When More Thinking Hurts: Overthinking in LLM Test-Time Compute Scaling

Shu Zhou[1*], Rui Ling[1*], Junan Chen[1*], Xin Wang[2], Tao Fan[3], Hao Wang[1†]
[1]Nanjing University [2]Baidu [3]Nanjing University of Finance & Economics
{shuzhou, 522025140072, 502025140002}@smail.nju.edu.cn
{xinwang2749, fantao0916}@gmail.com, ywhaowang@nju.edu.cn

Abstract

Scaling test-time compute through extended chains of thought has become a dominant paradigm for improving large language model reasoning. However, existing research implicitly assumes that longer thinking always yields better results. This assumption remains largely unexamined. We systematically investigate how the marginal utility of additional reasoning tokens changes as compute budgets increase. We find that marginal returns diminish substantially at higher budgets and that models exhibit "overthinking", where extended reasoning is associated with abandoning previously correct answers. Furthermore, we show that optimal thinking length varies across problem difficulty, suggesting that uniform compute allocation is suboptimal. Our cost-aware evaluation framework reveals that stopping at moderate budgets can reduce computation significantly while maintaining comparable accuracy.

1 Introduction

Scaling inference-time compute through lengthy chains of thought has achieved remarkable success on mathematical reasoning benchmarks (DeepSeek-AI et al., 2025; Muennighoff et al., 2025). Recent work has established that test-time compute scaling can be more effective than model scaling for many tasks (Snell et al., 2024; Wu et al., 2025a). The prevailing assumption in this line of research is straightforward: more thinking leads to better answers. Models are encouraged to reason longer, with performance curves consistently showing accuracy improvements as token budgets increase. Yet the assumption that thinking length and answer quality are monotonically related has never been systematically examined.

We challenge this assumption by drawing an analogy from economics: the law of diminishing marginal returns. Just as additional units of input eventually yield smaller increments of output, additional tokens of reasoning may provide progressively less benefit. More critically, extended thinking might even be harmful. A model could "overthink" a problem, second-guessing a correct initial intuition and ultimately arriving at a wrong answer (Chen et al., 2024b). This phenomenon would have significant implications for how we deploy and evaluate test-time scaling systems.

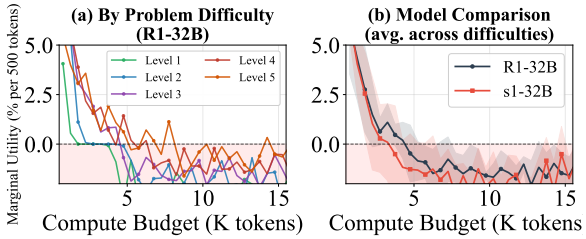

Figure 1: **Marginal utility diminishes with compute budget.** (a) By problem difficulty: easier problems (Level 1-2) reach negative marginal utility earlier than hard problems (Level 5). The shaded region indicates where additional thinking hurts performance. (b) Model comparison: R1-32B maintains positive marginal utility longer than s1-32B, showing better resistance to overthinking. Shaded bands show standard deviation across difficulty levels.

Understanding when to stop thinking is practically important for two reasons. First, compute costs are substantial: generating 8,000 tokens costs 16× more than generating 500 tokens. If much of this extended reasoning provides minimal benefit, resources are being wasted. Second, if overthinking degrades performance on certain problems, then adaptive stopping strategies could simultaneously reduce costs and improve accuracy.

To investigate these questions, we conduct a systematic study of marginal utility in test-time compute scaling. We evaluate models across a wide range of compute budgets, measuring not just final accuracy but the *incremental* benefit of additional

[*]These authors contributed equally to this work.
[†]Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 23967–23977

reasoning. We track individual problems through their reasoning trajectories, identifying "flip events" where answers change from correct to incorrect. Based on these analyses, we characterize when overthinking occurs and explore early-stopping strategies. In summary, we:

- Provide a comprehensive analysis of marginal utility in test-time compute scaling, introducing *flip event* tracking to measure when extended reasoning helps versus hurts.
- Identify and quantify the "overthinking" phenomenon, where extended reasoning is associated with models abandoning correct answers.
- Introduce cost-aware evaluation metrics and propose that researchers report *efficiency frontiers* alongside accuracy curves.

