Thermometer of Thoughts: Enhancing LLM's Exploration via Attention Temperature Modulation

Zhiyuan Yu[1], Shijian Xiao[1], Cam-Tu Nguyen[1], Zhangyue Yin[2], Lekai Xing[1], Wenzhong Li[1]*, Sanglu Lu[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]School of Computer Science, Fudan University
zhiyuan_yu@smail.nju.edu.cn, lwz@nju.edu.cn

Abstract

Improving the exploration of reasoning is essential for advancing Large Language Models' (LLMs) problem-solving performance. Current methods primarily rely on output-level stochasticity, which decode within fixed reasoning patterns of LLM and suffer from insufficient exploration. In this paper, we introduce adjusting **attention temperature** to directly modulate the model's internal focus during reasoning, which enables a dynamic shift between exploratory and focused processing. We reveal that moderate adjustments preserve LLM's reasoning capability while producing problem hardness-dependent benefits: higher temperatures facilitate solving complex tasks by encouraging wider exploration, whereas lower temperatures mitigate overthinking on simpler problems. Leveraging this insight, we propose a two-stage inference strategy: first, *attention temperature scaling* modulates the LLM's reasoning patterns to diversify the reasoning traces; then, a *difficulty-aware aggregation* scheme is introduced to effectively identify the most reliable solution from the generated candidates. Extensive evaluations show that our method improves Pass@10 by 6.78–14.20% and aggregation accuracy by 9.74% across 7 reasoning benchmarks.

1 Introduction

Recent years have witnessed remarkable advancements in the reasoning capabilities of large language models (LLMs) (Jaech et al., 2024; OpenAI, 2025; DeepSeek-AI, 2025; Yang et al., 2025). Techniques such as chain-of-thought (Kojima et al., 2023; Wei et al., 2023) and reinforcement learning (Ouyang et al., 2022; Rafailov et al., 2024) have been instrumental in empowering LLMs to tackle complex tasks including mathematics reasoning (Wang et al., 2025a; Balachandran et al., 2025), code generation (Yu et al., 2025; Li et al., 2025a), and agent scenarios (Zhu et al., 2025; Chakraborty et al., 2025). This evolution marks a significant leap towards more reliable and transparent artificial intelligence systems.

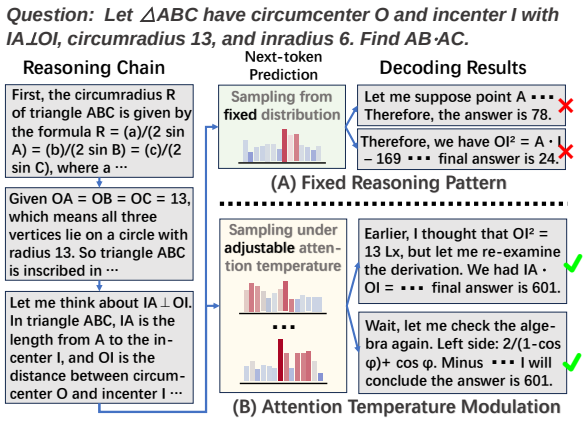


Figure 1: A case study of Qwen3-1.7B on AIME 2024 Problem 18, where we apply different exploration mechanism starting from step 4. **(A)** Standard LLM sampling relies on a fixed distribution, which cause the model to choose tokens like "let" or "therefore", yet both paths lead to incorrect answers. **(B)** Our attention temperature modulation enables adjustable sampling distributions. This allows the LLM to strategically sample tokens like "earlier" or "wait", triggering reflective reasoning that leads to correct answers.

Most of current methodologies rely on diversity of generation to enhance LLM's reasoning capability (Wang et al., 2023; Wu et al., 2025b; Shao et al., 2024), which ensures a rich set of potential problem-solving pathways is investigated to increase the likelihood of arriving at a correct solution. The standard approach to achieving output diversity is stochastic decoding (Schaeffer et al., 2025; Wu et al., 2025a), which draws tokens from the LLM's next-token-prediction distribution in each time step. While effective, this method only enables exploration within the model's fixed reasoning patterns without altering its underlying cognitive framework (Niu et al., 2025). This is analogous

* Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4355–4368
July 2-7, 2026

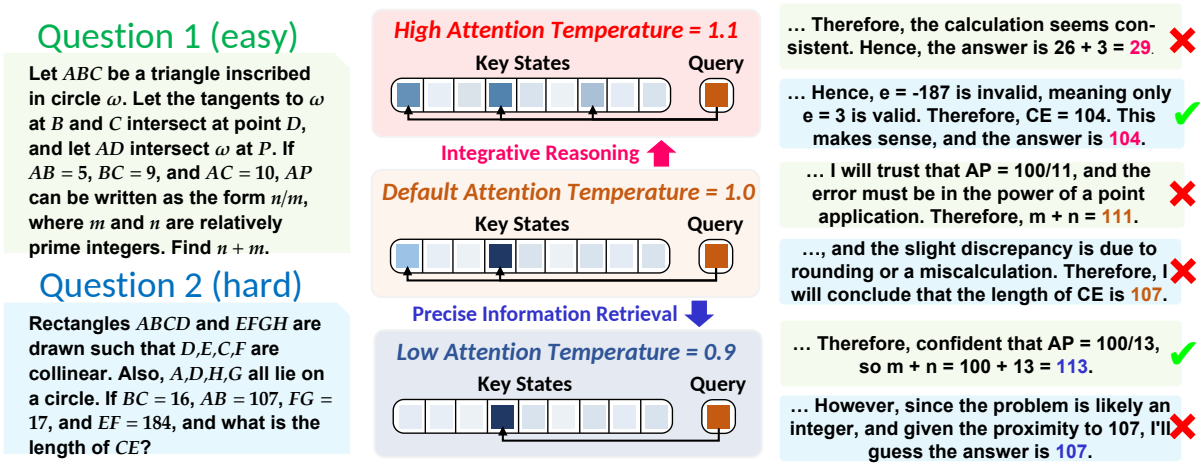


Figure 2: Enhancing LLM's reasoning exploration by adjusting the attention temperature.

to the limitation of a local optimization algorithm, which can efficiently sample nearby points in the solution space but lacks the mechanism to escape a local optimum and discover a fundamentally superior region. As shown in Fig. 1, such exploration mode is confined to a fixed next-token distribution, thus suffering from insufficient exploration.

To address this limitation, we propose adjusting **attention temperature**, which is the scaling parameter that controls the concentration of attention weights, to dynamically guide the LLM's internal reasoning patterns. Unlike stochastic sampling methods that only manipulate the output distribution, attention temperature operates directly within the attention layers, which is LLM's cognitive core (Huang et al., 2023; Zheng et al., 2024). By modulating the sharpness of attention weight distributions, it dynamically adjusts the model's cognitive focus during reasoning process, as shown in Fig. 2. Specifically, a high temperature smooths the distribution and promotes integrative reasoning by broadly attending to multiple key states; while a low temperature facilitates precise information retrieval by focusing on most critical states. Adjusting this parameter enables flexible switching between divergent exploration and focused execution during reasoning, thus allowing LLM to generate more diverse and adaptable outputs.

