# Adaptive Spatial and Temporal Redundancy Optimization for Efficient Reasoning in Large Language Models

**Tianle Chen**[♡*], **Pengyu Cheng**[♡*], **Qiyuan Zhu**[♣*], **Jiacheng Wang**[♡], **Bei Liu**[♣], **Hao Gu**[♣], **Ruijie Shen**[♠], **Xiaofeng Hou**[♢], **Sirui Han**[♣†], **Jiacheng Liu**[♣†]

[♣] The Hong Kong University of Science and Technology, [♡] Xi'an Jiaotong University

[♠] Toulouse School of Economics, [♢] Shanghai Jiao Tong University

{tianlechen1030,pengyucheng0423,jiacheng}@xjtu.stu.edu.cn

qzhuat@connect.ust.hk, {beiliu,siruihan,jiachengliu}@ust.hk

marcusguhao@gmail.com, ruijie.shen@tse-fr.eu, xfhelen@sjtu.edu.cn

## Abstract

Large Language Models (LLMs) have achieved exceptional performance in complex reasoning via Chain-of-Thought (CoT), yet the associated computational costs remain prohibitive. CoT reasoning contains significant untapped efficiency potential across two dimensions: *temporal redundancy*, where reasoning steps may be unnecessary, and *spatial redundancy*, where computations can be performed at reduced precision. While current optimization techniques often necessitate resource-intensive fine-tuning or data curation, we introduce ASTRO (Adaptive Spatial and Temporal Redundancy Optimization), a training-free framework that simultaneously addresses both dimensions. ASTRO leverages Dewey's reflective thinking model to segment reasoning phases, applying a progressive precision reduction strategy coupled with an entropy-based confidence mechanism for adaptive termination. Empirical results across diverse reasoning benchmarks demonstrate that ASTRO achieves up to an 11.3× efficiency gain without compromising accuracy, highlighting the advantages of holistic multi-dimensional redundancy management over isolated optimization methods.

## 1 Introduction

Chain-of-Thought (CoT) reasoning has emerged as a dominant paradigm for enhancing the problem-solving capabilities of Large Language Models (LLMs), enabling the systematic decomposition of complex tasks into intermediate steps (Wei et al., 2022; Parashar et al., 2025). Despite its success, the autoregressive generation of extended reasoning chains imposes significant computational overhead. The requirement to process a high volume of intermediate tokens leads to substantial inference latency, excessive memory consumption, and high energy demands (Sui et al., 2025; Feng et al., 2025).

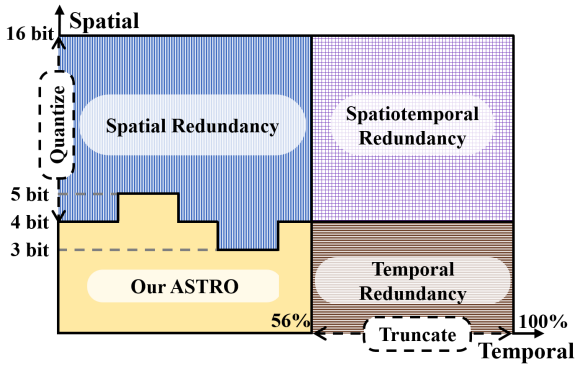



Figure 1: Reducing temporal and spatial redundancy is critical to optimize the efficiency of reasoning LLMs.

These inefficiencies present a formidable barrier to the practical deployment of reasoning-heavy models, particularly in latency-sensitive or resource-constrained environments (Liu et al., 2026; Yu et al., 2026, 2025).

Existing research to improve CoT efficiency generally falls into three categories, each with distinct limitations. *Training-based methods* optimize reasoning by fine-tuning models on specialized datasets to reduce reasoning steps (Liu et al., 2024; Xia et al., 2025); however, these approaches require massive computational resources and often lack cross-domain transferability. *Structural optimization* techniques utilize prompt engineering to streamline reasoning paths (Kang et al., 2024), yet they frequently sacrifice reasoning completeness for brevity. Finally, *computational optimization* applies general-purpose efficiency techniques, such as uniform quantization (Zhang et al., 2025) or confidence-based early exiting (Qiao et al., 2025). Crucially, these methods treat reasoning as generic text generation, failing to exploit the unique, multi-dimensional redundancy inherent in the step-by-step reasoning process.

The fundamental inefficiency in CoT reasoning stems from a failure to address its multi-dimensional redundancy patterns. As illustrated

[*] Equal contribution

[†] Corresponding authors



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24647–24662

in Figure 1, modern reasoning LLMs exhibit redundancy across two critical axes: (1) **temporal redundancy**, where subsequent reasoning steps become superfluous once the model achieves sufficient internal confidence (Sui et al., 2025; Li et al., 2024), and (2) **spatial redundancy**, where specific computations can be executed at reduced numerical precision without compromising the final output quality (Zhang et al., 2025). Effectively exploiting these redundancies is non-trivial because they are intrinsically coupled; for instance, aggressive spatial quantization can introduce numerical noise that destabilizes confidence estimation, leading to premature or delayed temporal termination. Furthermore, CoT reasoning is characterized by heterogeneous computational phases which range from high-precision problem formulation to low-precision verification, meaning that static, independent optimization strategies fail to capture the dynamic shifts in redundancy patterns. Consequently, there is a critical need for a coordinated, training-free approach that can accurately detect reasoning progress and adaptively synchronize precision and termination policies in real-time.

To address these challenges, we propose *Adaptive Spatial and Temporal Redundancy Optimization (ASTRO)*, a training-free framework that jointly optimizes temporal and spatial redundancy through coordinated multi-dimensional exploitation. ASTRO segments reasoning sequences into distinct cognitive phases based on *Dewey's reflective thinking model*, allowing for phase-specific optimization. The framework introduces a progressive precision allocation mechanism that adjusts quantization levels based on both the reasoning phase and temporal progress. Furthermore, ASTRO synchronizes termination decisions using entropy-based confidence estimation derived from thought transition patterns, preventing the cascading error effect typical of independent optimization. The main contributions of this work can be summarized as follows,

- To the best of our knowledge, ASTRO is the first training-free framework that jointly optimizes temporal and spatial redundancy through coordinated, phase-aware scheduling.
- We introduce dynamic precision allocation that adapts to both reasoning phase characteristics and temporal progress, extending beyond static uniform quantization to exploit reasoning-specific computational patterns.
- We establish unified scheduling where precision adaptation informs termination decisions, preventing cascading errors and enabling compound efficiency gains that exceed independent optimization approaches.
- Extensive evaluations on multiple reasoning tasks demonstrate that ASTRO achieves up to 11.3× efficiency gains while maintaining comparable accuracy.

## 2 Motivations and Observations

### 2.1 Multi-Dimensional Redundancy Analysis

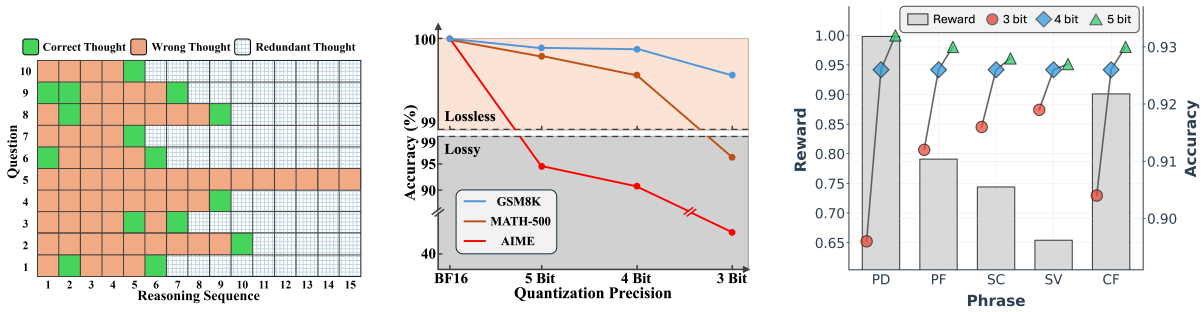
In order to quantitatively study the redundancy in large reasoning models, we conducted a study on the MATH-500 dataset (Hendrycks et al., 2021b). Our empirical analysis reveals redundancy patterns across two distinct dimensions that exhibit strong correlations, suggesting substantial potential for unified optimization strategies.