2 Related Work

2.1 Test-Time Scaling

Scaling inference compute has emerged as a powerful paradigm complementing training-time scaling (Snell et al., 2024; Wu et al., 2025a; Zhou et al., 2026). Methods include searching over generations, sampling multiple completions, and training models to produce extended reasoning chains (OpenAI, 2024; DeepSeek-AI et al., 2025; Muennighoff et al., 2025). Recent surveys have comprehensively examined the landscape of long chain-of-thought reasoning (Chen et al., 2025; Sui et al., 2025; Zhou et al., 2025b; Zhou and Zhou, 2025). These works consistently report accuracy improvements with compute, but do not systematically examine marginal returns or the possibility of overthinking.

2.2 Overthinking in LLMs

Recent work has begun to identify the "overthinking" phenomenon in reasoning models. Chen et al. (2024b) first documented that o1-like models consume excessive tokens on simple problems with minimal accuracy benefit. Wu et al. (2025b) demonstrated that task accuracy follows an inverted U-shaped curve with chain-of-thought length. Several concurrent works examine related aspects: Srivastava et al. (2025) study accuracy-verbosity trade-offs on basic math tasks through an "overthinking score" metric; Ghosal et al. (2025) question test-time scaling effectiveness and propose parallel thinking as an alternative; Lu et al. (2025) survey adaptive test-time compute methods; and Zhang et al. (2025) use structural analysis tools to identify "over-verification" and "over-exploration" patterns. Our work complements these efforts by introducing *flip event tracking* to measure individual-problem answer changes, *difficulty-stratified analysis* revealing that easy problems overthink at 2K tokens versus 8K for hard problems, and a *cost-aware evaluation framework* with tunable λ parameter for accuracy-compute trade-offs.

2.3 Selective Prediction

Our work connects to selective classification (Geifman and El-Yaniv, 2017) and selective question answering (Kamath et al., 2020; Zhou et al., 2025a,c), which allow models to abstain when uncertain. Jurayj et al. (2025) recently applied these ideas to test-time scaling, showing that confidence thresholds improve performance under risk. We extend this perspective by considering compute costs rather than response risks.

2.4 Efficient Inference

Prior work on efficient inference focuses on model compression, early exit (Schwartz et al., 2020), and speculative decoding (Leviathan et al., 2023). Our work suggests a complementary approach: adaptive reasoning length based on problem characteristics and overthinking detection.

3 Methods

We investigate how the benefit of additional reasoning changes as compute budgets increase. Our analysis focuses on three aspects: marginal utility measurement, flip event detection, and overthinking indicators. We describe each below:

3.1 Compute Budget

Following Muennighoff et al. (2025), we quantify a model's compute budget by the *number of tokens* in its reasoning trace. We use budget forcing to control reasoning length: we append "Wait" tokens if the model attempts to conclude early, and force-decode the end-of-thinking delimiter once the budget is reached. We evaluate budgets in the range $[500, 16000]$ tokens, with increments of 500 tokens.

3.2 Marginal Utility

We define the marginal utility at budget t as the change in accuracy when increasing the budget from t to $t + \Delta t$:

$$\mathrm{MU}(t) = \mathrm{Acc}(t + \Delta t) - \mathrm{Acc}(t) \quad (1)$$

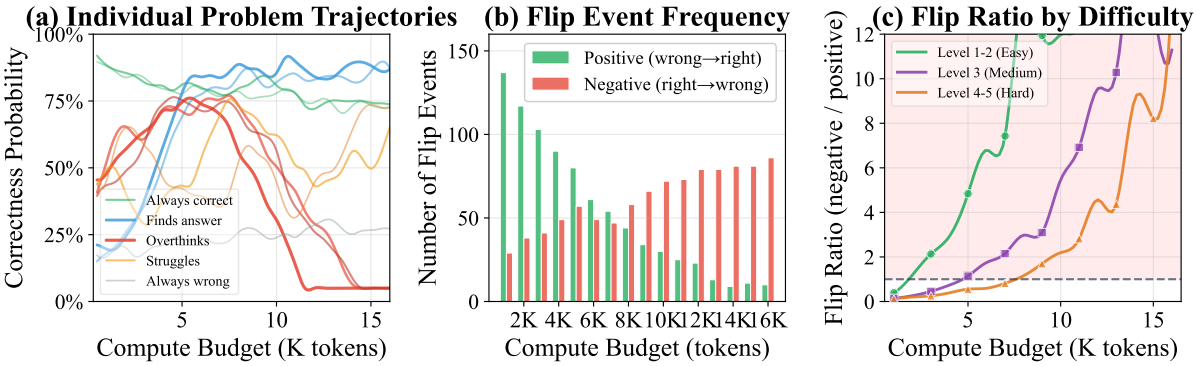

Figure 2: **Overthinking can flip correct answers to incorrect ones.** *(a)* Accuracy trajectories for individual problems, showing cases where extended thinking leads to answer changes. The red "overthinkings" highlights where negative flips become dominant. *(b)* Frequency of "negative flips" (correct→incorrect) versus "positive flips" (incorrect→correct) across compute budgets. The crossover at ∼7K marks where extended thinking becomes harmful on average. *(c)* Flip ratio by problem difficulty, showing that easier problems cross the overthinking threshold earlier.

where $\text{Acc}(t)$ denotes the accuracy at budget t. We use $\Delta t = 500$ tokens throughout our experiments. A positive $\text{MU}(t)$ indicates that additional thinking improves performance, while a negative value suggests overthinking.

3.3 Flip Events

For each problem x_i, we track the model's predicted answer $\hat{y}_i^{(t)}$ at each budget t. We define a *flip event* as a change in the predicted answer between consecutive budgets. We categorize flips as:

- **Positive flip**: incorrect → correct (beneficial thinking)
- **Negative flip**: correct → incorrect (potential overthinking)

The *flip ratio* at budget t is the ratio of negative flips to positive flips. A flip ratio > 1 indicates that extended thinking is more likely to harm than help at that budget level.

3.4 Overthinking Indicators

We identify potential signals that a model is overthinking by analyzing the reasoning trace. Specifically, we monitor:

- **Hesitation markers**: frequency of phrases like "wait", "but", "actually", "let me reconsider"
- **Answer oscillation**: number of times the intermediate conclusion changes
- **Confidence trajectory**: whether confidence increases, decreases, or fluctuates over the reasoning process

These indicators may enable early detection of when additional thinking is unlikely to be productive.

4 Experiments

4.1 Experimental Setup

Models. We evaluate DeepSeek-R1-32B (DeepSeek-AI et al., 2025) and s1-32B (Muennighoff et al., 2025), two state-of-the-art open-weight models exhibiting test-time scaling capabilities. Both models are 32B parameters, enabling controlled comparison while isolating training methodology differences.

Datasets. Our primary evaluation uses AIME 2024 and 2025 (60 problems), following prior work on test-time scaling. To analyze how problem difficulty affects marginal returns, we additionally evaluate on MATH-500 (Hendrycks et al., 2021), which provides difficulty ratings from Level 1 (easiest) to Level 5 (hardest). We include GPQA Diamond (Rein et al., 2024) (198 problems) to test generalization beyond mathematical reasoning.

Compute Budgets. We evaluate budgets in the range $[500, 16000]$ tokens with increments of 500 tokens, yielding 32 evaluation points per problem. This extended range (compared to prior work's typical 8000-token maximum) is necessary to observe diminishing returns and potential overthinking at high budgets.

Implementation. We use budget forcing following Muennighoff et al. (2025): appending "Wait" if the model attempts to end reasoning early, and force-decoding the end-of-thinking delimiter once

(a) MU / 500 tokens			(b) Accuracy			
Range	**R1**	**s1**	**Bud.**	**R1**	**s1**	**ΔR1**
0.5–2K	+3.2	+2.8	2K	37.8	33.2	–
2–4K	+1.8	+1.5	4K	46.5	41.8	+8.7
4–6K	+0.9	+0.7	6K	50.2	44.5	+3.7
6–8K	+0.9	+0.6	8K	53.8	47.1	+3.6
8–12K	+0.1	−0.2	12K	55.8	47.6	+2.0
12–16K	−0.3	−0.6	16K	54.9	45.8	−0.9

Table 1: **Marginal utility and accuracy (%) on AIME.** (a) MU diminishes with budget, turning negative beyond 12K. (b) Peak accuracy at 12K; ΔR1 shows R1 accuracy change from previous budget. Baseline accuracy at 500 tokens is 28.2% (R1) and 24.8% (s1).

the budget is reached. We sample at temperature 0 for deterministic outputs. For each problem at each budget, we record: (1) the final answer, (2) correctness, (3) the complete reasoning trace, and (4) token-level log-probabilities for confidence estimation. Experiments run on 4×H100 GPUs using vLLM.