Our systematic experiments yield two principal observations: (1) Although moderate adjustments to attention temperature create a divergence from the pre-training regime, this manageable gap does not degrade the output quality or reasoning performance of LLMs; (2) The optimal temperature setting is difficulty-relevant: for challenging tasks, a higher temperature promotes broader exploration of the reasoning space, increasing the likelihood of discovering a correct solution; conversely, for simpler problems, a lower temperature curbs over-thinking (Chen et al., 2025b; Sui et al., 2025) and yields more concise and accurate reasoning chains with lower computational overhead.

Building on these insights, we propose a two-stage inference strategy to enhance the reasoning capability of LLMs: (1) We introduce *attention temperature scaling*, which diversifies exploration by sampling reasoning paths under varied attention temperature settings to increase the likelihood of reaching high-quality solutions. (2) We further propose a *difficulty-induced aggregation scheme* which estimates problem difficulty and then intelligently aggregates outputs from different temperature configurations through weighted voting.

Our contributions can be summarized as follows:

- We first propose a novel attention temperature modulation to enhance LLM reasoning exploration without compromising its reasoning capability.
- We propose a two-stage inference strategy combining attention temperature scaling and difficulty-induced aggregation to diversify reasoning pathways and consolidate LLM results.
- Extensive evaluations show that our method improves Pass@10 by 6.78–14.20% and aggregation accuracy by 9.74% across 7 reasoning benchmarks.

2 Related Work

2.1 Enhancement of LLM Reasoning

As the scaling of model parameters and training data reaches practical limits (Kaplan et al., 2020; Snell et al., 2025), test-time scaling (TTS)

approaches has emerged as a key paradigm for enhancing the reasoning capabilities of LLMs by dynamically allocating additional computational resources during inference (Zhang et al., 2025a). Inference-time scaling methods generally operate under two complementary paradigms: parallel scaling and sequential scaling. Parallel scaling (Qi et al., 2025; Tu et al., 2025) generates multiple independent reasoning paths for a single query, which are then aggregated through majority voting (Wang et al., 2023) or reward model (Lightman et al., 2023; Zhang et al., 2025b) to improve answer robustness. Sequential scaling (Chang et al., 2025), on the other hand, iteratively refines the model's answer over multiple steps, often through self-refine (Madaan et al., 2023; Shi et al., 2025a) or external feedback (Yin et al., 2024), enabling stepwise verification (Shi et al., 2025b) and extended reasoning depth (Chang et al., 2025).

Moreover, reinforcement learning (RL) methods (OpenAI, 2025; DeepSeek-AI, 2025) offers a more powerful approach for test-time scaling by iteratively refining a model's reasoning policy through feedback. Unlike inference-time scaling, RL-based approaches incentivize the emergence of advanced, self-improving reasoning strategies. By leveraging techniques that guide multi-step reasoning through learned reward feedback (Ouyang et al., 2022; Zhong et al., 2025) or directly optimize policies against verifiable task outcomes (Rafailov et al., 2024; Xiong et al., 2025), RL enables LLMs to autonomously extend and refine their reasoning chains. This has been widely applied to a diverse range of complex tasks including mathematical reasoning (Wang et al., 2025a) and agentic tasks (Zhu et al., 2025).

2.2 Analysis of LLM Reasoning Mechanism

Research on the reasoning mechanisms of LLMs seeks to understand how these models perform logical steps internally. This field, known as mechanistic interpretability, focuses on reverse-engineering the models' internal computations (Wang et al., 2025c). A key approach involves circuit analysis and the functional dissection of attention heads to identify specialized components crucial for reasoning, such as iteration heads that enable multi-step loops (Cabannes et al., 2024) and semantic induction heads that support in-context learning and factual recall (Ren et al., 2024). Concurrently, other work investigates the limitations and characteristic failures of LLM reasoning, revealing that models may generate computational steps without systematically validating results (Lu et al., 2025; Zhang, 2025), or may rely on latent shortcuts and spurious correlations rather than robust logic (Ding et al., 2024; Bronzini et al., 2024). Together, these insights form a foundation for designing more reliable, efficient, and transparent reasoning architectures.

3 Preliminary

3.1 Decoding Temperature

To introduce diversity in text generation, stochastic decoding methods sample tokens from the model's output distribution rather than deterministically selecting the most probable tokens. Given a context $\mathbf{x}_{<t} = (x_1, \ldots, x_{t-1})$, the model produces a probability distribution over the vocabulary $\mathcal{V}$:

$$P_\theta(v \mid \mathbf{x}_{<t}) = \frac{\exp(\mathbf{h}_t^\top \mathbf{W}_v)/\tau}{\sum_{v' \in \mathcal{V}} \exp(\mathbf{h}_t^\top \mathbf{W}_{v'})/\tau}, \quad (1)$$

where $\mathbf{W}_v$ denotes the output embedding for token v, and τ is the **decoding temperature**. Stochastic decoding methods sample next token $\mathbf{x}_t$ on this probability distribution. For example, **Top-k Sampling** restricts the sampling space to the k most probable tokens, and **Top-p Sampling** dynamically selects the smallest set of tokens whose cumulative probability exceeds threshold p. Both operate on LLM's output distribution, with no intention of altering LLM's inherent cognitive patterns.

3.2 Attention Temperature

Previous research indicates that feed-forward layers in LLMs are typically used for storing knowledge (Geva et al., 2021; Chen et al., 2025a), while attention layers, due to their dynamic contextual awareness and ability to model interrelationships, are associated with complex functions like reasoning (Chen et al., 2025c; Jin et al., 2025).

In attention blocks, attention weights α_i are typically derived through a softmax transformation of scaled dot-products between the query and keys:

$$\alpha_i = \exp\left(\frac{\mathbf{q}^\top \mathbf{k}_i}{\sqrt{d_k}}\right) \Big/ \sum_{j=1}^{n} \exp\left(\frac{\mathbf{q}^\top \mathbf{k}_j}{\sqrt{d_k}}\right). \quad (2)$$

Here, the scaling factor $1/\sqrt{d_k}$ prevents gradient vanishing issues that arise from large dot-product values in high-dimensional spaces.

Based on it, the concept of *attention temperature* is introduced to further control the "peakiness"

or concentration of the attention distribution. By incorporating a temperature parameter $\tau > 0$ into the softmax computation, we obtain:

$$\alpha_i(\tau) = \exp\left(\frac{\mathbf{q}^\top \mathbf{k}_i}{\tau\sqrt{d_k}}\right) \Big/ \sum_{j=1}^{n} \exp\left(\frac{\mathbf{q}^\top \mathbf{k}_j}{\tau\sqrt{d_k}}\right). \tag{3}$$

In practical implementations, we introduce attention temperature by directly modifying the scaling factor ($1/\sqrt{d_k} \rightarrow 1/(\tau\sqrt{d_k})$). This obviates the necessity of incorporating an additional configuration parameter to the source code of LLMs.

```
T = your attention temperature
for i in range(num_hidden_layers):
    layer = model.model.layers[i]
    layer.self_attn.scaling /= T
```

3.3 Reasoning Patterns

Regulating the attention temperature essentially modifies the attention entropy, i.e., the degree of peakedness or uniformity in attention, which can be formulated as follows:

Definition 1 (Attention Entropy) *Given the attention weights $\mathbf{a}^t \in \mathbb{R}^{t-1}$ from query at time step t to previous keys, the attention entropy $H_{attn}(\mathbf{a}^t)$ is computed on the normalized top-K weights. Specifically, let $\{a^t_{(j)}\}_{j=1}^{K}$ be the K largest weightsLet $\hat{a}^t_j = a^t_{(j)} / \sum_{k=1}^{K} a^t_{(k)}$, we define:*

$$H_{attn}(\mathbf{a}^t) = -\sum_{j=1}^{K} \hat{a}^t_j \log \hat{a}^t_j. \tag{4}$$

Here K is a fixed hyperparameter independent of sequence length.