**Temporal redundancy** manifests when reasoning steps become unnecessary once sufficient understanding is achieved. For example, in the verification phases, reasoning often continues beyond the point where confidence stabilizes, with redundant verification steps providing minimal additional value while consuming substantial computational resources. Our analysis in Figure 2 (a) shows that a substantial fraction of reasoning steps could be eliminated without affecting final answer quality.

**Spatial redundancy** appears when computations can be performed at reduced precision without affecting reasoning quality. E.g., routine mathematical operations often do not require full precision, while complex analysis demands higher accuracy for numerical stability. Our statistical analysis in Figure 2 (b) reveals that this redundancy is substantial. For a majority of tasks, dramatically reducing precision from full BF16 down to 4-bits (w4), and in some cases even 3-bits (w3), incurs only a negligible drop in accuracy.

### 2.2 Phase-Based Computational Requirement

We hypothesize that CoT reasoning is not a monolithic computational process but rather consists of distinct phases with heterogeneous resource requirements. To test this, we conducted an experiment grounded in the principles of John Dewey's model of reflective thinking (Dewey, 1933).




(a) Temporal redundancy. (b) Spatial redundancy. (c) Phase-aware reward and quantization.

Figure 2: Redundancy and sensitivity analysis. (a) Temporal redundancy in reasoning sequences. (b) Spatial redundancy and accuracy loss across quantization levels. (c) Phase-specific reward and quantization sensitivity.

Specifically, we split the thoughts generated by LLMs into these phases (with the method introduced in the Methodology Section) and analyze them using a large reward model. Using Qwen2.5-Math-PRM-7B (Yang et al., 2024), we analyze the sensitivity of LLM reasoning phases to quantization. As shown in Figure 2(c), stages exhibit distinct reward magnitudes and precision requirements: while Problem Decomposition (PD) is highly sensitive to bit-width, Solution Consolidation (SC) remains robust even at 3-bit precision. This confirms that different reasoning phases possess heterogeneous computational demands.

These phase-dependent patterns reveal that optimal efficiency strategies must adapt dynamically rather than apply uniform policies. Static quantization fundamentally cannot capture these variations, leading to systematic resource misallocation where high-redundancy phases receive excessive computational resources while precision-critical phases may be under-provisioned.

## 3 Methodology

In this section, we introduce the ASTRO framework, depicted in Figure 3.

### 3.1 Problem Formulation

We formalize the multi-dimensional redundancy optimization problem for CoT reasoning. Consider model $M_\theta$ generating reasoning sequence $\mathbf{s} = (s_1, s_2, \ldots, s_T)$ given input $\mathbf{x}$, where the $t$-th thought $s_t$ is computed using precision $p_t \in \mathcal{P} = \{2, 3, 4, \cdots\}$ bits with computational cost $c(p_t)$.

**Temporal Redundancy** $R_T(t)$ quantifies the unnecessary computational cost from continuing reasoning beyond sufficient understanding:

$$R_T(t) = c(p_t)\,\mathbf{1}[t > t^*] \tag{1}$$

where $t^* = \min\{t' \mid \mathcal{Q}(s_{1:t'}) \geq \mathcal{Q}(s_{1:T}) - \epsilon\}$, and $\mathcal{Q}(\cdot)$ measures reasoning quality and $t^*$ is the optimal reasoning depth and $\epsilon$ represents acceptable quality tolerance.

**Spatial Redundancy** $R_S(t)$ quantifies computational redundancy from precision reduction without quality loss:

$$R_S(t) = c(p_t) - c(p_t^*) \tag{2}$$

where $p_t^* = \min\{p \in \mathcal{P} | \mathcal{Q}(s_t, p) \geq \mathcal{Q}(s_t, p_{\max}) - \delta\}$, and $\delta$ represents spatial quality tolerance.

### 3.2 Reasoning Phase Classification

Inspired by our observation of phase-dependent computational needs, we structure our adaptive framework around a cognitive model. John Dewey's reflective thinking model, as outlined in his foundational work How We Think (Dewey, 1933), provides a cognitive framework for thoughtful inquiry and problem-solving. It describes a sequential process involving five key phases: (1) the recognition of a felt difficulty or problem, (2) the location and intellectualization of the problem (defining it clearly), (3) the suggestion of possible solutions or hypotheses, (4) the development of these suggestions through reasoning and deduction, and (5) the testing of hypotheses through observation, experimentation, or verification, leading to acceptance or rejection.

Leveraging this cognitive science foundation, we adapt Dewey's model to the context of LLM reasoning, distilling it into five distinct phases that align with its core principles while emphasizing heterogeneous computational characteristics. These phases enable targeted optimizations that traditional static approaches cannot exploit effectively:

- Problem Definition (PD), which corresponds to Dewey's problem recognition (phase 1), focusing on initial context establishment;

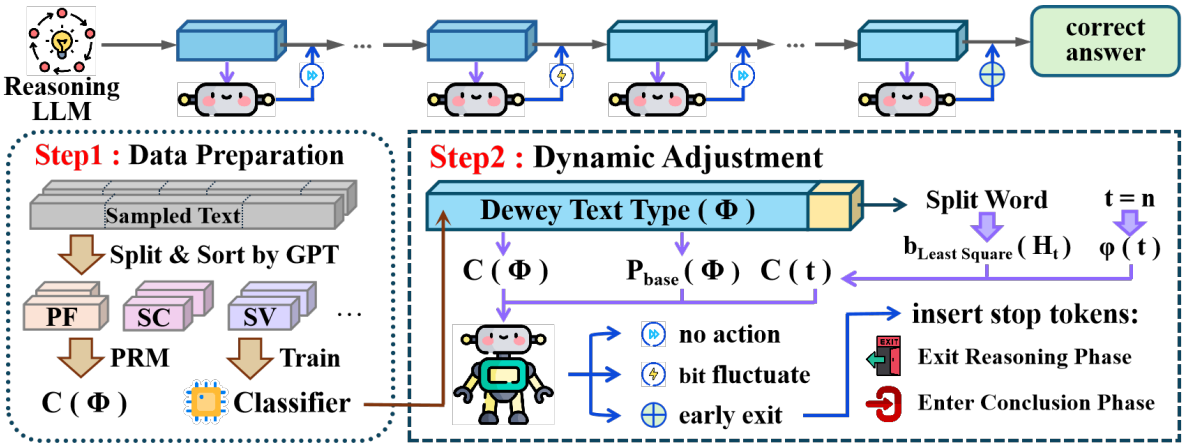



Figure 3: Framework of ASTRO, which effectively identifies different reasoning phases, dynamically adjusts their precision levels and timely enables early termination.

- Problem Formulation (PF), which extends intellectualization by formalizing problem structures, constraints, and representations (phase 2);
- Solution Computation (SC), analogous to hypothesis suggestion/initial development (phase 3), involves generative reasoning;
- Solution Verification (SV), reflecting further reasoning, deduction, and testing (phase 4), emphasizing validation and error checking;
- Conclusion Formation (CF), mirroring final verification and acceptance (phase 5), synthesizing results into a coherent outcome.

We segment reasoning sequences at thought transition points identified by linguistic indicators:

$$\mathcal{S}(s_t) = \mathbf{1}[\exists k \in \mathcal{K}_{split} : k \in s_t] \tag{3}$$

where $\mathcal{K}_{split} = \{\text{"wait"}, \text{"Wait"}, \text{"Alternatively"}\}$.

To ensure efficiency during the reasoning process, we model reasoning phase identification as a lightweight supervised text classification task at the segment level. Each segment is represented using sparse lexical features derived from a set of discriminative keywords, and classified with a Random Forest classifier.

Formally, given a segment $s$, the predicted reasoning phase is defined as

$$\Phi(s) = \arg\max_{\psi \in \Psi} p(\psi \mid s) \tag{4}$$

where $\Psi = \{\text{PD}, \text{PF}, \text{SC}, \text{SV}, \text{CF}\}$ denotes the predefined set of reasoning phases, and $p(\psi \mid s)$ is the posterior probability estimated by the classifier.