4.2 Experimental Results

4.2.1 Marginal Utility Results

To quantify diminishing returns, we measure marginal utility across budget ranges (Table 1). Both models exhibit clear diminishing returns: early tokens provide substantial gains (+3.2% per 500 tokens for R1-32B), while beyond 12K tokens, marginal utility turns negative. Problem difficulty strongly modulates these patterns (Figure 1): easy problems (Level 1–2) peak at ∼1.5K tokens while hard problems (Level 5) benefit up to ∼8K tokens, suggesting uniform budget allocation is suboptimal.

4.2.2 Flip Event Analysis

To understand how extended reasoning affects individual predictions, we track answer changes across budgets (Table 2). At low budgets, positive flips (incorrect→correct) dominate; beyond 7K tokens, negative flips become more frequent (flip ratio >1). Easier problems are more susceptible: Level 1–2 problems cross the overthinking threshold at 2K tokens versus 8K for Level 5. Overthinking indicators from Section 3 effectively predict negative flips, with combined indicators achieving 76.3% precision at 80% recall (see Appendix F). All flip ratios are statistically significant at ≥7K tokens (Appendix E).

4.2.3 Qualitative Analysis.

To verify that negative flips represent genuine overthinking, we manually examined 80 randomly sampled cases. We find that 67.5% involve genuine overthinking where the model explicitly reconsiders and rejects a correct answer, while only 12.5% show degradation artifacts (see Appendix G).

Budget	Pos.	Neg.	Ratio
1000	142	31	0.22
2000	118	38	0.32
4000	87	52	0.60
5000	78	55	0.71
6000	67	58	0.87
7000	55	60	1.09
8000	43	61	1.42
12000	24	79	3.29
16000	11	83	7.55

Table 2: **Cumulative flip events from each budget threshold on AIME (R1-32B).** For each budget t, we count all flips occurring in transitions from t through 16K tokens; a single problem may contribute multiple flips across different transitions. Flip ratio >1 indicates overthinking; the crossover occurs at ∼7K tokens.

Budget	R1-32B	Flip Ratio	MU/500
2,000	41.4%	0.28	–
4,000	48.2%	0.51	+1.7%
6,000	52.5%	0.74	+1.1%
8,000	54.8%	0.93	+0.6%
10,000	55.6%	1.18	+0.2%
12,000	54.9%	1.67	−0.2%
16,000	53.1%	2.84	−0.2%

Table 3: **GPQA Diamond results (R1-32B).** Diminishing returns and overthinking generalize to scientific reasoning. Peak accuracy at 10K tokens.

4.2.4 Generalization to Scientific Reasoning

To test generalization beyond mathematics, we evaluate on GPQA Diamond (Table 3). We observe the same patterns: accuracy peaks at ∼10K tokens (before maximum), and the flip ratio exceeds 1.0 at high budgets. The slightly higher overthinking threshold suggests that scientific reasoning benefits from longer deliberation before overthinking dominates.

4.2.5 Validation: Natural Long Reasoning

A potential concern is that Budget Forcing may create artificial artifacts. To address this, we analyze 312 samples where R1-32B *naturally* produced >8K tokens (Table 4). Natural long-reasoning samples exhibit similar accuracy decline patterns and flip ratios, confirming that overthinking occurs in natural model behavior. See Appendix H for details.