As the counterpart, decoding entropy measures the uncertainty of LLM's output:

Definition 2 (Decoding Entropy) *Given next-token probability distribution $\mathbf{p}^t \in \mathbb{R}^V$ at time step t, the decoding entropy $H_{pred}(\mathbf{p}^t)$ is defined as:*

$$H_{pred}(\mathbf{p}^t) = -\sum_{i=1}^{V} p^t_i \log p^t_i. \tag{5}$$

Prior works (Wang et al., 2025b; Li et al., 2025b) have suggested that enhancements of LLM reasoning is primarily driven by optimizing a small fraction of "forking tokens" with high decoding entropy, including logical connectives (e.g., "however", "wait") and key reasoning terms (e.g., "suppose", "define"). Intuitively, these tokens signify cognitive shifts in the model's reasoning process.

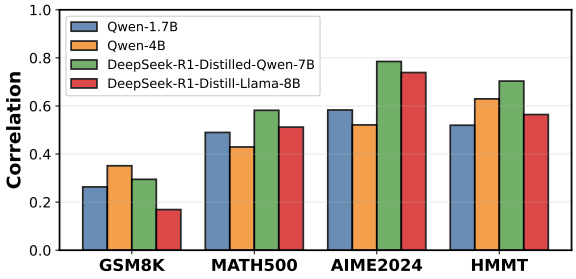


Figure 3: Solution tokens are grouped by decoding entropy, and the correlation between average decoding entropy and attention entropy is computed across chunks. The correlation is low on simple tasks (GSM8K) but stronger on challenging tasks.

For instance, "wait" may indicate a moment of reflection, while "suppose" introduces a hypothetical scenario. Such reasoning steps often require integrating information from previous context.

In this work, we observe that the generation of tokens with high decoding entropy tends to co-occur with high attention entropy, indicating that LLM attends to a broader set of information when producing these pivotal tokens. We quantitatively validate this correlation across four mathematical reasoning datasets and visualize the results in Figure 3. It suggests that elevated attention entropy may support greater reflection and exploration during reasoning, while lower attention entropy encourages more deterministic thinking. Inspired by this finding, we propose to modulate the attention temperature to steer the reasoning patterns of LLMs.

4 How attention temperature influence LLM's reasoning behaviour?

Prior research on attention temperature has primarily focused on its optimization during the training phase of LLMs (Zou et al., 2024; Ram et al., 2025), while inference typically relies on a fixed and static value. Therefore, directly adjusting attention temperature at inference stage will create a train–inference gap, which may potentially lead to unpredictable and suboptimal model behaviors. In this paper, we rule out the existence of this risk, instead, we reveal that LLM's reasoning exploration can be enhanced via attention temperature modulation during decoding.

4.1 Impact on Output Quality

To assess the impact of attention temperature t on output quality, we adopt the following evaluation protocol. Let $Q = \{q_1, q_2, \ldots, q_n\}$ denote

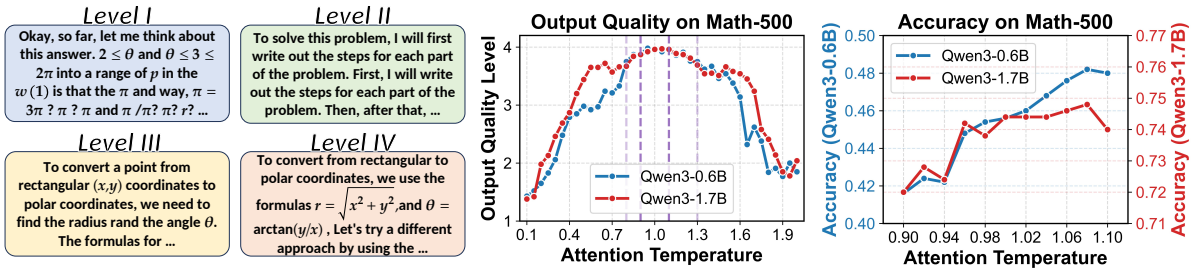


Figure 4: We show examples for each output quality level. Our analysis on Qwen3-1.7B reveals that variations in attention temperature between 0.9 and 1.1 have negligible effects on LLM output quality. Within this range, LLM's reasoning performance slightly improves as the attention temperature increases.

a benchmark of n questions. For each question $q_i \in Q$, the model generates an answer $a_i(t)$ under attention temperature t. Each question-answer pair $(q_i, a_i(t))$ is then scored by an evaluation function f according to four predefined quality levels:

Level I: Output consists of nonsensical or garbled tokens. **Level II**: Output is grammatically coherent but irrelevant to the given task. **Level III**: Output contains contextually appropriate sentences but lacks coherent reasoning. **Level IV**: Output demonstrates a clear reasoning process, even if the final answer is incorrect.

The overall output quality metric $S(t)$ at attention temperature t is computed as the mean score across all questions in the benchmark:

$$S(t) = \frac{1}{n} \sum_{i=1}^{n} f(q_i, a_i(t)). \tag{6}$$

Our experiment results with DeepSeek-R1 as evaluator f (Figure 4) show that output quality remains stable for attention temperatures between 0.9 and 1.1, with a limited effect from 0.8 to 1.3. However, temperatures below 0.5 or above 1.7 frequently lead to reasoning chains irrelevant to the input, indicating that only extreme values substantially degrade output quality.

Building on these findings, we further examine LLM's task-performance on reasoning benchmarks under the optimal temperature range of 0.9 to 1.1. Results on the MATH-500 benchmark (Figure 4) indicate a slight overall improvement as temperature increases. Notably, this positive correlation is more consistent on challenging tasks, whereas performance on simpler benchmarks like GSM8K shows a marginal decline with higher temperature (as shown in Appendix). We will elaborate on this phenomenon in following section.

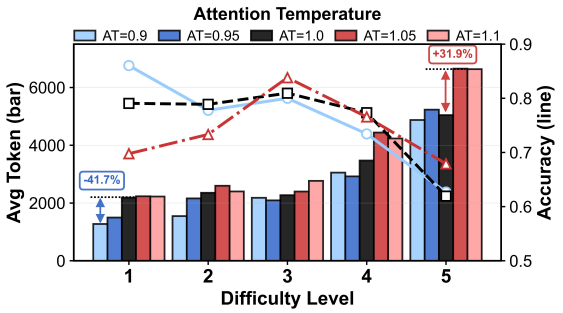


Figure 5: Accuracy (line) and average tokens (bars) for Qwen3-1.7B on MATH-500 across attention temperatures, grouped by difficulty level.