The keyword set is constructed based on class-conditional frequency statistics, retaining tokens that exhibit strong discriminability across reasoning phases. This design strikes a balance between computational efficiency and semantic sensitivity, enabling scalable reasoning-stage analysis without introducing substantial overhead. For further details, please refer to Appendix D.

### 3.3 Phase-Aware Progressive Quantization

Traditional quantization methods require separate model copies for each precision level, creating prohibitive memory overhead for dynamic switching. Any-Precision maintains a single model that can operate at multiple precisions through nested quantization (Park et al., 2024), enabling seamless precision adaptation essential for reasoning optimization. This capability is particularly valuable for reasoning tasks where precision requirements can change within a single reasoning sequence.

Our dynamic precision allocation adapts based on phase characteristics, reasoning progress, and confidence. The scheduler is guided by a principled linear model, which serves as a computationally efficient, first-order approximation of an ideal precision control function. This design enables fine-grained precision modulation during reasoning while avoiding the overhead of complex nonlinear schedulers. The precision level at step $t$ is:

$$p_t = \text{clip}(p_{t-1} + \Delta p_t,\ p_{\min},\ p_{\max}),$$
$$\Delta p_t = \begin{cases} +1, & C(t) < C(\Phi_t) - \epsilon, \\ -1, & C(t) > C(\Phi_t) + \epsilon, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where $p_{t-1}$ denotes the precision used at the previous reasoning step, with $p_{\min}$ and $p_{\max}$ define the

allowable precision range. $C(t)$ and $C(\Phi_t)$ denote the instantaneous and phase-conditioned reference confidence respectively, which are defined in equation (8). $\epsilon$ represents the tolerance margin.

This bounded and symmetric adjustment mechanism naturally permits both upward and downward precision transitions across reasoning steps, yielding non-monotonic yet stable precision trajectories.

### 3.4 Coordinated Termination

We design a coordinated termination strategy that jointly considers local confidence dynamics and progression-aware content signals, enabling robust early stopping without being misled by precision-induced uncertainty fluctuations.

**Split-token Confidence.** For reasoning steps involving split tokens, we quantify model confidence using a pointwise surprisal formulation:

$$H_t = -\log_2 P(v_t \mid \text{context}_t), \tag{6}$$

where $v_t$ denotes the selected split token at step $t$. Unlike entropy-based definitions that aggregate over a transition vocabulary, this formulation focuses on the realized decision and directly reflects the model's instantaneous confidence at each reasoning step. Unlike conventional confidence measures, a higher value of $H_t$ in our formulation indicates a lower probability of generating a split token, suggesting that the model is more inclined to continue along the current reasoning trajectory rather than switch to an alternative line of thought. This behavior provides supporting evidence for the correctness of the ongoing reasoning process.

Empirically, the trajectory of $H_t$ across reasoning steps $t$ reveals distinct patterns between successful and unsuccessful outcomes. For correct reasoning paths, $H_t$ typically increases smoothly, suggesting a stable and monotonic progression. Conversely, incorrect paths are characterized by high-frequency oscillations in $H_t$, even when the absolute magnitudes are comparable to those of correct paths. Detailed visualizations of these confidence patterns are provided in Appendix C.

Importantly, this separation becomes evident only in aggregate. At the level of individual reasoning traces, consecutive values of $H_t$ are often numerically close, such that global trend estimation methods (e.g., least-squares slope over $t$) become unstable and insufficiently sensitive to localized confidence transitions. To address this issue, we propose Local Confidence Dynamics as follows.

**Local Confidence Dynamics.** To capture short-range confidence variations, we adopt a first-order temporal difference:

$$\Delta H_t = H_t - H_{t-1}. \tag{7}$$

This local signal emphasizes step-wise confidence dynamics and highlights sudden decrease aggressively, corresponding to rapid consolidation of model belief. Compared to global slope-based measures, $\Delta H_t$ provides a more responsive and robust indicator of decisive reasoning transitions.

**Unified Confidence Score.** We combine local confidence dynamics with a progression-aware regularizer to obtain a unified confidence score:

$$C(t) = \omega \cdot \Delta H_t + (1 - \omega) \cdot \log_2 t \tag{8}$$

where $\log_2 t$ serves as a lightweight proxy for accumulated reasoning content and progression, and $\omega \in [0, 1]$ balances temporal confidence change against reasoning depth.

**Confidence-based Termination.** The coordinated termination decision integrates confidence dynamics with progression constraints through a phase-aware criterion:

$$\mathcal{T}(t) = (C(t) \geq \tau_{\phi_t}) \wedge (t \geq t_{min}) \tag{9}$$

where $t_{min}$ enforces a minimum number of reasoning steps to avoid premature stopping, $\tau_{\phi_t}$ denotes a phase-dependent confidence threshold conditioned on the predicted reasoning phase $\phi_t$ at step $t$.

To ensure natural and coherent completion, once the termination condition $\mathcal{T}(t)$ is satisfied, we inject a completion indicator (e.g., "Okay, I think I have finished thinking") into the context. This enables efficient early exit while preserving the semantic continuity of the reasoning trace.

## 4 Experiments

### 4.1 Experimental Setup

**Models and Datasets.** We conduct comprehensive evaluation using two model variants: *DeepSeek-R1-Distill-Qwen-7B* (Guo et al., 2025) and *Qwen-3-8B* (Yang et al., 2025), implementing our framework with the Any-Precision quantization algorithm (Park et al., 2024) to enable seamless dynamic precision switching without memory overhead. Our experiments cover three mathemat-

Table 1: Model performance on various reasoning datasets. Efficiency is measured as a composite of temporal and spatial efficiency.

| | DeepSeek-R1-Distill-Qwen-7B | | | | Qwen-3-8B | | | |
|---|---|---|---|---|---|---|---|---|
| Method | Accuracy | Avg. Bit | Avg. Tokens | Efficiency | Accuracy | Avg. Bit | Avg. Tokens | Efficiency |
| Dataset | **MATH-500 Dataset** | | | | | | | |
| Original | 0.930 | 16.00 | 4186.74 | 1.0× | 0.950 | 16.00 | 5469.73 | 1.0× |
| Uniform 4-bit | 0.926 | 4.00 | 4202.62 | 4.0× | 0.926 | 4.00 | 5342.95 | 4.1× |
| Uniform 3-bit | 0.896 | 3.00 | 4223.48 | 5.3× | 0.870 | 3.00 | 6140.33 | 4.8× |
| PMPD | 0.826 | 3.49 | 4161.34 | 4.6× | 0.924 | 3.56 | 5425.36 | 4.5× |
| NoThinking | 0.890 | 16.00 | 3309.78 | 1.3× | 0.926 | 16.00 | 4922.34 | 1.1× |
| S1 length-control | 0.908 | 16.00 | 2836.54 | 1.5× | 0.875 | 16.00 | 4727.32 | 1.2× |
| ASTRO (Ours) | 0.924 | 3.78 | 3215.47 | **5.5**× | 0.922 | 3.90 | 4563.04 | **4.9**× |
| Dataset | **AIME-120 Dataset** | | | | | | | |
| Original | 0.450 | 16.00 | 14909.66 | 1.0× | 0.675 | 16.00 | 17242.02 | 1.0× |
| Uniform 4-bit | 0.408 | 4.00 | 14140.91 | 4.2× | 0.667 | 4.00 | 15061.76 | 4.6× |
| Uniform 3-bit | 0.217 | 3.00 | 12481.43 | 6.4× | 0.292 | 3.00 | 18522.59 | 5.0× |
| PMPD | 0.342 | 3.46 | 12188.29 | 5.7× | 0.583 | 3.53 | 18134.53 | 4.3× |
| NoThinking | 0.283 | 16.00 | 9267.87 | 1.6× | 0.375 | 16.00 | 14862.65 | 1.2× |
| S1 length-control | 0.317 | 16.00 | 10131.44 | 1.5× | 0.492 | 16.00 | 13653.46 | 1.3× |
| ASTRO (Ours) | 0.383 | 3.70 | 10127.40 | **6.4**× | 0.542 | 3.77 | 11426.50 | **6.4**× |
| Dataset | **AMC-23 Dataset** | | | | | | | |
| Original | 0.900 | 16.00 | 6854.68 | 1.0× | 0.900 | 16.00 | 9756.05 | 1.0× |
| Uniform 4-bit | 0.900 | 4.00 | 6660.15 | 4.1× | 0.900 | 4.00 | 8560.50 | 4.6× |
| Uniform 3-bit | 0.825 | 3.00 | 7929.35 | 4.6× | 0.800 | 3.00 | 11231.74 | 4.6× |
| PMPD | 0.825 | 3.45 | 7342.91 | 4.3× | 0.825 | 3.45 | 8617.82 | 5.3× |
| NoThinking | 0.775 | 16.00 | 4856.43 | 1.4× | 0.850 | 16.00 | 8751.86 | 1.1× |
| S1 length-control | 0.850 | 16.00 | 5097.83 | 1.3× | 0.850 | 16.00 | 6423.78 | 1.5× |
| ASTRO (Ours) | 0.925 | 3.82 | 3871.98 | **7.4**× | 0.875 | 3.84 | 6692.15 | **6.1**× |
| Dataset | **GPQA-Diamond-MC Dataset** | | | | | | | |
| Original | 0.505 | 16.00 | 10268.76 | 1.0× | 0.586 | 16.0 | 10229.74 | 1.0× |
| Uniform 4-bit | 0.485 | 4.00 | 8145.08 | 5.0× | 0.566 | 4.00 | 10170.12 | 4.0× |
| Uniform 3-bit | 0.359 | 3.00 | 9387.17 | 5.8× | 0.429 | 3.00 | 10503.03 | 5.2× |
| PMPD | 0.429 | 3.51 | 8245.86 | 5.7× | 0.530 | 3.55 | 10435.67 | 4.4× |
| NoThinking | 0.364 | 16.00 | 3895.11 | 2.7× | 0.505 | 16.00 | 8802.09 | 1.2× |
| S1 length-control | 0.420 | 16.00 | 4923.51 | 2.1× | 0.484 | 16.00 | 8531.45 | 1.2× |
| ASTRO (Ours) | 0.455 | 3.84 | 3801.78 | **11.3**× | 0.561 | 3.78 | 6855.37 | **6.3**× |

