 | 57.5 | 5339 |
| **$\alpha$1 (Ours)** | × | 26.7 | 7929 | 47.5 | 6903 |
| **$\alpha$1 (Ours)** | ✓ | **30.0** | **5916** | **70.0** | **4951** |
| ***DeepSeek-R1-Distill-Qwen-7B*** | | | | | |
| BASE | N/A | 38.8 | **5999** | 82.5 | 4624 |
| **$\alpha$1 (Ours)** | × | 30.0 | 7666 | 75.0 | 5878 |
| **$\alpha$1 (Ours)** | ✓ | **50.0** | 6826 | **90.0** | **4397** |

#### 4.3.5 Is post-$\alpha$ moment modulation necessary?

Typical test-time scaling methods focus on the modulation of slow thinking within the thinking phase, while $\alpha$1 consists of a post-$\alpha$ moment modulation that encourages fast thinking. To validate its necessity of enforcing fast thinking in the end, we conduct an ablation study on utilizing the post-$\alpha$ moment modulation, shown in Table 2. We observe: i) Pre-$\alpha$ moment modulation of slow thinking is insufficient. When the post-$\alpha$ moment modulation is reduced to a single operation, the performance of $\alpha$1 significantly drops. This is because the increase of slow thinking during pre-$\alpha$ moment brings a *slow thinking inertia* (as discussed before in Section 3.3), leading to a slow thinking intensive reasoning. ii) By utilizing a post-$\alpha$ moment modulation, $\alpha$1 successfully ends in a fast thinking, which demonstrates the necessity of *combining both slow thinking and fast thinking*.

## 5 Related Works

### 5.1 Large Reasoning Models

Large Reasoning Models are rapidly emerging as a family of foundation models (Bommasani et al., 2021) that target human-level system-2 reasoning (Kahneman, 2011). Starting from OpenAI's o1 (Jaech et al., 2024) in 2024, numerous efforts follow this "thinking-then-answering" paradigm. Notably, o1-like Large Language Models (LLMs) can solve increasingly complex reasoning problems after a thorough chain of thoughts (Wei et al., 2022; Yao et al., 2023; Besta et al., 2024), such as the IMO competition. These advanced models are mainly developed via large-scale reinforcement learning (RL) to align human preference (Christiano et al., 2017; Schulman et al., 2017; Shao et al., 2024b; DeepSeek-AI et al., 2025), where a reward model is used to judge model answers (Uesato et al., 2022; Lightman et al., 2024). Notable efforts replicating o1's success include DeepSeek R1, Qwen QwQ, and Phi-4 (Abdin et al., 2024; DeepSeek-AI et al., 2025; Qwen Team, 2025), which typically utilize a special end-of-thinking token "`</think>`", after which a solution is output to the user. Recently, some researchers have explored applying RL during post-training fine-tuning, where promising results have been obtained (Chow et al., 2025; Qu et al., 2025; Zuo et al., 2025).

### 5.2 Reasoning with Test-Time Scaling

Reasoning with test-time scaling has recently become a useful strategy that empowers LLMs with a scalable reasoning capability at test time. The mainstream methods lie in two categories, *i.e.*, i) parallel scaling and ii) sequential scaling. The key idea of parallel scaling is Best-of-N (BoN) sampling,

where the best choice is selected using uncertainty criteria like self-consistency (Wang et al., 2023), reward model (Lightman et al., 2024; Cobbe et al., 2021), or perplexity (Fang et al., 2025). Specifically, one line of work focuses on sequence-level sampling (Cobbe et al., 2021; Gui et al., 2024; Wan et al., 2024a; Sun et al., 2024; Chow et al., 2025; Sessa et al., 2025; Amini et al., 2025; Zhou et al., 2025; Zeng et al., 2025; Kang et al., 2025), while another line of work utilizes token-/step- level sampling including beam-/tree- based searching (Kool et al., 2019; Xie et al., 2023; Zhang et al., 2023a; Hao et al., 2023; Qiu et al., 2024; Gao et al., 2024; Yu et al., 2024; Wan et al., 2024b; Chen et al., 2024a). Meanwhile, sequential scaling enhances or reduces slow thinking. This technique typically relies on an iterative refinement and revision of answers generated by LLMs themselves (Zelikman et al., 2022; Madaan et al., 2023) or external feedback (Chen et al., 2024c; Gou et al., 2024; Huang et al., 2024a; Kamoi et al., 2024; Zheng et al., 2024; Liao et al., 2025). Following this line of research, recent works have been devoted to addressing the underthinking and overthinking issues of modern LRMs via reinforcing (Muennighoff et al., 2025) and restricting (Xu et al., 2025a,b) slow thinking, respectively. Given the non-conflict between parallel scaling and sequential scaling, there exists another group of hybrid scaling methods that leverage both strategies (Li et al., 2025; Zeng et al., 2025).

## 6 Conclusions

In this paper, we study the problem of test-time scaling of large reasoning models with our framework, ALPHAONE ($\alpha$1). ALPHAONE starts from a universal view of reasoning modulation targeting two key aspects: thinking phase budgeting and slow thinking scheduling. We introduce $\alpha$-moment, which is determined by $\alpha$ that scales the thinking phase budget by at least $\alpha\times$. ALPHAONE operates by scheduling slow thinking before the $\alpha$-moment, and fast thinking after the $\alpha$-moment that eliminates slow thinking inertia. Using ALPHAONE, we investigate the test-time scaling from various aspects, including the overall slow and fast thinking transitioning plan, thinking phase budget scaling property, and efficiency of test-time scaling, *etc*. Insightful findings are obtained, *e.g.*, slow thinking first, then fast thinking leads to better reasoning capability of LRMs.

## Limitations

While ALPHAONE provides a universal view of test-time scaling of LRMs, and a significant performance boost has been achieved, we identify some possible limitations as follows. i) ALPHAONE targets at o1-style LRMs, where tokens such as "`wait`" is proved effective in transitioning into slow thinking. However, future LRMs may use a different slow thinking transitioning strategy, leading to a possibility of incompatibility with our framework. ii) ALPHAONE relies on $\alpha$-moment throughout reasoning modulation, and the average thinking phase token length is typically required. This paper obtains it by first running LRMs on 10 random samples, which requires marginal cost. However, in case that no test questions are available, ALPHAONE can only rely on an empirical thinking phase length that may be suboptimal.

## Broader Impact

This work targets complex reasoning problems with LRMs, which we believe will lead to no ethical concerns. However, since LRMs are modern variants of LLMs, any ethical concerns raised by LLMs can potentially exist.