4.2 Impact on Reasoning Capability

For deeper investigation, we examine how adjusting attention temperature shapes internal reasoning in LLMs. A central issue is that whether tuning attention temperature, rather than decoding temperature, can foster more diverse and adaptive reasoning pathways.

Our analysis specifically examines the performance variations resulting from attention temperature adjustments on reasoning benchmarks with varying difficulty levels. Using the results from Qwen3-1.7B on the MATH-500 benchmark (Figure 5) as a representative example, we identify two distinct patterns:

(1) For problems of lower difficulty (e.g., Level 1-2), reducing the attention temperature yields more concise responses, which helps prevent the model from "over-thinking" and avoid potential performance degradation associated with elongated reasoning chains.

(2) Conversely, for highly challenging problems (e.g., Level 4-5), elevating the attention temperature promotes the generation of longer and more exploratory reasoning paths. This effectively broadens the model's thinking basis and increases the

Table 1: Pass@K (%) of different scaling methods, including *random sampling (RS), decoding temperature sampling (DTS) and attention temperature sampling (ATS)*, on seven reasoning datasets.

Model	Method	MATH500		AIME2024		AIME2025		HMMT		GSM8K		GPQA		HumanEval	
		P@1	P@10	P@1	P@10	P@1	P@10	P@1	P@10	P@1	P@10	P@1	P@10	P@1	P@10
Hunyuan-1.8B-Instruct	RS	50.3	86.5	15.2	36.1	18.4	46.1	10.0	26.4	53.4	74.3	22.4	74.8	28.7	65.8
	DTS	52.4	88.1	16.5	**44.1**	20.3	46.4	12.9	30.1	57.2	76.3	24.7	81.9	31.6	69.1
	ATS	**53.9**	**89.4**	**20.3**	43.6	**24.7**	**50.8**	**16.5**	**33.7**	**59.7**	**77.3**	**25.1**	**83.0**	**35.2**	**72.8**
Qwen3-0.6B	RS	49.1	83.6	3.7	15.4	12.0	56.4	13.3	25.0	47.4	78.0	24.4	73.7	20.7	41.5
	DTS	49.7	84.8	4.0	16.2	15.3	65.3	13.4	31.5	48.5	79.6	26.5	74.2	25.6	44.5
	ATS	**50.7**	**85.5**	**5.4**	**16.9**	**16.9**	**67.1**	**17.7**	**36.7**	**49.0**	**80.4**	**26.8**	**74.4**	**26.8**	**48.8**
Qwen3-1.7B	RS	72.4	92.1	16.3	47.9	23.2	51.4	13.2	31.5	72.2	91.6	24.9	70.8	42.7	68.3
	DTS	73.5	92.8	21.0	**51.2**	**30.3**	**56.3**	16.4	35.1	76.1	94.5	26.1	73.9	47.6	72.6
	ATS	**74.0**	**93.5**	**24.8**	50.0	26.1	53.5	**17.1**	**36.3**	**76.6**	**94.7**	**26.9**	**74.0**	**48.2**	**78.1**
DeepSeek-R1-Distilled-Qwen-1.5B	RS	76.6	90.5	16.2	42.9	20.7	43.4	11.7	26.7	55.8	90.7	22.7	66.1	3.7	12.2
	DTS	78.2	92.9	17.7	44.8	**23.1**	**46.9**	13.0	33.4	62.6	91.9	**24.2**	72.5	4.9	13.4
	ATS	**78.7**	**93.4**	**20.5**	**47.0**	22.3	46.4	**14.4**	**33.8**	**62.8**	**92.6**	24.1	**73.4**	**6.1**	**14.0**

likelihood of arriving at the correct solution.

This phenomenon can be interpreted through how temperature modulates the softmax distribution within the attention mechanism. A lower temperature sharpens the distribution, encouraging determinism and focus that is beneficial for straightforward problems. A higher temperature softens the distribution, introducing beneficial stochasticity that allows the model to consider a wider array of information or reasoning steps for complex problems. Compared to adjusting the decoding temperature which primarily affects token-level sampling diversity, modulating the attention temperature directly influences the model's focus and information routing during reasoning, leading to more fundamental changes in the reasoning patterns.

5 Thermometer of Thoughts

In this section, we propose a two-stage inference strategy for LLM sampling: first, *attention temperature scaling* modulates the LLM's reasoning patterns to diversify the reasoning traces; then, a *difficulty-aware aggregation* scheme is applied to effectively identify the most reliable solution from the generated candidates. We apply and evaluate these methods under the following benchmarks, models, and implementation settings:

Datasets: We primarily focus on mathematical reasoning problems, including GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), AIME2024, AIME2025 (Team, 2025), and HMMT (Balunović et al., 2025). Additionally, evaluations are also performed on the general reasoning task GPQA (Rein et al., 2024) and coding task HumanEval (Chen et al., 2021).

Models: To evaluate our proposed method, we employed four recently released, reasoning-specialized large language models, including Hunyuan-1.8B-Instruct (Cao et al., 2025), Qwen3-0.6B, Qwen3-1.7B (Yang et al., 2025), and DeepSeek-R1-Distilled-Qwen-1.5 (DeepSeek-AI, 2025).

Implementations: Our experiments were conducted on four NVIDIA RTX 4090 GPUs, with parallel acceleration implemented using vLLM.

5.1 Attention Temperature Scaling

To alter the model's focus during the reasoning process for output diversity, we introduce attention temperature scaling. In our experiments, we systematically vary the attention temperature from 0.9 to 1.1 in increments of 0.02. For each temperature value, we generate 10 reasoning traces per question. To evaluate the quality of generated solutions, we employ the *Pass@K* which measures the probability that at least one of k generated solutions passes all verification tests. Let n be the total number of generated samples and c the number of correct ones, it is defined as:

$$Pass@K = 1 - \binom{n-c}{k} \Big/ \binom{n}{k}. \quad (7)$$

We compare our attention temperature scaling approach against two established baselines for generating diverse reasoning paths. First, the *random sampling* baseline generates multiple reasoning chains through standard stochastic sampling with fixed sampling temperature ($\tau_{pt} = 0.8$). Second, the *decoding temperature scaling* baseline (Wu et al., 2025a) varies the decoding temperature from 0.2 to 1.2 in increments of 0.1 and also generate 10 reasoning chains per temperature setting.

Table 2: Relative improvement (%) of D-induced over Majority voting across datasets. $\Delta = \frac{\text{D-induced} - \text{Majority}}{\text{Majority}} \times 100\%$.