ical reasoning benchmarks of varying complexity: MATH-500 (competition), AIME-120 (advanced), and AMC-23 (strategic), together with one general-purpose reasoning benchmark, GPQA-Diamond-MC (Rein et al., 2024). In addition, we conduct further evaluations on GSM8K and MMLU (Hendrycks et al., 2021a), with detailed results deferred to the appendix.

**Baselines and Metrics.** We establish comprehensive baselines representing different optimization paradigms to evaluate ASTRO's effectiveness across multiple dimensions.

*Static Quantization Baselines.* We implement uniform quantization strategies where all reasoning phases operate at fixed precision levels: 3-bit uniform quantization and 4-bit uniform quantization (Frantar et al., 2022). These baselines represent conventional static approaches that cannot adapt to dynamic reasoning requirements.

*Adaptive Quantization Baseline.* Following the Progressive Mixed-Precision Decoding (PMPD) framework (Chen et al., 2025a), we implement a naive scheduler that switches from high- to low-precision models. This approach provides temporal adaptation without reasoning awareness, serving as a direct comparison for our coordinated strategy.

*Early Termination Baseline.* We compare our approach with two training-free early stopping methods, S1 length control (Muennighoff et al., 2025) and NoThinking (Ma et al., 2025a).In the compari-

son, we align their generation budgets with ours to enable a fair evaluation of performance.

*Efficiency Metric.* We use a composite efficiency metric that jointly captures gains from reducing *temporal redundancy* and *spatial redundancy*. Specifically, temporal efficiency is reflected by the reduction in generated token count relative to the original decoding process, while spatial efficiency is approximated by the reduction in average bit-width normalized to the 16-bit setting. This design aligns with ASTRO's two-dimensional optimization objective. Since the metric reflects algorithmic efficiency rather than directly measured runtime, we separately report measured inference latency in Section 4.3.

### 4.2 Main Results

As presented in Table 1, ASTRO demonstrates consistent and substantial superiority over all baselines across two distinct models and a wide spectrum of reasoning tasks. To better understand this behavior, we organize the results by task characteristics and difficulty.

**Navigating the Trade-off in Complex Problems (MATH-500).** On the challenging MATH-500 dataset, ASTRO demonstrates a strong ability to balance reasoning quality and efficiency. It retains over *97% of the original accuracy* on both models (99.3% for DeepSeek-R1, 97.1% for Qwen-3) while delivering a 5.3×-6.2× efficiency gain. This success is directly attributable to our phase-aware scheduling. The framework allocates higher precision to critical initial phases, then progressively reduces it, preventing the kind of catastrophic errors seen in baselines. For instance, the PMPD scheduler, with its one-way switch from high to low precision, is too brittle for this complexity, causing a *severe 11.2% accuracy drop* on DeepSeek-R1. Under comparable token counts relative to ASTRO while preserving their original precision, both S1 length-control and the NoThinking method demonstrate unstable accuracy at an efficiency ratio of 1.3×–1.5×, significantly underperforming compared to our method. This demonstrates that for moderately difficult reasoning tasks, a simple temporal schedule is insufficient; a coordinated, context-aware strategy like ASTRO's is required for strong performance.

**Robustness Under Pressure in Contest-Level Scenarios (AIME-120, AMC-23).** The advantages of ASTRO become even more evident on contest-level benchmarks, where achieving substantial acceleration without collapsing accuracy is particularly challenging. On AIME-120, ASTRO achieves a 6.4× speedup while still retaining 85.1% and 80.3% of the original accuracy on DeepSeek-R1 and Qwen-3, respectively, outperforming baselines that incur much larger accuracy degradation at similar or lower efficiency. On AMC-23, ASTRO further achieves 6.1×–7.4× efficiency gains with competitive accuracy, placing it on the *Pareto frontier* of accuracy–efficiency trade-offs. These results demonstrate that ASTRO delivers near-optimal acceleration while preserving strong reasoning capability in the most demanding settings.

**Broad Applicability on Scientific Reasoning (GPQA-Diamond-MC).** Beyond mathematical reasoning, ASTRO also generalizes well to scientific question answering. On GPQA-Diamond-MC, it delivers the most striking efficiency gains in the table, reaching *11.3*× on DeepSeek-R1 and *6.3*× on Qwen-3, while preserving *90.1%* and *95.7%* of the original accuracy, respectively. Compared with PMPD, which operates at a similar average bit-width, ASTRO achieves both higher accuracy and higher overall efficiency on both models. Compared with S1 length-control and NoThinking, ASTRO also shows a much stronger overall trade-off, since those methods can only reduce temporal cost while remaining spatially inefficient due to full-precision decoding. This demonstrates that ASTRO's principles of identifying and exploiting temporal-spatial redundancy are task-agnostic and highly effective for general-purpose reasoning as well.

### 4.3 Ablation Study

We conducted systematic ablation studies to evaluate the impact of each ASTRO component by progressively reducing its parameter values to 0 while keeping other components fixed. Using the MATH-500 benchmark, we measured performance through two key metrics: (1) per-question average token count (indicative of computational efficiency) and (2) dataset-wide accuracy (reflecting overall task performance). As detailed in Table 2, The study reveals two key results:

*Ablation of unified confidence weight:* By increasing $\omega$ from 0 to 1, we observe a clear shift in the balance between reasoning length and accuracy. When $\omega$ is small, the confidence score is dominated by progression, resulting in limited reasoning depth

Table 2: Ablation Study Results.

| $\omega$ | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 1 |
|---|---|---|---|---|---|---|
| Accuracy | 0.896 | 0.914 | 0.918 | 0.922 | 0.916 | 0.908 |
| Avg. Tokens | 3161.18 | 4172.95 | 3654.61 | 3405.83 | 3218.37 | 3101.78 |
| $\mathbf{t}_{\text{min}}$ | 0 | 1 | 2 | 3 | 4 | 5 |
| Accuracy | 0.908 | 0.914 | 0.918 | 0.924 | 0.924 | 0.926 |
| Avg. Tokens | 2786.58 | 2854.02 | 2976.72 | 3215.47 | 3533.46 | 3792.75 |

Table 3: Per-task latency (seconds) and token counts for DeepSeek-R1-Distill-Qwen-7B.

| **Dataset** | **16-bit** | | **4-bit** | | **3-bit** | | **ASTRO** | |
|---|---|---|---|---|---|---|---|---|
| | Lat. | Tok. | Lat. | Tok. | Lat. | Tok. | Lat. | Tok. |
| MATH-500 | 40.1 | 4187 | 25.0 | 4460 | 18.1 | 4223 | **17.1** | **3215** |
| AIME-120 | 151.3 | 14910 | 94.2 | 14141 | 61.4 | 12481 | **60.0** | **10127** |
| AMC-23 | 69.2 | 6855 | 43.1 | 6660 | 36.9 | 7929 | **22.9** | **3872** |
| GPQA | 105.8 | 10269 | 52.9 | 8145 | 44.7 | 9387 | **22.5** | **3802** |

and lower accuracy. Increasing $\omega$ improves accuracy and reduces token usage, while excessively large $\omega$ makes the score overly sensitive to entropy changes, leading to premature truncation.