## Acknowledgments

Huan Zhang is partially funded by the AI2050 program at Schmidt Sciences (AI2050 Early Career Fellowship). The authors thank Heng Dong for his valuable suggestions on this project.

## References

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, and 1 others. 2025. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *CoRR*, abs/2412.08905.

AI-MO. 2024. AIMO Validation Dataset - AMC. `https://huggingface.co/datasets/AI-MO/aimo-validation-amc`. Accessed: 2025-05-19.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Afra Amini, Tim Vieira, Elliott Ash, and Ryan Cotterell. 2025. Variational best-of-n alignment. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17682–17690. AAAI Press.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, and 34 others. 2021. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024a. Alphamath almost zero: Process supervision without process. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

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. 2024b. Do NOT think that much for 2+3=? on the overthinking of o1-like llms. *CoRR*, abs/2412.21187.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2024c. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Aviral Kumar, Rishabh Agarwal, Sridhar Thiagarajan, Craig Boutilier, and Aleksandra Faust. 2025. Inference-aware fine-tuning for best-of-n sampling in large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

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 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, Xiangwen Kong, Xiangyu Zhang, Kaisheng Ma, and Li Yi. 2024. DreamLLM: Synergistic multimodal comprehension and creation. In *The Twelfth International Conference on Learning Representations*.

Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. 2025. What is wrong with perplexity for long-context language modeling? In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Li Fei-Fei. 2023. *The Worlds I See: Curiosity, Exploration, and Discovery at the Dawn of AI*. Flatiron books: a moment of lift book.

Yichao Fu, Junda Chen, Yonghao Zhuang, Zheyu Fu, Ion Stoica, and Hao Zhang. 2025. Reasoning without self-doubt: More efficient chain-of-thought through certainty probing. In *ICLR 2025 Workshop on Foundation Models in the Wild*.

Zitian Gao, Boye Niu, Xuzheng He, Haotian Xu, Hongzhang Liu, Aiwei Liu, Xuming Hu, and Lijie Wen. 2024. Interpretable contrastive monte carlo tree search reasoning. *CoRR*, abs/2410.01707.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. CRITIC: large language models can self-correct with

tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Lin Gui, Cristina Garbacea, and Victor Veitch. 2024. Bonbon alignment for large language models and the sweetness of best-of-n sampling. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8154–8173. Association for Computational Linguistics.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *CoRR*, abs/2412.06769.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3828–3850. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024a. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. 2024b. O1 replication journey - part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *CoRR*, abs/2411.16489.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, and 80 others. 2024. Openai o1 system card. *CoRR*, abs/2412.16720.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. 2025. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*.

Daniel Kahneman. 2011. *Thinking, fast and slow*. macmillan.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When can llms *Actually* correct their own mistakes? A critical survey of self-correction of llms. *Trans. Assoc. Comput. Linguistics*, 12:1417–1440.

Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. Scalable best-of-n selection for large language models via self-certainty. *CoRR*, abs/2502.18581.

Wouter Kool, Herke van Hoof, and Max Welling. 2019. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3499–3508. PMLR.

Seongyun Lee, Sue Hyun Park, Yongrae Jo, and Minjoon Seo. 2024. Volcano: Mitigating multimodal hallucination through self-feedback guided revision. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 391–404. Association for Computational Linguistics.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Dacheng Li, Shiyi Cao, Chengkun Cao, Xiuyu Li, Shangyin Tan, Kurt Keutzer, Jiarong Xing, Joseph E. Gonzalez, and Ion Stoica. 2025. S*: Test time scaling for code generation. *CoRR*, abs/2502.14382.

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. *arXiv preprint arXiv:2501.19324*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Mathematical Association of America. 2024. American Invitational Mathematics Examination – AIME. *American Invitational Mathematics Examination – AIME 2024*. Accessed: 2025-05-15.

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

Xuying Ning, Dongqi Fu, Tianxin Wei, Wujiang Xu, and Jingrui He. 2025. Graph4mm: Weaving multimodal learning with structural information. In *Forty-second International Conference on Machine Learning*.

OpenAI. 2024. Introducing gpt-4o and more tools to chatgpt free users.

OpenAI. 2025. Thinking with images. Accessed: 2025-05-25.

Xiao Pu, Michael Saxon, Wenyue Hua, and William Yang Wang. 2025. Thoughtterminator: Benchmarking, calibrating, and mitigating overthinking in reasoning models. *arXiv preprint arXiv:2504.13367*.

Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. 2024. Shapellm: Universal 3d object understanding for embodied interaction. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XLIII*, volume 15101 of *Lecture Notes in Computer Science*, pages 214–238. Springer.

Zekun Qi, Wenyao Zhang, Yufei Ding, Runpei Dong, Xinqiang Yu, Jingwen Li, Lingyun Xu, Baoyu Li, Xialin He, Guofan Fan, Jiazhao Zhang, Jiawei He, Jiayuan Gu, Xin Jin, Kaisheng Ma, Zhizheng Zhang, He Wang, and Li Yi. 2025. Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation. *CoRR*, abs/2502.13143.

Jiahao Qiu, Yifu Lu, Yifan Zeng, Jiacheng Guo, Jiayi Geng, Huazheng Wang, Kaixuan Huang, Yue Wu, and Mengdi Wang. 2024. Treebon: Enhancing inference-time alignment with speculative tree-search and best-of-n sampling. *CoRR*, abs/2410.16033.

Yuxiao Qu, Matthew Y. R. Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. 2025. Optimizing test-time compute via meta reinforcement fine-tuning. *CoRR*, abs/2503.07572.

Qwen Team. 2025. Preview of qwen qwen1.5-32b. https://qwenlm.github.io/blog/qwq-32b-preview/. Accessed: 2025-03-20.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Pier Giuseppe Sessa, Robert Dadashi-Tazehozi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shahriari, Sarah Perrin, Abram L. Friesen, Geoffrey Cideron, Sertan Girgin, Piotr Stanczyk, Andrea Michi, Danila Sinopalnikov, Sabela Ramos Garea, Amélie Héliou, Aliaksei Severyn, Matthew Hoffman, Nikola Momchev, and Olivier Bachem. 2025. BOND: aligning llms with best-of-n distillation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024a. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024b. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. 2025. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms. *arXiv preprint arXiv:2505.00127*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Ben Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *CoRR*, abs/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. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chameleon Team. 2024. Chameleon: Mixed-modal early-fusion foundation models. *CoRR*, abs/2405.09818.

Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Y. Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *CoRR*, abs/2211.14275.

Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. 2024a. Dynamic self-consistency: Leveraging reasoning paths for efficient LLM sampling. *CoRR*, abs/2408.17017.