Model	Method	MATH500	AIME2024	AIME2025	HMMT	GSM8K	GPQA
Hunyuan-1.8B-Instruct	Majority	64.40	50.00	36.67	13.33	77.13	31.31
	D-induced	66.80	56.67	43.33	16.67	78.70	32.32
	Δ (%)	+3.73	+13.34	+18.16	+25.06	+2.04	+3.23
Qwen3-0.6B	Majority	58.20	6.67	13.33	10.00	55.72	25.76
	D-induced	59.80	6.67	16.67	13.33	57.48	27.27
	Δ (%)	+2.74	+0.00	+25.00	+33.30	+3.16	+5.86
Qwen3-1.7B	Majority	74.20	43.33	33.33	16.67	82.56	29.29
	D-induced	76.00	46.67	36.67	20.00	84.21	30.30
	Δ (%)	+2.43	+7.71	+10.02	+20.00	+2.00	+3.45
DeepSeek-R1-Distilled-Qwen-1.5B	Majority	78.80	30.00	30.00	16.67	79.98	23.23
	D-induced	81.20	36.67	30.00	20.00	82.41	24.24
	Δ (%)	+3.05	+22.22	0.00	19.98	+3.04	+4.35

Experimental Results: The results are shown in Table 1. It can be observed that on mathematical reasoning tasks, our proposed attention temperature sampling method achieves improvements of 6.78% and 14.20% over the conventional next-token random sampling on the Qwen3-1.7B and Qwen3-0.6B respectively. The improvements also consistently exceed 1.3% compared to decoding temperature sampling. Our method also shows significant gains on general reasoning tasks. This indicates that adjusting the attention temperature can indeed enhance output diversity in LLMs.

5.2 Difficulty-Induced Aggregation

Building on attention temperature scaling, we propose a difficulty-induced aggregation method to optimally consolidate the results from different temperature configurations.

Let $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_k\}$ be a set of k distinct attention temperature values, where $\tau_1 < \tau_2 < \cdots < \tau_k$. For each temperature τ_i, we generate n reasoning paths, resulting in a total of nk candidate solutions for a given problem q. We first estimate the problem difficulty by analyzing the consensus among low-temperature generations. Let $\mathcal{T}_{\text{low}} = \{\tau_1, \tau_2, \ldots, \tau_{k'}\}$ be the subset of k' lowest temperatures ($k' < k$). For each temperature $\tau_i \in \mathcal{T}_{\text{low}}$, we have answer set $A_i = \{a_{i1}, a_{i2}, \ldots, a_{in}\}$.

The confidence score for low-temperature generations is computed as:

$$C_{\text{low}} = \max_{a \in \bigcup_{i=1}^{k'} A_i} \frac{\sum_{i=1}^{k'} \sum_{j=1}^{n} \mathbb{I}(a_{ij} = a)}{nk'}. \quad (8)$$

Based on the confidence threshold θ, we employ different aggregation strategies

If $C_{\text{low}} > \theta$, we classify the problem as simple and directly employ majority voting on the low-temperature generations:

$$a_{\text{final}} = \arg \max_{a \in \bigcup_{i=1}^{k'} A_i} \sum_{i=1}^{k'} \sum_{j=1}^{n} \mathbb{I}(a_{ij} = a). \quad (9)$$

If $C_{\text{low}} \leq \theta$, the problem is considered challenging. We then aggregate all generations with temperature-dependent weights:

$$a_{\text{final}} = \arg \max_{a \in \bigcup_{i=1}^{k} A_i} \sum_{i=1}^{k} w(\tau_i) \cdot \sum_{j=1}^{n} \mathbb{I}(a_{ij} = a). \quad (10)$$

The weighting function $w(\tau_i)$ assigns higher weights to high-temperature generations:

$$w(\tau_i) = \frac{\exp(\beta \cdot \tau_i)}{\sum_{j=1}^{k} \exp(\beta \cdot \tau_j)}, \quad (11)$$

where $\beta > 0$ is a scaling parameter that controls the emphasis on high-temperature explorations.

In our implementations, we have found that in some cases, LLMs will fail to produce an answer within the token budget due to over-thinking, leading to *None* as the final result. As a result, we filter out such responses for majority voting.

This adaptive approach ensures computational efficiency for simple problems while maintaining robust exploration capabilities for challenging tasks, effectively balancing the exploration-exploitation tradeoff in LLM reasoning.

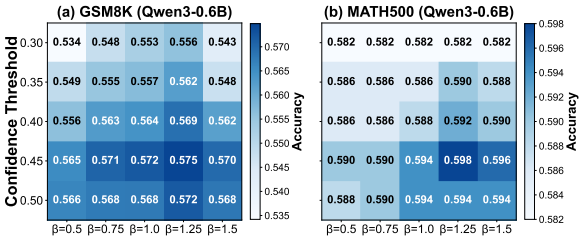


Figure 6: Hyperparameter analysis for D-induced.

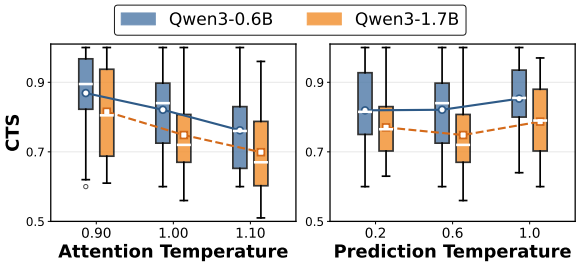


Figure 7: Reasoning Patterns Analysis.

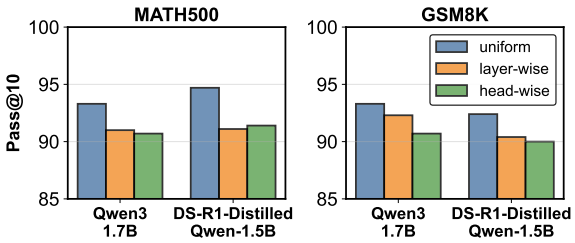


Figure 8: Inconsistent attention temperature across heads can lead to degraded reasoning performance.

Experimental Results: In Table 2, we compare our difficulty-induced aggregation (D-induced) method against the majority voting. Overall, D-induced achieves an average relative improvement of 9.74% across all settings. The gains are most pronounced on challenging competition datasets (e.g., AIME, HMMT) with an average improvement of 16.2%, compared to more moderate gains on GSM8K and GPQA (2.6% and 4.2%, respectively). This demonstrates that D-induced is a more suitable aggregation scheme for attention-temperature scaling, especially for complex reasoning tasks.

Hyperparameter Analysis: We conduct hyperparameter analysis using the Qwen3-0.6B model on the GSM8K and MATH500 datasets, focusing on the impact of the confidence threshold C_{low} and the weighting parameter β on the aggregation performance. As shown in Figure 6, both datasets achieve optimal performance when $C_{\text{low}} = 0.45$ and $\beta = 1.25$.

6 Further Discussion

6.1 Reasoning Patterns Analysis

To systematically interpret the impact of attention temperature on the reasoning process, we decompose the model's reasoning chain into consecutive segments, each representing a coherent step (e.g., a logical deduction or evidence integration). The relationship between adjacent segments is categorized into two types: (1) **continuation relations**, where the subsequent segment follows naturally from the previous one (e.g., entailment or elaboration), and (2) **transition relations**, which involve shifts such as reflection, correction, or exploration of alternatives (e.g., sentences begin with "wait" or "alternatively"). This analysis allows us to examine how attention temperature influences the logical flow in a semantic level.