*Ablation of reasoning window:* As $t_{\text{min}}$ is progressively reduced to 0, the robustness of our method declines. Intuitively, early exits may occur at suboptimal points due to inflated confidence values. Correspondingly, we observe a consistent decrease in both token count and accuracy.

We evaluate inference latency following the methodology of PMPD (Chen et al., 2025a), reporting *per-task latency* (average seconds per question) to enable direct comparison across methods. Unlike end-to-end dataset latency measurements common in some works, per-task latency isolates hardware efficiency from dataset size variations.

The results are shown in Table 3. While static quantization at 4-bit and 3-bit precision is expected to reduce latency compared to the 16-bit original, our ASTRO framework demonstrates overall greater gains. Specifically, ASTRO achieves speedups ranging from 2.3× to 4.7× across various datasets. Moreover, it attains higher accuracy (as shown in Table 1) than the Uniform 3-bit model while simultaneously achieving lower latency, indicating that ASTRO realizes an optimal trade-off between performance and efficiency.

## 5 Related Work

**CoT Reasoning Efficiency.** Recent efforts to improve CoT reasoning efficiency focus on token reduction by shortening CoT paths, building smaller models, or accelerating decoding (Feng et al., 2025; Hashemi et al., 2025; Chen et al., 2024; Lee et al., 2025; Gu et al., 2026; Wang et al., 2026; Nie et al., 2026). Methods to shorten CoT chains include training-dependent approaches like reinforcement learning (RL) with length penalties (Ma et al., 2025b; Li et al., 2025; Aggarwal and Welleck, 2025; Xia et al., 2025; Hou et al., 2025) (e.g., O1-Pruner (Luo et al., 2025), DAST (Shen et al., 2025a)) and supervised fine-tuning (SFT) on variable-length data (Xia et al., 2025; Ma et al., 2025b). Training-free alternatives use prompting to enforce brevity or route queries to specialized models (Renze and Guven, 2024; Ong et al., 2024). Other strategies build smaller, more capable models via knowledge distillation (Feng et al., 2024; Chen et al., 2025b; Shen et al., 2025b) or accelerate decoding with techniques like problem decomposition (Teng et al., 2025) and speculative decoding (Pan et al., 2025). These approaches often require training or treating redundancy dimensions independently, overlooking phase-specific patterns. Unlike them, our ASTRO framework provides training-free, phase-aware joint optimization of temporal and spatial redundancy.

**LLM Efficiency.** Model compression is pivotal for alleviating the high resource demands of LLMs, with key strategies including knowledge distillation (Gou et al., 2021), pruning (Frantar and Alistarh, 2023), and quantization (Lin et al., 2024; Frantar et al., 2022; Gu et al., 2025; Xu et al., 2026). Among these, post-training quantization (PTQ) provides a practical option by compressing models after training while retaining most performance with low overhead. Recent PTQ innovations include SmoothQuant (Xiao et al., 2023), which handles activation outliers for smoother low-bit conversion; GPTQ (Frantar et al., 2022), which fine-tunes quantization layer by layer; and AWQ (Lin et al., 2024), which prioritizes salient weights to maintain generation quality. Versatile systems like Any-Precision LLM (Park et al., 2024) allow runtime selection of bit-widths from a single model without added storage costs, and Progressive Mixed-Precision Decoding (Chen et al., 2025a) varies precision adaptively throughout the decoding sequence. Despite their effectiveness in curbing spatial overhead, these methods typically enforce broad, non-specialized policies that disregard the varying demands of CoT phases.

## 6 Conclusion

We presented ASTRO, a training-free framework for optimizing Chain-of-Thought reasoning through coordinated temporal and spatial redundancy exploitation. Our approach achieves up to 11.3× speedup while maintaining accuracy without requiring training. The results establish coordinated multi-dimensional optimization as substantially superior to conventional strategies, opening new directions for practical reasoning optimization.

## Limitations

Despite the efficiency gains of our approach, this work has several limitations. First, while our method reduces overall inference costs, the adaptive mechanism introduces a minor computational overhead during the decision-making process. Second, our experiments were conducted on English tasks, and the generalizability of these redundancy patterns to other languages with different information densities has yet to be fully established.

## Acknowledgments

This work is funded in part by the HKUST Start-up Fund (R9911), Theme-based Research Scheme grant (T45-205/21-N), the InnoHK funding for Hong Kong Generative AI Research and Development Center, Hong Kong SAR, the National Natural Science Foundation of China (No.62441225), and the research funding under HKUST-DXM AI for Finance Joint Laboratory (DXM25EG01).

## References

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

Hao (Mark) Chen, Fuwen Tan, Alexandros Kouris, Royson Lee, Hongxiang Fan, and Stylianos I. Venieris. 2025a. Progressive Mixed-Precision Decoding for Efficient LLM Inference. In *International Conference on Learning Representations (ICLR)*.

Xinghao Chen, Zhijing Sun, Wenjin Guo, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, and 1 others. 2025b. Unveiling the key factors for distilling chain-of-thought reasoning. *arXiv preprint arXiv:2502.18001*.

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

John Dewey. 1933. *How we think*. Houghton Mifflin.

Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903*.

Tao Feng, Yicheng Li, Li Chenglin, Hao Chen, Fei Yu, and Yin Zhang. 2024. Teaching small language models reasoning through counterfactual distillation.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Hao Gu, Lujun Li, Zheyu Wang, Bei Liu, Qiyuan Zhu, Sirui Han, and Yike Guo. 2025. Btc-llm: Efficient sub-1-bit llm quantization via learnable transformation and binary codebook. *arXiv preprint arXiv:2506.12040*.

Hao Gu, Hao Wang, Jiacheng Liu, Lujun Li, Qiyuan Zhu, Bei Liu, Binxing Xu, Lei Wang, Xintong Yang, Sida Lin, Sirui Han, and Yike Guo. 2026. Qarl: Rollout-aligned quantization-aware rl for fast and stable training under training–inference mismatch. *Preprint*, arXiv:2604.07853.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Masoud Hashemi, Oluwanifemi Bamgbose, Sathwik Tejaswi Madhusudhan, Jishnu Sethumadhavan Nair, Aman Tiwari, and Vikas Yadav. 2025. Dnr bench: Benchmarking over-reasoning in reasoning llms. *arXiv preprint arXiv:2503.15793*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2024. C3ot: Generating shorter chain-of-thought without compromising effectiveness. *arXiv preprint arXiv:2412.11664*.

Ayeong Lee, Ethan Che, and Tianyi Peng. 2025. How well do llms compress their own chain-of-thought? a token complexity approach. *arXiv preprint arXiv:2503.01141*.

Chen Li, Nazhou Liu, and Kai Yang. 2025. Adaptive group policy optimization: Towards stable training and token-efficient reasoning. *arXiv preprint arXiv:2503.15952*.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. *arXiv preprint arXiv:2401.10480*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100.

Jiacheng Liu, Peng Tang, Wenfeng Wang, Yuhang Ren, Xiaofeng Hou, Pheng Ann Heng, Minyi Guo, and Chao Li. 2026. A survey on inference optimization techniques for mixture of experts models. *ACM Computing Surveys*, 58(10):1–37.

Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2024. Can language models learn to skip steps? *arXiv preprint arXiv:2411.01855*.