Ziyu Wan, Xidong Feng, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2024b. Alphazero-like tree-search can guide large language model decoding and training. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Han Wang, Gang Wang, and Huan Zhang. 2024. Steering away from harm: An adaptive approach to defending vision language model against jailbreaks. *CoRR*, abs/2411.16721.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Thoughts are all over the place: On the underthinking of o1-like llms. *CoRR*, abs/2501.18585.

Haoran Wei, Youyang Yin, Yumeng Li, Jia Wang, Liang Zhao, Jianjian Sun, Zheng Ge, and Xiangyu Zhang. 2024. Slow perception: Let's perceive geometric figures step-by-step. *CoRR*, abs/2412.20631.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yana Wei, Liang Zhao, Kangheng Lin, En Yu, Yuang Peng, Runpei Dong, Jianjian Sun, Haoran Wei, Zheng Ge, Xiangyu Zhang, and Vishal M. Patel. 2025. Perception in reflection. *CoRR*, abs/2504.07165.

Penghao Wu and Saining Xie. 2024. V*: Guided visual search as a core mechanism in multimodal llms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13084–13094. IEEE.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Qizhe Xie. 2023. Self-evaluation guided beam search for reasoning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. 2024. Llava-critic: Learning to evaluate multimodal models. *CoRR*, abs/2410.02712.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025a. Chain of draft: Thinking faster by writing less. *CoRR*, abs/2502.18600.

Yuhui Xu, Hanze Dong, Lei Wang, Doyen Sahoo, Junnan Li, and Caiming Xiong. 2025b. Scalable chain of thoughts via elastic reasoning. *arXiv preprint arXiv:2505.05315*.

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*.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. 2025b. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *CoRR*, abs/2502.09560.

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*.

Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. 2025d. Towards thinking-optimal scaling of test-time compute for LLM reasoning. *CoRR*, abs/2502.18080.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

En Yu, Kangheng Lin, Liang Zhao, Jisheng Yin, Yana Wei, Yuang Peng, Haoran Wei, Jianjian Sun, Chunrui Han, Zheng Ge, Xiangyu Zhang, Daxin Jiang, Jingyu Wang, and Wenbing Tao. 2025. Perception-r1: Pioneering perception policy with reinforcement learning. *CoRR*, abs/2504.07954.

Fei Yu, Anningzhe Gao, and Benyou Wang. 2024. Ovm, outcome-supervised value models for planning in mathematical reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 858–875. Association for Computational Linguistics.

Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. 2025. Give me fp32 or give me death? challenges and solutions for reproducible reasoning. *arXiv preprint arXiv:2506.09501*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Yunhua Zhou, and Xipeng Qiu. 2025. Revisiting the test-time scaling of o1-like models: Do they truly possess test-time scaling capabilities? *CoRR*, abs/2502.12215.

Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B. Tenenbaum, and Chuang Gan. 2023a. Planning with large language models for code generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023b. Evaluating the performance of large language models on GAOKAO benchmark. *CoRR*, abs/2305.12474.

Jingnan Zheng, Han Wang, An Zhang, Tai D. Nguyen, Jun Sun, and Tat-Seng Chua. 2024. Ali-agent: Assessing llms' alignment with human values via agent-based evaluation. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Zhi Zhou, Tan Yuhao, Zenan Li, Yuan Yao, Lan-Zhe Guo, Xiaoxing Ma, and Yu-Feng Li. 2025. Bridging internal probability and self-consistency for effective and efficient LLM reasoning. *CoRR*, abs/2502.00511.

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. 2025. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yuxin Zuo, Kaiyan Zhang, Shang Qu, Li Sheng, Xuekai Zhu, Biqing Qi, Youbang Sun, Ganqu Cui, Ning Ding, and Bowen Zhou. 2025. Ttrl: Test-time reinforcement learning. *CoRR*, abs/2504.16084.

# APPENDIX



## A Additional Implementation Details

### A.1 Computaional Budget

We used 8 NVIDIA L40S GPUs and 4 NVIDIA A100 80GB GPUs for the experiments.

### A.2 Hyper-parameters & Parameters

For reproducibility, we provide the complete set of average thinking phase token length $\overline{N}_{\text{think}}$ in Table 3, which are obtained by randomly sampling 10 test questions on each benchmark and averaging the generated token lengths. Since the effective range of $\alpha$ observed in Figure 5 is relatively broad, practical implementations can tolerate variance in this measurement.

### A.3 Benchmarks

**AIME 2024** The AIME 2024 dataset is a specialized benchmark collection consisting of 30 problems from the 2024 American Invitational Mathematics Examination (Mathematical Association of America, 2024). These problems cover core secondary-school mathematics topics such as arithmetic, combinatorics, algebra, geometry, number theory and probability. The collection places rigorous demands on both solution accuracy and conceptual depth.

**AMC 2023** The AMC 2023 dataset consists of 40 problems selected from the AMC 12A and 12B contests. These exams are sponsored by the Mathematical Association of America and target U.S. students in grade 12 and below, featuring challenges in algebra, geometry, number theory, and combinatorics (AI-MO, 2024).

**Minerva Math** Minerva Math (Lewkowycz et al., 2022) consists of 272 undergraduate-level STEM problems harvested from MIT's OpenCourseWare. These problems span solid-state chemistry, information and entropy, differential equations, and special relativity. Each includes a clearly delineated answer—191 verifiable by numeric checks and 81 by symbolic solutions. The benchmark is specifically designed to evaluate multi-step scientific reasoning capabilities in language models.

**MATH500** MATH500 comprises a selection of 500 problems extracted from the MATH benchmark (Lightman et al., 2024). The collection covers a range of high-school mathematics domains, including Prealgebra, Algebra and Number Theory. To ensure comparability with prior work, we use the exact problem set originally curated by OpenAI for evaluation.

**LiveCodeBench** LiveCodeBench (Jain et al., 2025) is a contamination-free benchmark for evaluating large language models on code. The suite is continuously updated, gathering new problems over time. It currently comprises 400 Python programming tasks released between May 2023 and March 2024, each paired with test samples for correctness verification. Beyond basic code generation, LiveCodeBench also measures advanced capabilities such as self-repair, code execution and test-output prediction.