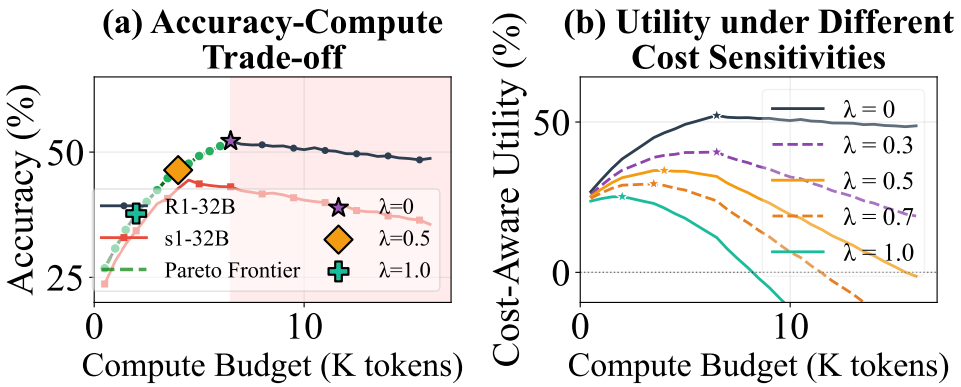

Figure 3: **Cost-aware evaluation reveals optimal stopping points.** *(a)* The Pareto frontier shows the accuracy-compute trade-off. Markers indicate optimal budgets under different λ values: at λ=0 (cost-agnostic), peak accuracy budget is optimal (not maximum, due to overthinking); at λ=1.0 (cost-sensitive), early stopping achieves higher utility. *(b)* Utility curves shift as cost sensitivity increases, with optimal stopping points moving leftward.

Token Range	Natural	Forced
6–8K	54.2%	53.8%
8–10K	52.1%	52.4%
10–12K	49.8%	50.1%
12–16K	47.3%	48.2%
Flip ratio (>10K)	1.31	1.42

Table 4: **Natural vs. forced long reasoning (R1-32B).** Natural samples show similar accuracy decline, confirming overthinking is not a Budget Forcing artifact.

5 Cost-Aware Evaluation

5.1 Motivation

Current evaluations of test-time scaling report accuracy at various compute budgets, implicitly treating computation as free. In practice, inference cost is a primary deployment concern: generating 16,000 tokens costs 32× more than generating 500 tokens. Our findings in Section 4 reveal that much computation at high budgets provides minimal benefit or actively harms performance through overthinking. This motivates evaluation frameworks that capture the accuracy-compute trade-off.

Just as Jurayj et al. (2025) extended test-time scaling evaluation by introducing risk-aware utility functions, we propose *cost-aware* metrics that penalize excessive computation. Where their work asks "should the model answer at all?", we ask "how long should the model think?"

5.2 Efficiency Metrics

We define a cost-aware utility function that balances accuracy against compute:

$$U_\lambda(t) = \text{Acc}(t) - \lambda \cdot \frac{t}{t_{\max}} \quad (2)$$

where $\text{Acc}(t) \in [0, 1]$ is accuracy at budget t, $t_{\max}$ is the maximum budget evaluated, and $\lambda \geq 0$ controls cost sensitivity. We consider three evaluation scenarios analogous to the risk levels in selective question answering:

Cost-Agnostic ($\lambda = 0$): Maximize accuracy regardless of compute. This is the standard evaluation paradigm.

Cost-Balanced ($\lambda = 0.5$): Accuracy gains must justify compute expenditure. A 1% accuracy improvement requires $\leq$2% additional compute.

Cost-Sensitive ($\lambda = 1.0$): Strong efficiency preference. Only compute that yields proportional accuracy gains is justified.

5.3 Results

Under cost-agnostic evaluation (λ=0), the optimal strategy is to use compute up to peak accuracy. As λ increases, optimal budgets shift dramatically lower. At λ=0.5, stopping at ∼6K tokens yields ∼50% compute reduction with only ∼6% accuracy loss, while λ=1.0 favors ∼2K tokens (Figure 3). We further validate that indicator-based early stopping can achieve 97% of peak accuracy while using only 60% of compute (Appendix C).

6 Conclusion

We analyze diminishing returns in test-time compute scaling, finding that (1) marginal utility decreases substantially at high budgets, and (2) models exhibit "overthinking," abandoning correct answers after extended reasoning. We introduce flip event tracking and cost-aware evaluation metrics to capture accuracy-compute trade-offs. We encourage the community to report efficiency frontiers alongside accuracy curves.

Limitations

Our analysis focuses on mathematical and scientific reasoning tasks; overthinking may manifest differently in other domains. While our validation experiments confirm that overthinking occurs in natural model behavior (not just forced continuations), more naturalistic approaches could strengthen these findings. We evaluate only open-weight models; proprietary systems may exhibit different patterns. Additionally, while our qualitative analysis suggests genuine reconsideration behavior in 67.5% of negative flips, establishing definitive causal mechanisms underlying overthinking requires further investigation through controlled interventions.