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

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

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

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

Shuaiyi Nie, Siyu Ding, Wenyuan Zhang, Linhao Yu, Tianmeng Yang, Yao Chen, Tingwen Liu, Weichong Yin, Yu Sun, and Hua Wu. 2026. Attnpo: Attention-guided process supervision for efficient reasoning. *arXiv preprint arXiv:2602.09953*.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. 2024. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*.

Rui Pan, Yinwei Dai, Zhihao Zhang, Gabriele Oliaro, Zhihao Jia, and Ravi Netravali. 2025. Specreason: Fast and accurate inference-time compute via speculative reasoning. *arXiv preprint arXiv:2504.07891*.

Shubham Parashar, Blake Olson, Sambhav Khurana, Eric Li, Hongyi Ling, James Caverlee, and Shuiwang Ji. 2025. Inference-time computations for llm reasoning and planning: A benchmark and insights. *arXiv preprint arXiv:2502.12521*.

Yeonhong Park, Jake Hyun, SangLyul Cho, Bonggeun Sim, and Jae W Lee. 2024. Any-precision llm: Low-cost deployment of multiple, different-sized llms. *arXiv preprint arXiv:2402.10517*.

Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Fandong Meng, Jie Zhou, Ju Ren, and Yaoxue Zhang. 2025. Concise: Confidence-guided compression in step-by-step efficient reasoning. *arXiv preprint arXiv:2505.04881*.

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

Matthew Renze and Erhan Guven. 2024. The benefits of a concise chain of thought on problem-solving in large language models. In *FLLM*.

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

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025b. Codi: Compressing chain-of-thought into continuous space via self-distillation. *arXiv preprint arXiv:2502.21074*.

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

Fengwei Teng, Zhaoyang Yu, Quan Shi, Jiayi Zhang, Chenglin Wu, and Yuyu Luo. 2025. Atom of thoughts for markov llm test-time scaling. *arXiv preprint arXiv:2502.12018*.

Hao Wang, Hao Gu, Hongming Piao, Kaixiong Gong, Yuxiao Ye, Xiangyu Yue, Sirui Han, Yike Guo, and Dapeng Wu. 2026. Learning while staying curious: Entropy-preserving supervised fine-tuning via adaptive self-distillation for large reasoning models. *arXiv preprint arXiv:2602.02244*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models.

Heming Xia, Yongqi Li, Chak Tou Leong, Wenjie Wang, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR.

Binxing Xu, Hao Gu, Lujun Li, Hao Wang, Bei Liu, Jiacheng Liu, Qiyuan Zhu, Xintong Yang, Chao Li, Sirui Han, and Yike Guo. 2026. Bit-by-bit: Progressive qat strategy with outlier channel splitting for stable low-bit llms. *Preprint*, arXiv:2604.07888.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Han Yu, Nuo Chen, Bingqing Shen, Tieying Li, and Hongming Cai. 2025. A spatiotemporal-aware decentralized service discovery framework for drone swarms. *IEEE Internet of Things Journal*.

Han Yu, Qidan Qian, Hongming Cai, Bingqing Shen, and Lihong Jiang. 2026. Dsr: A dnn service recommendation system based on pragmatic information model for industrial defect detection. *IEEE Transactions on Services Computing*.

Nan Zhang, Yusen Zhang, Prasenjit Mitra, and Rui Zhang. 2025. When reasoning meets compression: Benchmarking compressed large reasoning models on complex reasoning tasks. *arXiv preprint arXiv:2504.02010*.

## A The Use of LLMs

During the writing process, we utilized LLMs, specifically GPT-5, to refine the manuscript's language for clarity and fluency. The authors retained full responsibility for all content, with the LLMs serving exclusively as a tool for language enhancement.

## B ASTRO Algorithm

Algorithm 1 presents our ASTRO framework. The algorithm operates by monitoring reasoning generation for thought transition points, which serve as coordination opportunities. At each transition, it performs phase classification, computes unified confidence metrics, and makes coordinated decisions about precision adaptation and potential termination.

## C Analysis of Confidence Patterns

We evaluate the evolution of $H_t$ over the reasoning process on four benchmarks: MATH500, GSM8K, AMC-23, and MMLU. The results are shown in Figure 1. Across all datasets, we observe a consistent linear increasing trend of $H_t$ with respect to the reasoning step, regardless of whether the final answer is correct or incorrect. This indicates that $H_t$ naturally grows as reasoning progresses.

More importantly, for correctly answered instances, the growth rate of $H_t$ is consistently steeper than that observed in incorrect cases. Since $H_t$ serves as a proxy for model confidence at each reasoning step, a larger slope indicates a faster accumulation of confidence as reasoning progresses. This suggests that correct reasoning paths tend to reinforce confidence more rapidly and consistently, whereas incorrect paths exhibit slower confidence growth, reflecting weaker or less reliable reasoning trajectories.

Motivated by these observations, we incorporate the trend of $H_t$ as a component of our method to assess the quality of generated answers, leveraging its growth behavior as an informative signal of reasoning correctness.

## D Details of Reasoning Phrase Classification

To understand the internal mechanics of ASTRO, we analyze two key components: the accuracy of our phase classifier and the resulting dynamic precision allocation strategy. The effectiveness of our framework hinges on correctly identifying the current reasoning phase to apply the appropriate optimization policy. As shown in Figure 2, our lightweight, keyword-based phase classifier (Eq. (5)) achieves over 95% average accuracy. The classifier is trained on 2,360 annotated text segments using a Random Forest model. Candidate tokens are first filtered by minimum occurrence frequency, and only those whose class-conditional proportion exceeds that of other classes by a factor of 2.5 are retained as discriminative keywords(details of this classifier can be found in Table 1). While this keyword-based classifier demonstrates high accuracy and efficiency for the tasks evaluated, we acknowledge that its robustness may vary on out-of-domain problems. Future work could explore replacing this with a small, lightweight learned classifier to enhance generalizability without significantly increasing computational overhead.

Building on this accurate phase detection, ASTRO demonstrates a sophisticated, task-aware approach to resource management, as illustrated in Figure 3. The framework learns to dynamically allocate precision based on the complexity and nature of the dataset. For the highly complex AIME dataset, the highest precision is allocated to Solution Computation (SC) (3.64 bits), emphasizing the need to calculation the right answer. In contrast, for the arithmetically-focused MATH dataset, the highest precision is shifted to Problem formulation (PF) (3.62 bits), reflecting the critical importance of rigorously problem analysis. This adaptive behavior confirms that ASTRO does not use a one-size-fits-all policy; instead, it intelligently distributes computational resources to the reasoning phases where they are most impactful, tailoring its strategy to the unique demands of each task.

## E Results on Additional Datasets

To further confirm that ASTRO's benefits extend across different reasoning settings, we evaluated it on the foundational mathematical reasoning dataset GSM8K multi-domain MMLU benchmark. The results further affirm its broad applicability. On GSM8K, ASTRO delivers substantial efficiency gains of *4.5*×*-5.5*× with almost no accuracy degradation relative to the full-precision baseline (e.g., 0.925 vs. 0.923 on DeepSeek-R1; 0.949 vs. 0.955 on Qwen-3). In contrast, S1 length-control and NoThinking remain spatially inefficient due to full-precision decoding, while Uniform 3-bit quantiza-

**Algorithm 1** ASTRO: Adaptive Spatial and Temporal Redundancy Optimization

**Require:** Reasoning LLM $M$, phase-conditioned confidence $C(\Phi_t)$, terminal threshold $\tau$ and $t_{min}$
**Ensure:** Generate sequence $\mathbf{s}$, precision $\mathbf{p}$

**while not** terminated **do**
  $s_{t+1} \leftarrow M(s_{1:t}, p_t)$
  **if** $\mathcal{S}(s_{t+1})$ **then**
    $\phi_{t+1} \leftarrow \Phi(s_{t+1})$
    $H_t \leftarrow$ ComputeEntropy$(s_{t+1})$
    $\Delta H_t \leftarrow H_t - H_{t-1}$
    $C_t \leftarrow$ UnifiedConfidence$(\phi_{t+1}, H_t, S_t)$
    $p_{target} \leftarrow$ ComputeTargetPrecision$(s_{t+1}, p_t)$
    **if** $C_t \geq \tau_{\phi_t}$ **and** $t \geq t_{min}$ **then**
      InjectCompletion$(\phi_{t+1})$
      **return** $\mathbf{s}_{1:t+1}$, $\mathbf{p}_{1:t+1}$
    **else**
      $p_{t+1} \leftarrow p_{target}$
    **end if**
  **else**
    $p_{t+1} \leftarrow p_t$
  **end if**
  $t \leftarrow t + 1$
**end while**
**return** $\mathbf{s}_{1:t}$, $\mathbf{p}_{1:t}$

tion causes a more noticeable accuracy drop, especially on Qwen-3-8B. This shows that ASTRO achieves a strong balance between temporal and spatial efficiency even on relatively straightforward reasoning tasks.