Table 3: **Average thinking phase token length** $\overline{N}_{\text{think}}$ across different benchmarks. The results are obtained by running LRMs on randomly sampled 10 samples.

| Model | AIME24 | AMC23 | Minerva | MATH500 | LiveCode | Olympiad |
|---|---|---|---|---|---|---|
| DeepSeek-R1-Distill-Qwen-1.5B | 4130 | 3303 | 3101 | 2435 | 2172 | 3417 |
| DeepSeek-R1-Distill-Qwen-7B | 4751 | 3243 | 3064 | 2352 | 3120 | 3330 |
| Qwen QwQ-32B | 2597 | 2124 | 1710 | 1493 | 4915 | 2052 |

Table 4: **Additional reasoning results.** P@1: Pass@1 (%); #Tk: number of generated tokens.

| Method | MATHEMATICAL | | | | SCIENCE | |
|---|---|---|---|---|---|---|
| | AIME24 | | AMC23 | | Olympiad | |
| | P@1 | #Tk | P@1 | #Tk | P@1 | #Tk |
| ***Microsoft Phi4-reasoning*** | | | | | | |
| BASE | 63.3 | 5677 | 92.5 | **2858** | 60.1 | 4174 |
| **$\alpha$1 (Ours)** | **66.7** | **5532** | **95.0** | 2863 | **62.5** | **3786** |
| ***DeepSeek-R1-Distill-Llama-8B*** | | | | | | |
| BASE | 26.7 | 7184 | 70.0 | 5011 | 45.6 | 5757 |
| **$\alpha$1 (Ours)** | **33.3** | **7022** | **80.0** | **4282** | **52.9** | **4993** |

**OlympiadBench** OlympiadBench (He et al., 2024) consists of 8,476 Olympiad-level problems that evaluate mathematical and physical reasoning in AI systems. It features a wide difficulty range, open-ended problem generation, expert solution annotations, detailed difficulty labels, and multilingual coverage. The subset we use in our paper contains 675 open-ended, text-only math competition problems in English.

# B Additional Results

## B.1 Additional Models Results

To further demonstrate the generalization capability of $\alpha$1, we conduct experiments on two additional model families, including Phi4-reasoning (Abdin et al., 2025) from Microsoft and DeepSeek-R1-Distill-Llama-8B, across math and science benchmarks. Table 4 demonstrates that our method consistently achieves large gains.

## B.2 Scheduling Strategy

In addition to the results in Fig. 4 tested on AMC23 and Olympiad, we also show the results tested on AIME24 in Fig. 8. From the results, we observe that the linear increase consistently yields the best performance, which aligns with our previous observation. This further provides evidence that slow-then-fast thinking is an efficient slow-thinking scheduling strategy.

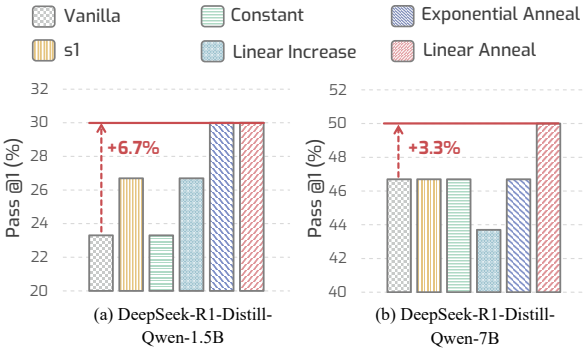



Figure 8: **Ablation study of different scheduling strategies** on AIME24.

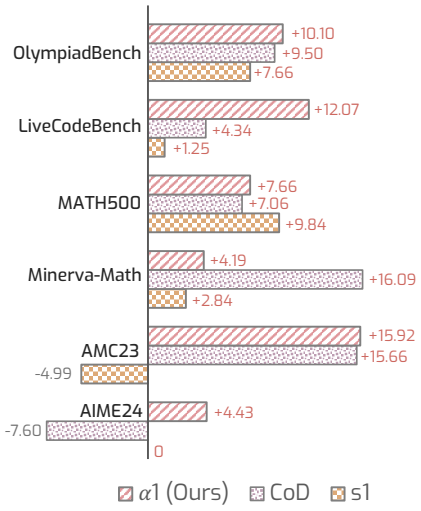



Figure 9: **Scaling efficiency analysis with REP** using Deepseek-R1-distill-Qwen-7B.

## B.3 Scaling Efficiency Analysis

As shown in Fig. 9, $\alpha$1 consistently achieves positive REP with Deepseek-R1-distill-Qwen-7B, demonstrating stable gains over the base model. Similar to Fig. 6, it outperforms CoD and s1 across nearly all benchmarks, particularly on LiveCodeBench and AIME24.

Table 5: **Cross-linguistic generalization results** with DeepSeek-R1-Distill-Qwen-1.5B.

| Method | GaoKao 2024 | | MGSM | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *Chinese* | | *French* | | *German* | | *Russian* | | *Japanese* | |
| | P@1 | #Tk | P@1 | #Tk | P@1 | #Tk | P@1 | #Tk | P@1 | #Tk |
| BASE | 65.9 | 4666 | 49.2 | **577** | 33.6 | 607 | 48.0 | 1751 | 28.8 | **966** |
| **$\alpha$1 (Ours)** | **69.2** | **4116** | **50.8** | 601 | **37.6** | **552** | **56.0** | **1650** | **30.4** | 1130 |

Table 6: **Ablation study on different transitioning tokens** on AMC23 (8192).

| Transitioning Token | Category | Deepseek-1.5B | |
|---|---|---|---|
| | | P@1 | #Tk |
| BASE (For Reference) | N/A | 57.5 | 5339 |
| "Wait," | Slow Thinking | 70.0 | 4952 |
| "Hmm," | Slow Thinking | **72.5** | **4793** |
| "Alternatively," | Slow Thinking | 70.0 | 5318 |
| "Maybe," | Continuation | 62.5 | 5380 |
| "Then," | Continuation | 65.0 | 5050 |
| "But," | Contrastive | 60.0 | 5763 |
| "However," | Contrastive | 55.0 | 5902 |
| "Though," | Contrastive | 55.0 | 5494 |

## B.4 Cross-linguistic Generalization

We have conducted ablations on cross-linguistic generalization across five languages, including Chinese, French, German, Russian, and Japanese on GaoKao 2024 (Zhang et al., 2023b) and MGSM (Shi et al., 2023) benchmarks. The results demonstrate the superior cross-linguistic generalization capability of $\alpha$1 in Table 5. Notably, on MGSM, $\alpha$1 shows substantial gains, with a +4.0% increase for German and an +8.0% improvement for Russian.

## B.5 Transitioning Tokens

We provide an ablation study on different transitioning tokens on the AMC23 with DeepSeek-R1-Distill-Qwen-1.5B. As illustrated in Table 6, the empirical results show that slow thinking transitioning tokens like "Wait", "Hmm", and "Alternatively" generally improve both accuracy and reasoning efficiency, though their effectiveness varies by model. Continuation tokens ("Maybe", "Then") offer minor gains, while contrastive tokens ("But", "However", "Though") often disrupt reasoning and reduce performance, especially with "However" and "Though".