To quantify the reasoning pattern, we propose the **Continuation Tendency Score (CTS)**, which measures the model's propensity for linear or divergent reasoning. Formally, for a dataset $\mathcal{D}$ with N samples, denote that sample i have a reasoning chain segmented into S_i parts, i.e., it has $S_i - 1$ adjacent pairs. For each pair (s_j, s_{j+1}), indicator $\mathbb{I}_{\text{cont}}(s_j, s_{j+1})$ equals 1 if the relation is a continuation. The sample-level continuation is defined:

$$\textbf{CTS}_i = \frac{1}{S_i - 1} \sum_{j=1}^{S_i - 1} \mathbb{I}_{\text{cont}}(s_j, s_{j+1}). \quad (12)$$

We use DeepSeek-R1 for measuring this metric, with detailed configurations and prompts provided in Table 5 the Appendix. In Figure 7, we present experimental results on `AIME 2024`. It shows that as the attention temperature increases, the distribution of the CTS metric gradually shifts downward, indicating that the LLMs engage more readily in behaviors such as reflection and alternative exploration. In contrast, no such trend is observed with the increase of decoding temperature.

6.2 Layer/Head-wise Adjustment

In previous experiments, we applied a uniform attention temperature across all attention heads in the LLM. We further explore the possibility of layer-wise and head-wise adjustments. However, given the large number of attention heads in modern LLMs, exhaustively searching all possible configurations leads to an exponentially large exploration space. Instead, we employ a simple random sampling strategy: for the layer-wise method, we uniformly sample a value from the set 0.90, 0.95, 1.00, 1.05, 1.10 for all heads within the same layer; for the head-wise method, we sample independently for each head. We present the results on two mathematical reasoning tasks, as shown in Figure 8. It can be observed that the layer-wise method leads to an average decrease of 2.37% in Pass@10, while the head-wise method exhibits a further reduction of 2.91%. This suggests that a

consistent attention temperature across the model helps maintain stable reasoning patterns, thereby facilitating higher performance.

7 Conclusion

In this paper, we demonstrate that modulating attention temperature dynamically can shift LLM reasoning between exploratory and focused modes, thus enhancing LLM's output diversity without compromising quality. We show that higher temperatures benefit complex tasks by broadening exploration, while lower temperatures prevent overthinking on simpler problems. Our attention temperature scaling and difficulty-induced aggregation consistently boost reasoning performance across benchmarks, offering a novel pathway for efficient LLM reasoning.

Limitations

Since our work involves modifying the source code of LLMs, we are unable to evaluate the performance of API-based models like GPT-5. Additionally, due to resource constraints, we did not investigate whether adjusting the attention temperature helps promote reasoning exploration in larger-scale models such as Qwen3-32B. Moreover, owing to time limitations, we were unable to conduct more reasoning trajectory sampling to mitigate issues arising from randomness.

Ethical Considerations

This work is based on the publicly available datasets, all of which contain English text, and the associated questions are also in English. We comply with all dataset licenses, and confirm the content contains neither private nor offensive information. We utilized Claude-3.7-Sonnet to assist us with code generation.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62572236,W2532049,62441225), the Basic Research Program of Jiangsu Province (Grant Nos. BK20222003, BK20251198, BK20253011), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Sino-German Institutes of Social Computing.

References

Vidhisha Balachandran, Jingya Chen, Lingjiao Chen, Shivam Garg, Neel Joshi, Yash Lara, John Langford, Besmira Nushi, Vibhav Vineet, Yue Wu, et al. 2025. Inference-time scaling for complex tasks: Where we stand and what lies ahead. *arXiv preprint arXiv:2504.00294*.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. 2025. Matharena: Evaluating llms on uncontaminated math competitions.

Marco Bronzini, Carlo Nicolini, Bruno Lepri, Jacopo Staiano, and Andrea Passerini. 2024. Unveiling llms: The evolution of latent representations in a dynamic knowledge graph.

Vivien Cabannes, Charles Arnal, Wassim Bouaziz, Xingyu Alice Yang, Francois Charton, and Julia Kempe. 2024. Iteration head: A mechanistic study of chain-of-thought. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, Tiankai Hang, Duojun Huang, Jie Jiang, Zhengkai Jiang, Weijie Kong, Changlin Li, Donghao Li, Junzhe Li, Xin Li, Yang Li, Zhenxi Li, Zhimin Li, Jiaxin Lin, Linus, Lucaz Liu, Shu Liu, Songtao Liu, Yu Liu, Yuhong Liu, Yanxin Long, Fanbin Lu, Qinglin Lu, Yuyang Peng, Yuanbo Peng, Xiangwei Shen, Yixuan Shi, Jiale Tao, Yangyu Tao, Qi Tian, Pengfei Wan, Chunyu Wang, Kai Wang, Lei Wang, Linqing Wang, Lucas Wang, Qixun Wang, Weiyan Wang, Hao Wen, Bing Wu, Jianbing Wu, Yue Wu, Senhao Xie, Fang Yang, Miles Yang, Xiaofeng Yang, Xuan Yang, Zhantao Yang, Jingmiao Yu, Zheng Yuan, Chao Zhang, Jian-Wei Zhang, Peizhen Zhang, Shi-Xue Zhang, Tao Zhang, Weigang Zhang, Yepeng Zhang, Yingfang Zhang, Zihao Zhang, Zijian Zhang, Penghao Zhao, Zhiyuan Zhao, Xuefei Zhe, Jianchen Zhu, and Zhao Zhong. 2025. Hunyuanimage 3.0 technical report.

Souradip Chakraborty, Mohammadreza Pourreza, Ruoxi Sun, Yiwen Song, Nino Scherrer, Furong Huang, Amrit Singh Bedi, Ahmad Beirami, Jindong Gu, Hamid Palangi, and Tomas Pfister. 2025. On the role of feedback in test-time scaling of agentic ai workflows.

Kaiyan Chang, Yonghao Shi, Chenglong Wang, Hang Zhou, Chi Hu, Xiaoqian Liu, Yingfeng Luo, Yuan Ge, Tong Xiao, and Jingbo Zhu. 2025. Step-level verifier-guided hybrid test-time scaling for large language models.

Lei Chen, Joan Bruna, and Alberto Bietti. 2025a. Distributional associations vs in-context reasoning: A study of feed-forward and attention layers.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large

language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

Yaofo Chen, Zeng You, Shuhai Zhang, Haokun Li, Yirui Li, Yaowei Wang, and Mingkui Tan. 2025c. Core context aware transformers for long context language modeling.