On the multi-domain MMLU benchmark, ASTRO again achieves a high efficiency gain of *4.4×-7.0×* while closely matching the original model's accuracy (e.g., 0.631 vs. 0.634 on DeepSeek-R1; 0.823 vs. 0.826 on Qwen-3). Compared with PMPD, S1 length-control, and NoThinking, our method delivers a much stronger overall trade-off between accuracy and efficiency. These results further demonstrate that ASTRO's principles of identifying and exploiting temporal-spatial redundancy generalize effectively across diverse reasoning workloads.

## F Results on Large-Size model

We evaluated our ASTRO method on the DeepSeek-R1-Distill-Qwen-32B model and compared it with the best-performing quantization baseline, as shown in Table 3. Our results show that ASTRO achieves comparable accuracy to the 4-bit quantization baseline while delivering a 4.7× – 6.5× speedup, effectively improving inference efficiency.

## G Completion Indicators

To implement natural termination, we employ confidence-based completion phrase injection. The contextually appropriate completion indicators employed in our experiments are as follows,

- **Prompt 1:** `Final Answer` The usage of this prompt stems from our empirical observation of model outputs. We consistently observed that the model generates the token sequence `Final Answer` immediately preceding its final output. We therefore hypothesize that this prompt effectively triggers early exit behavior. Experimental results demonstrate that while this prompt achieves excellent truncation performance, it inadvertently suppresses the generation of the solution reasoning component, thereby exerting non-negligible negative impacts on final accuracy.
- **Prompt 2:** **`Okay, I think I have finished thinking.`** This formulation draws inspiration from (Ma et al., 2025a), where the original work employed it at the beginning of model outputs to skip chain-of-thought reasoning. We posit that inserting

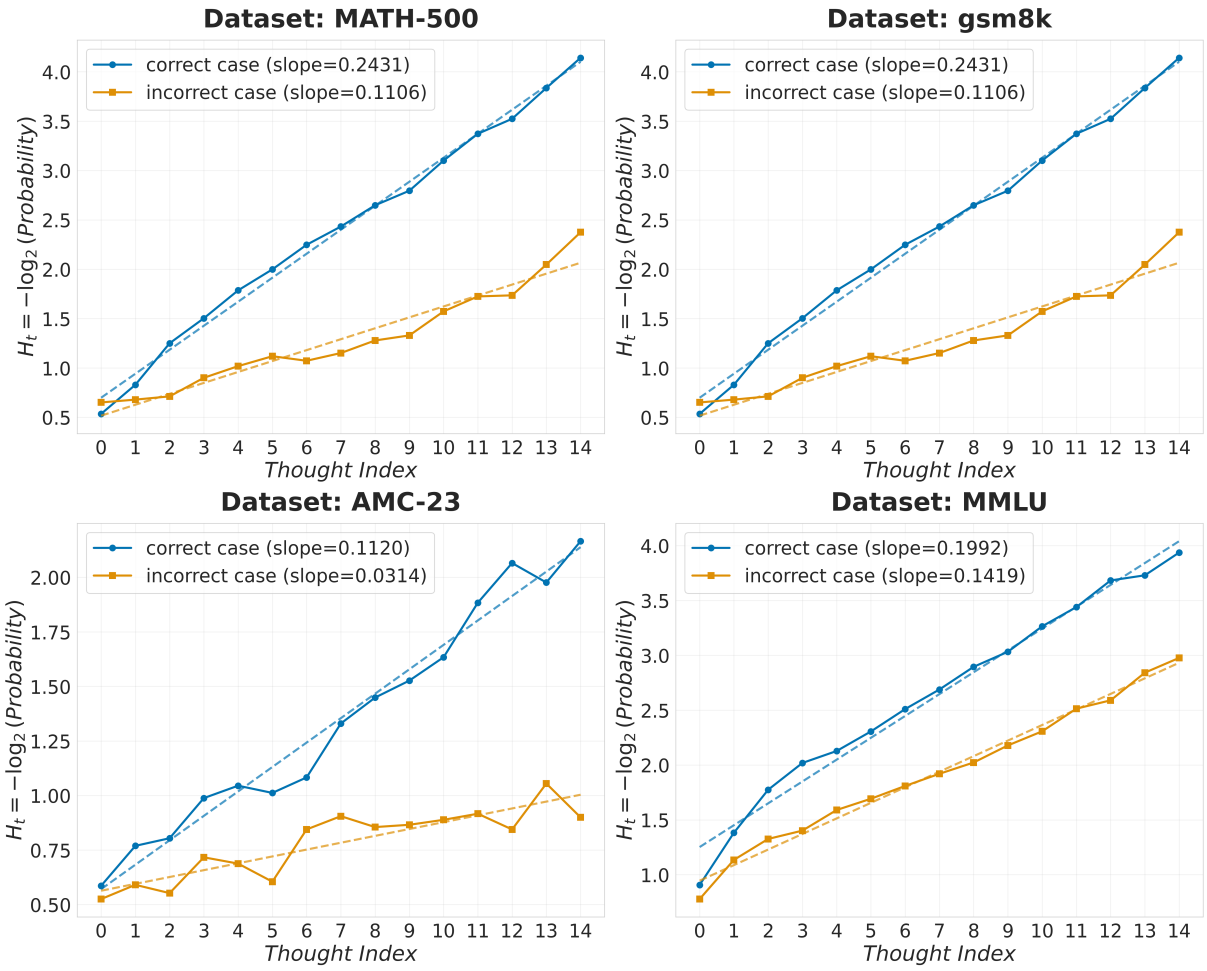



Figure 1: Split-token Confidence with Least-Squares Fit.

this prompt within reasoning chains can effectively induce early exit. Our experiments reveal that this prompt maintains an optimal balance between truncation efficiency and solution reasoning length, consequently enabling the model to simultaneously optimize both token generation quantity and prediction accuracy.

## H Case Study

To provide a granular view of our framework's mechanics, we present a case study on a challenging problem from the MATH-500 dataset. This analysis contrasts the lengthy, resource-intensive reasoning process of the original DeepSeek-R1-Distill-Qwen-7B model with the highly efficient, adaptive process guided by our ASTRO framework. The comparison, illustrated in Figure 4, reveals how ASTRO dynamically prunes both spatial and temporal redundancy without compromising the final answer's accuracy.

It can be observed that after generating 584 tokens, the original model continues to produce an additional 946 tokens. In contrast, when applying our ASTRO (Temporal-Spatial Adaptive Reasoning) method, the bit allocation process (indicated by blue arrows) can be observed, and the corresponding text output is nearly identical to that of the original model. Upon detecting that the reasoning quality meets the coordinated early-termination criterion, ASTRO inserts the phrase "Okay, I think I have finished thinking." to halt further generation by the original model. Subsequently, the quantized model with ASTRO generates only 180 tokens before concluding the reasoning process.

Although both approaches ultimately produce correct answers, our ASTRO method achieves dual improvements in temporal efficiency (reducing inference time) and spatial efficiency (optimizing computational resource usage).