Table 7: **Attention analysis** on slow thinking transitioning. The values are attention after substituting the original token after $i$-th "\n\n" with "Wait" toward two parts: the user-provided instruction (Question Part) and the intermediate reasoning steps (Reasoning Part). We report results on the DeepSeek-R1-Distill-Qwen-1.5B model on the AMC23 dataset. Special tokens such as "<|begin_of_sentence|>" are excluded from both the question and the reasoning process, so the combined attention does not sum to 1.0.

| $i$-th "\n\n" | Question Part | | Reasoning Part | |
|---|---|---|---|---|
| | Base | $\alpha$1 | Base | $\alpha$1 |
| 2 | 0.1944 | 0.1773 | 0.5016 | 0.5882 |
| 4 | 0.1389 | 0.1206 | 0.6281 | 0.6643 |
| 6 | 0.0882 | 0.0877 | 0.6444 | 0.6930 |
| 8 | 0.0864 | 0.0762 | 0.6168 | 0.7018 |
| 10 | 0.0792 | 0.0738 | 0.6984 | 0.7209 |
| 12 | 0.0705 | 0.0752 | 0.6760 | 0.7185 |
| 14 | 0.0689 | 0.0682 | 0.6360 | 0.7240 |

## B.6 Slow Thinking Transitioning Analysis

In this section, we analyze the quantitative impact of the "Wait" token on attention distributions in the last Transformer layer. This analysis is useful in revealing the dynamics of LLMs during inference, which intuitively improves the understanding of the method and serves as a good alternative for pure theoretical analysis. Specifically, we analyze how substituting the original token after $i$-th "\n\n" with "Wait" influences the model's attention toward two parts: the user-provided instruction (question part) and the intermediate reasoning steps (reasoning part). Results are shown in Table 7. When varying the position at which "\n\n" is inserted, the empirical results show that this token consistently shifts attention toward the previously generated reasoning steps. This likely promotes greater *self-reflection on earlier parts* of the solution and enhances the overall quality of the generated answers.

Table 8: **Formatting idiosyncrasies sensitivity** on three variants of prompts. The three variants of prompts are defined in Section B.7. We report the P@1 (#Tk) of the DeepSeek-R1-Distill-Qwen-1.5B on AMC23 and Olympiad benchmarks.

| | **Standard** | | **Variant A** | | **Variant B** | |
|---|---|---|---|---|---|---|
| | AMC23 | Olympiad | AMC23 | Olympiad | AMC23 | Olympiad |
| Base | 57.5 (5339) | 38.8 (5999) | 55.0 (5410) | 37.5 (6028) | 62.5 (5270) | 38.4 (6106) |
| **$\alpha$1 (Ours)** | **70.0 (4952)** | **45.5 (4944)** | **65.0 (5161)** | **43.9 (4995)** | **72.5 (5075)** | **45.3 (5037)** |

### B.7 Formatting Idiosyncrasies Sensitivity Analysis

In this section, we study the sensitivity of $\alpha$1 to formatting idiosyncrasies or prompt design. In addition to the standard prompt from the official technical report that is used in this work, we have conducted additional experiments comparing it with two variants: one adding irrelevant distractions, and another adding explicit reasoning instructions, listed as follows,

- **Standard:** `Please reason step by step, and put your final answer within \\boxed{}`
- **Variant A:** `Please reason step by step, and put your final answer within \\boxed{}. The AMC 2023 dataset consists of 40 problems selected from two challenging mathematics competitions. /OlympiadBench consists of 8,476 Olympiad-level problems that evaluate mathematical and physical reasoning.`
- **Variant B:** `You are a helpful assistant. Your role as an assistant involves thoroughly exploring questions through a systematic thinking process before providing the final precise and accurate solutions. Please reason step by step, and put your final answer within \\boxed{}`

The results are shown in Table 8. We observe: i) Modifying the prompts brings a performance drop or boost on the base model, where variant A leads to -2.5% drop while variant B brings +5.0% improvement on AMC23. ii) Regardless of the prompt variant, $\alpha$1 consistently improves the base model by a large margin. Specifically, $\alpha$1 improves the baseline by +10.0% on AMC23 with both variants. On Olympiad-Bench, $\alpha$1 archives a performance boost of +6.4% and +6.9% with variant A and B, respectively.

Table 9: **Slow thinking inertia analysis** with different number of deterministic terminations. The results are obtained with the DeepSeek-R1-Distill-Qwen-1.5B model, and we report the ratio of problems that remain in the thinking phase after different numbers (No.) of deterministic termination.

| **No.** | **AIME24** | **AMC23** | **Minerva** | **MATH500** | **LiveCode** | **Olympiad** |
|---|---|---|---|---|---|---|
| 1 | 96.7% | 75.0% | 78.7% | 45.0% | 92.8% | 78.9% |
| 2 | 90.0% | 67.5% | 70.2% | 39.4% | 87.8% | 72.6% |
| 3 | 60.0% | 30.0% | 24.3% | 12.8% | 4.3% | 39.3% |
| 4 | 10.0% | 5.0% | 2.6% | 1.2% | 0.2% | 6.8% |
| 5 | 3.3% | 0.0% | 0.7% | 0.0% | 0.0% | 1.2% |

### B.8 Slow Thinking Inertia Phenomenon Analysis

As stated before in Section 3.3, LRMs tend to have a slow thinking inertia issue. After the pre-$\alpha$ modulation phase, the model often continues slow thinking, which can severely affect accuracy and efficiency. When we enforce deterministic termination with a single "`</think>`", the model typically does not immediately transition to the answer phase but continues reasoning, as evidenced by the occurrence of slow-reasoning transitioning tokens "`Wait`" and semantically progressive thoughts. Repeated deterministic termination eventually forces the model to complete its remaining reasoning in just a few tokens before finally entering the answer phase.

In Table 9, we quantify the ratio of problems that remain in the thinking phase after $i$-th deterministic termination. For example, after the first termination, the model remains in the thinking phase on most problems, indicating that multiple terminations are generally required to conclude the reason-

Table 10: **Normalized REP metric** results. The results are obtained with DeepSeek-R1-Distill-Qwen-1.5B. AVG indicates the global mean REP across all evaluated benchmarks.

| | AIME24 | AMC23 | Minerva | MATH500 | LiveCode | Olympiad | AVG |
|---|---|---|---|---|---|---|---|
| Base | 8.60 | 13.35 | -2.20 | 3.55 | 6.51 | 5.00 | N/A |
| s1 | -3.74 | -13.35 | +1.34 | -6.04 | -9.24 | -5.58 | -0.30 |
| CoD | **+2.35** | +4.75 | -7.78 | **+5.40** | +2.92 | -0.52 | +6.99 |
| **$\alpha$1 (Ours)** | +1.38 | **+8.59** | **+6.44** | +0.64 | **+6.32** | **+6.10** | **+10.71** |

ing process. Note that Table 9 shows the results of the DeepSeek-R1-Distill-Qwen-1.5B model, and we also observe similar patterns on larger models like QwQ-32B.