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.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Mengru Ding, Hanmeng Liu, Zhizhang Fu, Jian Song, Wenbo Xie, and Yue Zhang. 2024. Break the chain: Large language models can be shortcut reasoners.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Zhongzhan Huang, Mingfu Liang, Jinghui Qin, Shanshan Zhong, and Liang Lin. 2023. Understanding self-attention mechanism via dynamical system perspective.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Mingyu Jin, Kai Mei, Wujiang Xu, Mingjie Sun, Ruixiang Tang, Mengnan Du, Zirui Liu, and Yongfeng Zhang. 2025. Massive values in self-attention modules are the key to contextual knowledge understanding.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

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

Yang Li, Zhichen Dong, Yuhan Sun, Weixun Wang, Shaopan Xiong, Yijia Luo, Jiashun Liu, Han Lu, Jiamang Wang, Wenbo Su, Bo Zheng, and Junchi Yan. 2025b. Attention illuminates llm reasoning: The preplan-and-anchor rhythm enables fine-grained policy optimization.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step.

Jiahao Lu, Ziwei Xu, and Mohan Kankanhalli. 2025. Reasoning llms are wandering solution explorers.

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.

Qian Niu, Junyu Liu, Ziqian Bi, Pohsun Feng, Benji Peng, Keyu Chen, Ming Li, Lawrence KQ Yan, Yichao Zhang, Caitlyn Heqi Yin, Cheng Fei, Tianyang Wang, Yunze Wang, Silin Chen, Ming Liu, Ziyuan Qin, Riyang Bao, Xinyuan Song, and Zekun Jiang. 2025. Large language models and cognitive science: A comprehensive review of similarities, differences, and challenges.

OpenAI. 2025. Gpt-5 is here.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Jianing Qi, Xi Ye, Hao Tang, Zhigang Zhu, and Eunsol Choi. 2025. Learning to reason across parallel samples for llm reasoning.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model.

Dhananjay Ram, Wei Xia, and Stefano Soatto. 2025. Learning to focus: Focal attention for selective and scalable transformers.

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*.

Jie Ren, Qipeng Guo, Hang Yan, Dongrui Liu, Quanshi Zhang, Xipeng Qiu, and Dahua Lin. 2024. Identifying semantic induction heads to understand in-context learning.

Rylan Schaeffer, Joshua Kazdan, John Hughes, Jordan Juravsky, Sara Price, Aengus Lynch, Erik Jones, Robert Kirk, Azalia Mirhoseini, and Sanmi Koyejo. 2025. How do large language monkeys get their power (laws)?

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

Haizhou Shi, Ye Liu, Bo Pang, Zeyu Leo Liu, Hao Wang, Silvio Savarese, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2025a. Ssr: Socratic self-refine for large language model reasoning.

Weijie Shi, Han Zhu, Jiaming Ji, Mengze Li, Jipeng Zhang, Ruiyuan Zhang, Jia Zhu, Jiajie Xu, Sirui Han, and Yike Guo. 2025b. Legalreasoner: Step-wised verification-correction for legal judgment reasoning.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling llm test-time compute optimally can be more effective than scaling model parameters. In *The Thirteenth International Conference on Learning Representations*. OpenReview.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models.

AoPSWiki Team. 2025. Aime problems and solutions.

Shangqing Tu, Yaxuan Li, Yushi Bai, Lei Hou, and Juanzi Li. 2025. Deepprune: Parallel scaling without inter-trace redundancy.

Jian Wang, Boyan Zhu, Chak Tou Leong, Yongqi Li, and Wenjie Li. 2025a. Scaling over scaling: Exploring test-time scaling pareto in large reasoning models. *arXiv preprint arXiv:2505.20522*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025b. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning.

Xu Wang, Yan Hu, Wenyu Du, Reynold Cheng, Benyou Wang, and Difan Zou. 2025c. Towards understanding fine-tuning mechanisms of llms via circuit analysis.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

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

Yuheng Wu, Azalia Mirhoseini, and Thierry Tambe. 2025a. On the role of temperature sampling in test-time scaling.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2025b. When more is less: Understanding chain-of-thought length in llms.

Wei Xiong, Wenting Zhao, Weizhe Yuan, Olga Golovneva, Tong Zhang, Jason Weston, and Sainbayar Sukhbaatar. 2025. Stepwiser: Stepwise generative judges for wiser reasoning.

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

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Xiaonan Li, Junqi Dai, Qinyuan Cheng, Xuanjing Huang, and Xipeng Qiu. 2024. Reasoning in flux: Enhancing large language models reasoning through uncertainty-aware adaptive guidance. In *ACL (1)*, pages 2401–2416.

Zhaojian Yu, Yinghao Wu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. 2025. Z1: Efficient test-time scaling with code. *arXiv preprint arXiv:2504.00810*.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, et al. 2025a. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*.

Zefeng Zhang, Xiangzhao Hao, Hengzhu Tang, Zhenyu Zhang, Jiawei Sheng, Xiaodong Li, Zhenyang Li, Li Gao, Daiting Shi, Dawei Yin, and Tingwen Liu. 2025b. Cooper: A unified model for cooperative perception and reasoning in spatial intelligence.

Zheng Zhang. 2025. Comprehension without competence: Architectural limits of llms in symbolic computation and reasoning.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. Attention heads of large language models: A survey.

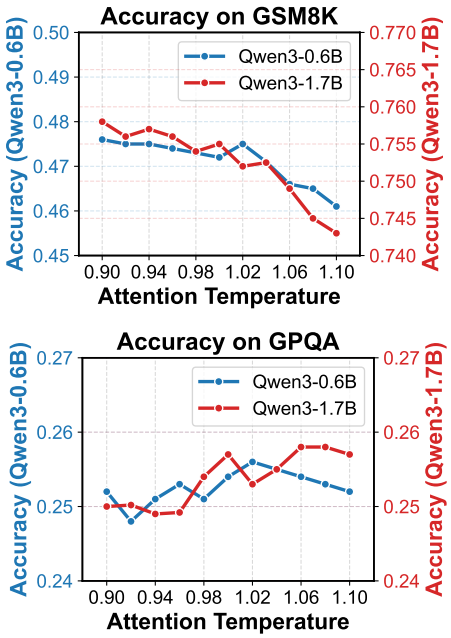
Jialun Zhong, Wei Shen, Yanzeng Li, Songyang Gao, Hua Lu, Yicheng Chen, Yang Zhang, Wei Zhou, Jinjie Gu, and Lei Zou. 2025. A comprehensive survey of reward models: Taxonomy, applications, challenges, and future.

King Zhu, Hanhao Li, Siwei Wu, Tianshun Xing, Dehua Ma, Xiangru Tang, Minghao Liu, Jian Yang, Jiaheng Liu, Yuchen Eleanor Jiang, et al. 2025. Scaling test-time compute for llm agents. *arXiv preprint arXiv:2506.12928*.

Yixiong Zou, Ran Ma, Yuhua Li, and Ruixuan Li. 2024. Attention temperature matters in vit-based cross-domain few-shot learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 116332–116354. Curran Associates, Inc.

A How adjusting attention temperature affects reasoning performance?

In the main text, we compared the impact of different attention temperatures on the reasoning performance of LLMs on MATH500. Here, we provide results on two additional datasets: GSM8K and GPQA. The problems within the former are relatively simpler, so when the attention temperature is low, LLMs can engage in focused reasoning, thereby avoiding the drawbacks of overthinking and achieving better performance. In contrast, the effect of attention temperature on GPQA is less pronounced.