Table 1: All the keywords for the reasoning phase classification.

| Phase | Keywords |
|---|---|
| **PF** | about, alright, axis, bit, break, confused, denote, down, figure, find, first, front, gecko, given, has, have, here, how, its, know, least, length, looking, many, means, need, okay, order, other, out, points, problem, recall, says, should, solve, some, somewhere, starting, step, three, triangles, try, units, vertices, what, where |
| **SC** | ceiling, compute, cos, express, formula, multiply, now, product, remember, simplify, sin, tan, together, use, wall, write |
| **SV** | answer, boxed, check, correct, final, hold, maximum, no, only |

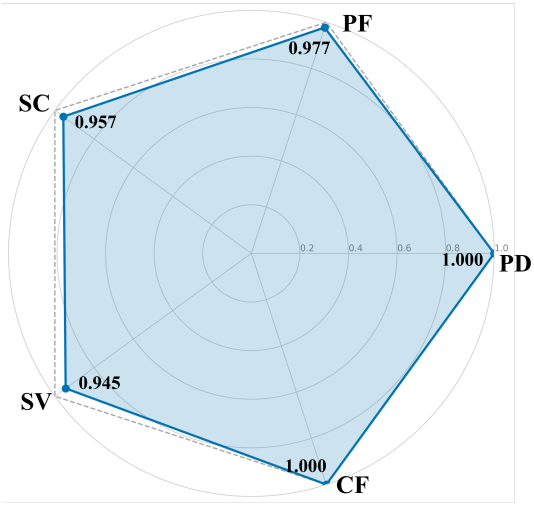



Figure 2: Phase classification accuracy.

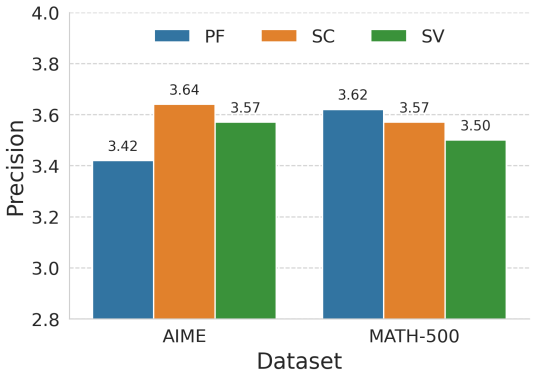



Figure 3: Task-precision allocation.

Table 2: Model performance on additional datasets.

| | DeepSeek-R1-Distill-Qwen-7B | | | | Qwen-3-8B | | | |
|---|---|---|---|---|---|---|---|---|
| Method | Accuracy | Avg. Bit | Avg. Tokens | Efficiency | Accuracy | Avg. Bit | Avg. Tokens | Efficiency |
| Dataset | **GSM8K Dataset** | | | | | | | |
| Original | 0.923 | 16.00 | 1847.16 | 1.0× | 0.955 | 16.00 | 2335.76 | 1.0× |
| Uniform 4-bit | 0.922 | 4.00 | 1822.44 | 4.1× | 0.952 | 4.00 | 2336.23 | 4.0× |
| Uniform 3-bit | 0.919 | 3.00 | 1857.97 | 5.3× | 0.924 | 3.00 | 2459.24 | **5.1**× |
| PMPD | 0.920 | 3.52 | 1835.73 | 4.6× | 0.938 | 3.53 | 2312.44 | 4.6× |
| NoThinking | 0.903 | 16.00 | 1984.76 | 0.9× | 0.942 | 16.00 | 2016.14 | 1.2× |
| S1 length-control | 0.920 | 16.00 | 1526.43 | 1.2× | 0.932 | 16.00 | 2162.08 | 1.1× |
| ASTRO (Ours) | 0.925 | 3.78 | 1430.34 | **5.5**× | 0.949 | 3.85 | 2165.03 | 4.5× |
| Dataset | **MMLU Dataset** | | | | | | | |
| Original | 0.634 | 16.00 | 1958.83 | 1.0× | 0.826 | 16.00 | 2252.57 | 1.0× |
| Uniform 4-bit | 0.633 | 4.00 | 1821.87 | 4.3× | 0.825 | 4.00 | 2089.12 | 4.3× |
| Uniform 3-bit | 0.592 | 3.00 | 2029.23 | 5.1× | 0.791 | 3.00 | 2124.62 | **5.7**× |
| PMPD | 0.603 | 3.54 | 1523.49 | 5.8× | 0.805 | 3.58 | 2203.87 | 4.6× |
| NoThinking | 0.571 | 16.00 | 1162.55 | 1.7× | 0.786 | 16.00 | 1857.23 | 1.2× |
| S1 length-control | 0.612 | 16.00 | 1547.64 | 1.3× | 0.812 | 16.00 | 2135.45 | 1.1× |
| ASTRO (Ours) | 0.631 | 3.85 | 1319.30 | **7.0**× | 0.823 | 3.95 | 2072.02 | 4.4× |

Table 3: Model performance on general-purpose datasets.

| Method | DeepSeek-R1-Distill-Qwen-32B | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Accuracy | Avg. Bit | Avg. Tokens | Efficiency | Accuracy | Avg. Bit | Avg. Tokens | Efficiency |
| | **MATH-500 Dataset** | | | | **AMC Dataset** | | | |
| Original | 0.940 | 16.00 | 4526.17 | 1.0× | 0.950 | 16.00 | 6824.46 | 1.0× |
| Uniform 4-bit | 0.928 | 4.00 | 4416.69 | 4.1× | 0.925 | 4.00 | 6703.30 | 4.1× |
| Uniform 3-bit | 0.904 | 3.00 | 4478.26 | 5.4× | 0.875 | 3.00 | 6754.93 | **5.4**× |
| ASTRO (Ours) | 0.926 | 3.88 | 2971.85 | **6.3**× | 0.925 | 3.85 | 5718.83 | 5.0× |
| | **MMLU Dataset** | | | | **GPQA-Diamond-MC Dataset** | | | |
| Original | 0.894 | 16.00 | 1496.48 | 1.0× | 0.694 | 16.00 | 7936.74 | 1.0× |
| Uniform 4-bit | 0.874 | 4.00 | 1476.68 | 4.1× | 0.677 | 4.00 | 7684.58 | 4.1× |
| Uniform 3-bit | 0.834 | 3.00 | 1802.39 | 4.4× | 0.616 | 3.00 | 7947.23 | 5.3× |
| ASTRO (Ours) | 0.878 | 3.83 | 1335.43 | **4.7**× | 0.667 | 3.91 | 5013.15 | **6.5**× |

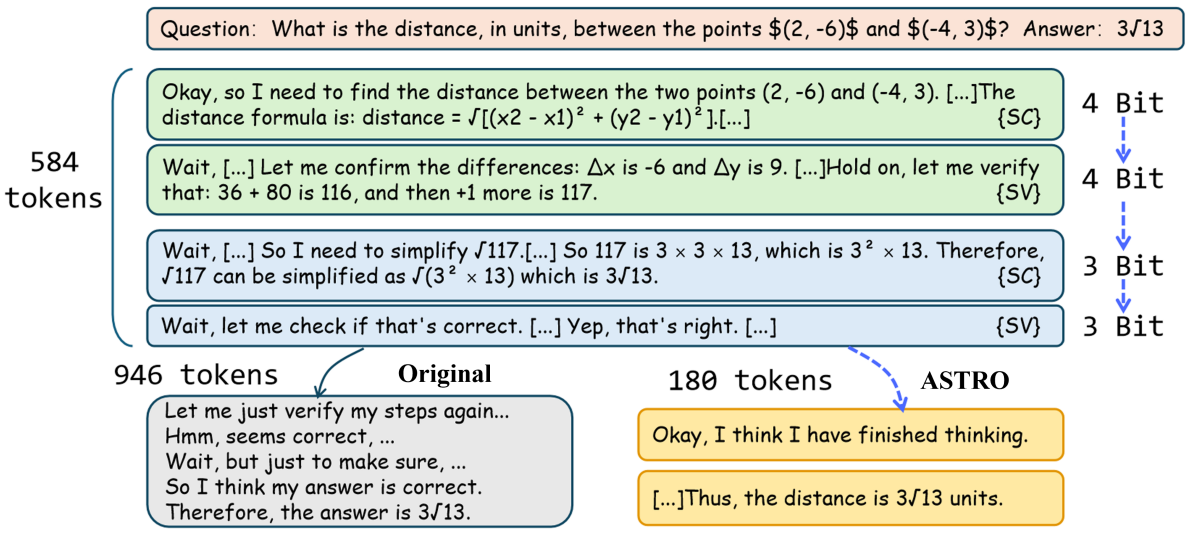



Figure 4: A case study demonstrating ASTRO's optimization process. ASTRO identifies reasoning phases, adaptively reduces precision for computational steps, and terminates early upon reaching a stable conclusion, pruning redundant verification steps.