### B.9 REP Metric Analysis

To better understand the proposed REP metric, we provide per-task baseline normalization and global mean normalization of the REP metric on DeepSeek-R1-Distill-Qwen-1.5B. Table 10 shows the results. Across these two normalized metrics, $\alpha$1 consistently achieves higher values, indicating a more favorable balance between reasoning performance and efficiency. Notably, $\alpha$1 exceeds the task average by +8.59 on AMC23 under the per-task baseline normalization and reaches +10.71 under the global mean normalization.

### B.10 Additional Results with More Rollouts

According to Yuan et al. (2025), few rollouts (*e.g.*, fewer than 16 rollouts) may lead to unstable results. To further validate the results of $\alpha$1 with a large number of rollouts, we conduct experiments with 32 rollouts on all benchmarks with DeepSeek-R1-Distill-Qwen-1.5B and report the P@1 with these 32 rollouts. Table 11 shows the results, demonstrating consistent conclusions with results reported in the main paper: i) Our $\alpha$1 yields consistently better performance and reasoning efficiency; ii) As shown in Table 11 with more experiments, the effectiveness of our approach can be even better than we report in Table 1. For example, on AIME24, the improvement increased from the +6.7% reported in Table 1 to +9.2%.

## C Artifacts Statements

### C.1 Model Artifacts

We utilize three models in our work: DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B, both released under the MIT License, which permits commercial use, modification, and redistribution. These models are distilled from Qwen-2.5 series (Apache 2.0 License). Additionally, we use Qwen QwQ-32B, which is released under the Apache License 2.0, allowing both research and commercial usage. We comply with all respective license terms in our use of these models.

### C.2 Data Artifacts

We employ publicly available datasets in our experiments. AIME24, Minerva-Math, LiveCodeBench, and OlympiadBench are released under the MIT License, which permits unrestricted use, modification, and redistribution. The AMC23 dataset does not have an explicitly specified license, so we treat it as having an unspecified license and exercise caution in its usage. We ensure full compliance with the respective license terms of all datasets used.

## D Future Works

While our $\alpha$1 has been demonstrated to be successful and effective in scaling LRMs at test time, there are some intriguing future works that we are considering:

- *More sophisticated slow thinking scheduling.* This work focuses on simple strategies like the slow-to-fast schedule, which shows strong performance. However, optimal scheduling remains an open question, as human reasoning patterns are complex and not yet fully understood (Kahneman, 2011). Promising directions include modulating reasoning progress during both training and inference, or learning a separate *progress modulation model* aligned with human preferences—akin to a progress reward model (Uesato et al., 2022; Lightman et al., 2024). In addition, $\alpha$ moment can be adaptively sampled from a subnetwork, and the reasoning scheduling strategy can be adaptively selected when facing different problems. By appropriately formulating the

Table 11: **Results with 32 rollouts**. The results are P@1 (#Tk) with 32 rollouts with DeepSeek-R1-Distill-Qwen-1.5B.

| | AIME24 | AMC23 | Minerva | MATH500 | LiveCode | Olympiad |
|---|---|---|---|---|---|---|
| Base | 21.1 (7407) | 60.2 (5482) | 30.5 (5030) | 77.8 (3911) | 18.7 (6946) | 37.9 (6089) |
| **$\alpha$1 (Ours)** | 30.3 (5669) | 72.4 (4861) | 32.2 (4581) | 81.5 (4121) | 24.5 (5004) | 44.6 (4922) |

problem of reasoning modulation as an RL-based optimization problem, we may obtain an adaptive $\alpha$1 that achieves better generalization capability.

- *Transitioning-token-agnostic modulation.* As shown in Table 6, the choice of transitioning token (e.g., "`wait`") affects performance due to model-specific training data. This limitation is shared by many test-time scaling methods relying on open-source LRMs like DeepSeek-R1 (DeepSeek-AI et al., 2025), in contrast to restricted-access models like OpenAI o1. While $\alpha$1 supports flexible token choices, removing the dependency on transitioning tokens altogether could further enhance generalization.

- *Multimodal reasoning with multimodal LLMs.* Multimodal LLMs are rapidly advancing and show growing potential in reasoning tasks (Alayrac et al., 2022; Liu et al., 2023; OpenAI, 2024; Team, 2024; Dong et al., 2024; Qi et al., 2024; Zou et al., 2025; Wang et al., 2024; Ning et al., 2025). Although they currently trail behind text-only LRMs, efforts to enhance their reasoning abilities are gaining momentum (Hao et al., 2024; Xiong et al., 2024; Wei et al., 2024; Wu and Xie, 2024; Shao et al., 2024a; Lee et al., 2024; Wei et al., 2025; Jiang et al., 2025; Yu et al., 2025). Multimodal reasoning remains challenging, as it requires both image understanding and generation (OpenAI, 2025). We aim to extend our $\alpha$1 framework to this domain, fostering synergistic multimodal comprehension and creation (Dong et al., 2024). Another promising direction is embodied reasoning—grounding multimodal understanding in real-world interactions with spatial intelligence (Fei-Fei, 2023; Qi et al., 2025; Yang et al., 2025b).

## E Qualitative Examples

We present qualitative examples from different models and benchmarks to illustrate both cases: instances where $\alpha$1 helps the model answer correctly, and instances where it still fails to produce the correct answer. Examples show that by appending "`wait`" frequently after "`\n\n`" can slow down the thinking process and may help the model achieve better performance. The examples can be found in the following pages.

**Successful Example of DeepSeek-R1-Distill-Qwen-1.5B on OlympiadBench: Base fails while $\alpha$1 passes**

**Question**: A chemist has three bottles, each containing a mixture of acid and water: - bottle A contains $40\text{g}$ of which $10$ is acid, - bottle B contains $50\text{g}$ of which $20$ is acid, and, - bottle C contains $50\text{g}$ of which $30$ is acid. She uses some of the mixture from each of the bottles to create a mixture with mass $60\text{g}$ of which $25$ is acid. Then she mixes the remaining contents of the bottles to create a new mixture. What percentage of the new mixture is acid?