B Dataset

Our comprehensive benchmark is composed of four challenging datasets meticulously selected to evaluate the advanced mathematical reasoning and deep scientific knowledge capabilities of LLMs. These datasets represent problems at the highest echelons of academic competition and expert-level understanding. Here are the detailed descriptions of the datasets included in our benchmark:

MATH-500 (Hendrycks et al., 2021): A comprehensive collection of 500 mathematical problems spanning diverse fields such as algebra and number theory. This dataset is designed to evaluate the general mathematical reasoning ability of models across a wide range of difficulties and topics.

AIME (Team, 2025): Features problems from the 2024 and 2025 American Invitational Mathematics Examination, a prestigious high school mathematics contest known for its exceptionally challenging problems. The dataset contains 60 problems that require deep mathematical insight and creative problem-solving skills.

HMMT (Balunović et al., 2025): Comprises problems from the Harvard-MIT Mathematics Tournament, one of the most prestigious high school mathematics contests in the United States. The problems in this dataset are renowned for their unique difficulty and require sophisticated mathematical thinking.

GPQA-Diamond (Rein et al., 2024): A challenging dataset consisting of 198 multiple-choice questions at a graduate-level difficulty across biology, physics, and chemistry. The questions were crafted by domain experts, and even specialists in these fields achieve only about 65% accuracy, making this an extremely rigorous test of specialized scientific knowledge.

GSM8K (Cobbe et al., 2021): The Grade School Math 8K dataset contains 8,500 high-quality, linguistically diverse grade school mathematics word problems created by OpenAI. The dataset is divided into 7473 training problems and 1319 test problems, with solutions presented in natural language format containing step-by-step reasoning processes.

HumanEval (Chen et al., 2021): Created by OpenAI, this dataset consists of 164 hand-written programming problems designed to evaluate code generation capabilities in LLMs. Each problem includes a function signature, docstring, and associated test cases, requiring models to generate Python code that satisfies the given specifications.

C Models

To comprehensively evaluate the mathematical reasoning capabilities, we selected four representative open-source large language models that span a range of model sizes and architectural characteristics. These models represent the state-of-the-art in efficient mathematical reasoning and instruction following capabilities.

Hunyuan-1.8B-Instruct (Cao et al., 2025): A lightweight instruction-tuned large language model

Dataset	Domain	Answer Format	# Samples	License
		Mathematical Reasoning		
MATH-500	Diverse Mathematical Fields	Free-form	500	CC BY-SA 4.0
AIME 2024	Competition Math	Integer (0-999)	60	CC BY-NC-SA 4.0
HMMT	Competition Math	Free-form	60	CC BY-NC-SA 4.0
GSM8K	Grade School Math	Free-form (Number)	8,500	MIT License
		Scientific Reasoning		
GPQA-Diamond	Graduate Science	Multi-Choice (4)	198	MIT License
		Code Generation		
HumanEval	Programming Problems (Python)	Code Implementation	164	MIT License

Table 3: An overview of the datasets in our benchmark.

Configuration	Hunyuan-0.5B-Instruct (Tencent, 2025)	Qwen3-0.6B (Qwen Team, 2024)	Qwen3-1.7B (Qwen Team, 2024)	DeepSeek-R1-Distilled-Qwen-1.5B (DeepSeek, 2024)
Parameters (B)	0.512	0.6	1.7	1.5
Hidden Size	1,024	1,280	1,536	1,536
Layers	24	24	24	24
Attention Heads	16	20	24	24
KV Heads	2	5	6	6
FFN Hidden Size	2,732	3,840	5,504	5,504
Max Sequence Length	256K	8,192	8,192	8,192
Vocabulary Size	152,064	151,936	151,936	151,936
Attention Type	GQA	GQA	GQA	GQA
Precision	BF16/FP16	BF16/FP16	BF16/FP16	BF16/FP16

Table 4: Overview of the large language models evaluated in our study.

developed by Tencent, featuring 1.8 billion parameters. It implements Grouped-Query Attention (GQA) for efficient inference and natively supports an extensive 256K context window for long-text tasks. The model is optimized for versatile deployment across edge devices and high-concurrency servers, delivering competitive performance in mathematical reasoning, code generation, and agent capabilities

Qwen3-0.6B (Yang et al., 2025): Smallest model in the Qwen3 series, featuring 0.6 billion parameters. Despite its compact size, it maintains strong performance across a variety of language understanding and generation tasks. This model serves as an important baseline for evaluating the minimal computational requirements for mathematical reasoning, making it particularly relevant for resource-constrained applications.

Qwen3-1.7B (Yang et al., 2025): A mid-sized model in the Qwen3 family that offers enhanced reasoning capabilities compared to its smaller counterpart. With 1.7 billion parameters, it provides a good balance between performance and computational cost, demonstrating improved instruction following and more robust mathematical reasoning compared to the 0.6B variant.

DS-R1-Distilled-Qwen-1.5B (DeepSeek-AI, 2025): A distilled version of DeepSeek's reasoning model, built upon the Qwen architecture. This model represents knowledge distillation techniques applied to reasoning capabilities, compressing the reasoning proficiency of larger models into a more computationally efficient 1.5B parameter package. It specializes in step-by-step reasoning processes while maintaining competitive performance on mathematical benchmarks.

Task: Compute the continuation tendency score p_i for the provided question-answer pair.

Step 1 – Segment the Answer: Decompose the model's answer into consecutive, semantically coherent segments. Each segment should represent a distinct reasoning step. Segments typically correspond to: a premise or fact extraction; a logical deduction or inference; a calculation or comparison; a conclusion or synthesis. Segments should be natural breaks in the reasoning flow. Number the segments sequentially as $(s_1, s_2, \ldots, s_{S_i})$.

Step 2 – Classify Adjacent Relations: For each adjacent pair of segments (s_j, s_{j+1}), classify the relationship as either:

- **Continuation:** The subsequent segment follows naturally from the previous one, for example:
  - Entailment: s_{j+1} is logically implied by s_j;
  - Elaboration: s_{j+1} provides details or examples for s_j;
  - Specification: s_{j+1} makes s_j more concrete;
  - Consequence: s_{j+1} states a consequence of s_j.
- **Transition:** The subsequent segment represents a shift, for example:
  - Reflection: Rethinking or reconsidering previous points;
  - Correction: Fixing an error or misconception;
  - Alternative exploration: Considering different approaches;
  - Meta-reasoning: Commenting on the reasoning process itself;
  - Hesitation: Phrases like "wait", "actually", "alternatively", "on second thought".

Step 3 – Compute continuation tendency score p_i: For a segmentation with S_i segments, there are $S_i - 1$ adjacent pairs. The continuation indicator function $\mathbb{I}_{\text{cont}}(s_j, s_{j+1})$ returns 1 if the relationship is CONTINUATION, and 0 if the relationship is TRANSITION. Then compute the sample-level continuation score as:

$$p_i = \frac{1}{S_i - 1} \sum_{j=1}^{S_i - 1} \mathbb{I}_{\text{cont}}(s_j, s_{j+1}). \tag{13}$$

Table 5: Prompt for computing the continuation tendency score p_i.