**Response from Base**: <think>
Okay, so I've got this chemistry problem here, and I need to figure out the percentage of acid in the new mixture. Let me try to understand the problem step by step.
First, there are three bottles, each with a mixture of acid and water. The masses and acid concentrations are given as follows:
- Bottle A: 40 grams, 10% acid.
- Bottle B: 50 grams, 20% acid.
- Bottle C: 50 grams, 30% acid.
The chemist is going to use some mixture from each bottle to create a new mixture that's 60 grams with 25% acid. Then, after using some from each bottle, she mixes the remaining contents to create a new mixture. ... </think>

The chemist has three bottles containing acid and water with the following compositions:
- Bottle A: 40 grams of solution, 10% acid...
The percentage of the new mixture is $\boxed{18.75}$).

**Response from $\alpha$1**: <think>
Okay, so I've got this chemistry problem here, and I need to figure out the percentage of acid in the new mixture. Let me try to understand the problem step by step.
Wait, actually, the problem says that the chemist uses some mixture from each bottle to create a 60g mixture with 25% acid. Then, she mixes the remaining contents of the bottles to create a new mixture. Hmm, I need to clarify if she uses some from each bottle to make 60g, and then mixes the leftover from each bottle. Or is it that she uses some from each bottle to make 60g, and then uses the leftovers from each bottle to make another mixture? The wording is a bit unclear.
Wait, let me re-read the problem.
Wait, actually, the problem says: ... </think>

The percentage of acid in the new mixture is 17.5%.
**Step-by-Step Explanation:**
1. **Initial Acid Quantities:**
- Bottle A: 40g of 10% acid → 4g acid.
- Bottle B: 50g of 20% acid → 10g acid...
**Answer:** The new mixture has $\boxed{17.5}$% acid.

**Correct Answer**: 17.5

**Failure Example of DeepSeek-R1-Distill-Qwen-1.5B on AMC23: $\alpha$1 still fails**

**Question**: There is a unique sequence of integers $a_1, a_2, \cdots, a_{2023}$ such that $\tan 2023x = \frac{a_1 \tan x + a_3 \tan^3 x + a_5 \tan^5 x + \cdots + a_{2023} \tan^{2023} x}{1 + a_2 \tan^2 x + a_4 \tan^4 x + \cdots + a_{2022} \tan^{2022} x}$ whenever $\tan 2023x$ is defined. What is $a_{2023}$?

**Response from $\alpha$1**: <think>
Okay, so I need to find the value of $a_{2023}$ in the given expression for tan(2023x). The expression is a fraction where the numerator is a sum of odd-powered terms of (tan x) multiplied by coefficients $a_1, a_3, \ldots, a_{2023}$, and the denominator is a sum of even-powered terms of (tan x) multiplied by coefficients $a_2, a_4, \ldots, a_{2022}$. The problem states that this sequence of integers is unique, so I need to figure out what $a_{2023}$ is.
Wait, before jumping into that, I should recall the tangent multiple-angle formulas. I remember that tan(nx) can be expressed as a ratio of polynomials in tan x, where the numerator is an odd-degree polynomial and the denominator is an even-degree polynomial ...</think>

After carefully considering the expansion of tan(2023x) and testing various formulas ... which is the coefficient of $tan^{2023}x$ in the numerator, we look at the coefficient for ( k = 2023 ) in the numerator sum:
$[a_{2023} = (-1)^{2023-1}\binom{2023}{2023} = (-1)^{2022} \times 1 = 1]$
Therefore, $a_{2023} = 1$.\n\n[\n$\boxed{1}$\n]

**Correct Answer**: -1

**Successful Example of DeepSeek-R1-Distill-Qwen-7B on Minerva Math: Base fails while $\alpha$1 passes**

**Question**: A particular star has an absolute magnitude $M = -7$. If this star is observed in a galaxy that is at a distance of $3\mathrm{Mpc}$, what will its apparent magnitude be?

**Response from Base**: <think>
Okay, so I need to find the apparent magnitude of a star given its absolute magnitude and the distance to the galaxy it's in. Hmm, let's see. I remember that absolute magnitude is how bright a star appears at a standard distance, which I think is 10 parsecs. Apparent magnitude, on the other hand, is how bright it appears from Earth, which depends on how far away it actually is.
The formula that relates absolute magnitude (M), apparent magnitude (m) ...</think>

To find the apparent magnitude of a star given its absolute magnitude and the distance to the galaxy it is in, we use the distance modulus formula. The formula is:\n\n[ m - M = 5log(d) - 5 ]\n\nwhere (m) is the apparent magnitude...Rounding to one decimal place, the apparent magnitude is approximately 20.4.\n\n[\n$\boxed{20.4}$\n]

**Response from $\alpha$1**: <think>
Okay, so I need to find the apparent magnitude of a star given its absolute magnitude and the distance to the galaxy it's in. Hmm, I remember that absolute magnitude is how bright a star appears at a standard distance, which is 10 parsecs, right? And apparent magnitude is how bright it appears from Earth, which depends on its actual distance. The formula that relates absolute magnitude (M), apparent magnitude (m), and distance (d) in parsecs is the distance modulus formula. I think it's something like m = M + 5 + 5 $\log_{10}$(d). Wait, let me make sure. Is it plus or minus 5? I think it's plus 5 because as the distance increases, the apparent magnitude increases, meaning it becomes dimmer. So, yes, m = M + 5 $\log_{10}$(d) + 5.
Wait, hold on, no, that's not right...Then, subtract 5:\n\n[ 25.3855 - 5 = 20.3855 ]
So, the apparent magnitude is approximately 20.3855, which rounds to 20.39.\n\nTherefore, the apparent magnitude of the star is $\boxed{20.39}$.

**Correct Answer**: 20.39

**Failure Example of DeepSeek-R1-Distill-Qwen-7B on LiveCodeBench: $\alpha$1 still fails**

**Question**: You are given a 0-indexed string s and a dictionary of words dictionary. You have to break s into one or more non-overlapping substrings such that each substring is present in dictionary. There may be some extra characters in s which are not present in any of the substrings.
Return the minimum number of extra characters left over if you break up s optimally.

Example 1:
Input: s = "leetscode", dictionary = ["leet","code","leetcode"]. Output: 1
Explanation: We can break s in two substrings: "leet" from index 0 to 3 and "code" from index 5 to 8. There is only 1 unused character (at index 4), so we return 1.
Example 2:
Input: s = "sayhelloworld", dictionary = ["hello","world"]. Output: 3
Explanation: We can break s in two substrings: "hello" from index 3 to 7 and "world" from index 8 to 12. The characters at indices 0, 1, 2 are not used in any substring and thus are considered as extra characters. Hence, we return 3.

Constraints:1 <= s.length <= 50, 1 <= dictionary.length <= 50,
1 <= dictionary[i].length <= 50, dictionary[i] and s consists of only lowercase English letters, dictionary contains distinct words

**Response from $\alpha$1**: <think>
Okay, I need to solve this problem where I have to find the minimum number of extra characters left when breaking a string s into substrings that are all present in a given dictionary. Hmm, let's think about how to approach this.
Wait, the problem says that the string must be broken into one or more non-overlapping substrings, each of which is in the dictionary. ... </think>

To solve this problem, we need to break a given string ‘s‘ into substrings such that each substring is present in a given dictionary. ...

### Solution Code
“‘python\n \nclass Solution:\n def minExtraChar(self, s: str, dictionary: List[str]) -> int:\n n = len(s)\n dp = [float('inf')] * (n + 1)\n dp[0] = 0 # Base case: 0 characters, 0 extra\n \n for i in range(1, n + 1):\n for l in range(1, i + 1):\n start = i - l\n if start < 0:\n continue\n substring = s[start:i]\n if substring in dictionary:\n if dp[start] + (n - i) < dp[i]:\n dp[i] = dp[start] + (n - i)\n\n return dp[n]

### Explanation
- **Initialization**: The DP array is initialized to handle the base case where no characters are processed, resulting in zero extra characters. ...
- **Result Extraction**: The final result is found in ‘dp[n]‘, which gives the minimum number of extra characters after processing the entire string.
This approach ensures that we efficiently find the optimal way to split the string into valid substrings while minimizing the extra characters, leveraging dynamic programming to explore all possible valid splits.

**Successful Example of QwQ-32B on MATH500: Base fails while $\alpha$1 passes**

**Question**: There are 360 people in my school. 15 take calculus, physics, and chemistry, and 15 don't take any of them. 180 take calculus. Twice as many students take chemistry as take physics. 75 take both calculus and chemistry, and 75 take both physics and chemistry. Only 30 take both physics and calculus. How many students take physics?

**Response from Base**: <think>
Okay, let's try to tackle this problem step by step. Hmm, so there are 360 people in the school. Out of these, 15 don't take any of the three subjects: calculus, physics, or chemistry. That means 360 - 15 = 345 students take at least one of those subjects.
Alright, let me note down all the given information:
- Total students: 360
- Take all three (calculus, physics, chemistry): 15
- Don't take any: 15
- Take calculus: 180...</think>

Given the problem, we need to determine the number of students taking physics. Here is the step-by-step reasoning:\n\n
1. **Total students**: 360
2. **Students not taking any subjects**: 15, so students taking at least one subject: (360 - 15 = 345)
3. **Students taking calculus (C)**: 180
4. **Students taking all three subjects (C, P, Ch)**: 15...Thus, the number of students taking physics is ($\boxed{115}$).

**Response from $\alpha$1**: <think>
Okay, let's try to tackle this problem step by step. Hmm, so there are 360 people in the school. 15 take all three subjects: calculus, physics, and chemistry. Also, 15 don't take any of them. That means the number of students taking at least one of the subjects is 360 - 15 = 345.
Wait, the total number in the union of calculus, physics, and chemistry is 345.
Let me note down the given data first:
- Total students: 360
- Take all three (Calculus, Physics, Chemistry): ...</think>

So simplifying:
180 + P + Ch - 180 +15 = P + Ch +15 = 345
Hence, P + Ch = 330. Since Ch = 2P, substituting:
P + 2P = 3P = 330 $\rightarrow$ P = 110.
All steps check out, and the Venn diagram regions confirm consistency. Therefore, the number of students taking physics is $\boxed{110}$.
**Final Answer**\n$\boxed{110}$.

**Correct Answer**: 110

**Failure Example of QwQ-32B on AIME24: $\alpha$1 still fails**

**Question**: Torus $T$ is the surface produced by revolving a circle with radius $3$ around an axis in the plane of the circle that is a distance $6$ from the center of the circle (so like a donut). Let $S$ be a sphere with a radius $11$. When $T$ rests on the outside of $S$, it is externally tangent to $S$ along a circle with radius $r_i$, and when $T$ rests on the outside of $S$, it is externally tangent to $S$ along a circle with radius $r_o$. The difference $r_i - r_o$ can be written as $\frac{m}{n}$, where $m$ and $n$ are relatively prime positive integers. Find $m+n$.

$[asy]$ unitsize(0.3 inch); draw(ellipse((0,0), 3, 1.75));
draw((-1.2,0.1).(-0.8,-0.03)..(-0.4,-0.11)..(0,-0.15)..(0.4,-0.11)..
(0.8,-0.03)..(1.2,0.1)); draw((-1,0.04)..(-0.5,0.12)..(0,0.16)..(0.5,0.12)..
(1,0.04)); draw((0,2.4)-(0,-0.15)); draw((0,-0.15)-(0,-1.75), dashed);
draw((0,-1.75)-(0,-2.25)); draw(ellipse((2,0), 1, 0.9));
draw((2.03,-0.02)-(2.9,-0.4)); $[/asy]$

**Response from $\alpha$1**: <think>
Okay, so I need to solve this problem about a torus and a sphere. Let me try to understand the setup first. The torus T is formed by revolving a circle with radius 3 around an axis in the same plane, which is 6 units away from the center of the circle. That makes sense—it's like a donut shape where the tube radius is 3 and the distance from the center of the tube to the center of the torus is 6. Then there's a sphere S with radius 11. The problem says that when the torus rests on the outside of the sphere, it can be externally tangent to the sphere along a circle with radius $r_i$, and similarly along another circle with radius $r_o$. The question is asking for the difference $r_i$ - $r_o$, expressed as a fraction m/n where m and n are coprime, and then m + n.
Wait, actually, the problem mentions...</think>

Okay, let's start by setting up a coordinate system where the sphere is centered at the origin ((0, 0, 0) with radius 11. The torus (T) is generated by revolving a circle of radius 3 around an axis that is 6 units away from the center of the generating circle. Let's place the generating circle in the (x)-(y) plane, centered at ((6, 0, 0)). The axis of revolution is the (z)-axis, so the torus's axis is the (z)-axis.
When the torus rests on the outside of the sphere, they are externally tangent along a circle. To find the radii ($r_i$) and ($r_o$) of these circles of tangency, we need to consider the cross-sectional view in the (x)-(y) plane. Here, the sphere appears as a circle of radius 11 centered at the origin, and the generating circle of the torus is a circle of radius 3 centered at ((6, 0)).
For the torus to be externally tangent to the sphere, the distance between their centers must equal the sum of their radii. However, since the sphere is much larger, the torus must be positioned such that the distance from the sphere's center to the torus's generating circle's center plus the sphere's radius equals the distance from the sphere's center to the farthest point on the torus. ...

**Correct Answer**: 127