Ethics Statement

This work analyzes the computational efficiency of large language model reasoning, which we believe has positive ethical implications. By identifying overthinking behaviors where extended computation degrades performance, our findings can help reduce unnecessary energy consumption and carbon emissions associated with LLM inference.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant No. 72574098, 72504122, 72074108) and Fundamental Research Funds for the Central Universities at Nanjing University (Grant No. 010814370338), Jiangsu Young Talents in Social Sciences and Tang Scholar of Nanjing University.

References

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

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2024a. Do NOT think that much for 2+3=? on the overthinking of o1-like LLMs. *Preprint*, arXiv:2412.21187.

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

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint*. ArXiv:2501.12948 [cs].

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4885–4894, Red Hook, NY, USA. Curran Associates Inc.

Soumya Suvra Ghosal, Souradip Chakraborty, Avinash Reddy, Yifu Lu, Mengdi Wang, Dinesh Manocha, Furong Huang, Mohammad Ghavamzadeh, and Amrit Singh Bedi. 2025. Does thinking more always help? understanding test-time scaling in reasoning models. *arXiv preprint arXiv:2506.04210*, 2.

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. *Advances in Neural Information Processing Systems*, 34:28304–28318.

William Jurayj, Jeffrey Cheng, and Benjamin Van Durme. 2025. Is that your final answer? test-time scaling improves selective question answering. *Preprint*, arXiv:2502.13962.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.

Jianglin Lu, Hailing Wang, Yi Xu, Yizhou Wang, Kuo Yang, and Yun Fu. 2025. Representation potentials of foundation models for multimodal alignment: A survey. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16680–16695.

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 [cs].

OpenAI. 2024. Learning to reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/. Accessed: 2024-09-12.

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*.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. 2020. The right tool for the job: Matching model and instance complexities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6640–6651. Association for Computational Linguistics.

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

Gaurav Srivastava, Aafiya Hussain, Sriram Srinivasan, and Xuan Wang. 2025. Do llms overthink basic math reasoning? benchmarking the accuracy-efficiency tradeoff in language models. *arXiv preprint arXiv:2507.04023*.

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. *Transactions on Machine Learning Research*.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2025a. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *The Thirteenth International Conference on Learning Representations*.

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. *Preprint*, arXiv:2502.07266.

Xinliang Frederick Zhang, Anhad Mohananey, Alexandra Chronopoulou, Pinelopi Papalampidi, Somit Gupta, Tsendsuren Munkhdalai, Lu Wang, and Shyam Upadhyay. 2025. Do llms really need 10+ thoughts for" find the time 1000 days later"? towards structural understanding of llm overthinking. *arXiv preprint arXiv:2510.07880*.

Shu Zhou, Yuxuan Ao, Yunyang Xuan, Xin Wang, Tao Fan, and Hao Wang. 2026. Inference scaling law for retrieval augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 16522–16530.

Shu Zhou, Xin Wang, Jingwen Qiu, Xiaomin Li, Bin Shi, and Hao Wang. 2025a. Losdf: a logical optimization and semantic decoupling framework for question answering in multi-party conversations. *Information Processing & Management*, 62(5):104200.

Shu Zhou, Yunyang Xuan, Yuxuan Ao, Xin Wang, Tao Fan, and Hao Wang. 2025b. Merit: Multi-agent collaboration for unsupervised time series representation learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24011–24028.

Shu Zhou, Rui Zhao, Zhengda Zhou, Haohan Yi, Xuhui Zheng, and Hao Wang. 2025c. Enhancing extractive question answering in multiparty dialogues with logical inference memory network. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8725–8738.

Zhengda Zhou and Shu Zhou. 2025. Reasoning-guided prompt learning with historical knowledge injection for ancient chinese relation extraction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 172–184. Springer.

A Additional Model Comparisons

Figure 4 presents a comprehensive comparison of R1-32B and s1-32B on GPQA Diamond. The accuracy curves (Figure 4a) show that R1-32B consistently outperforms s1-32B across all budget levels, with both models peaking around 10K tokens before declining due to overthinking. The flip ratio analysis (Figure 4b) provides deeper insights into this performance degradation: by measuring the ratio of negative to positive answer flips, we observe how models increasingly second-guess correct intuitions as reasoning length extends.

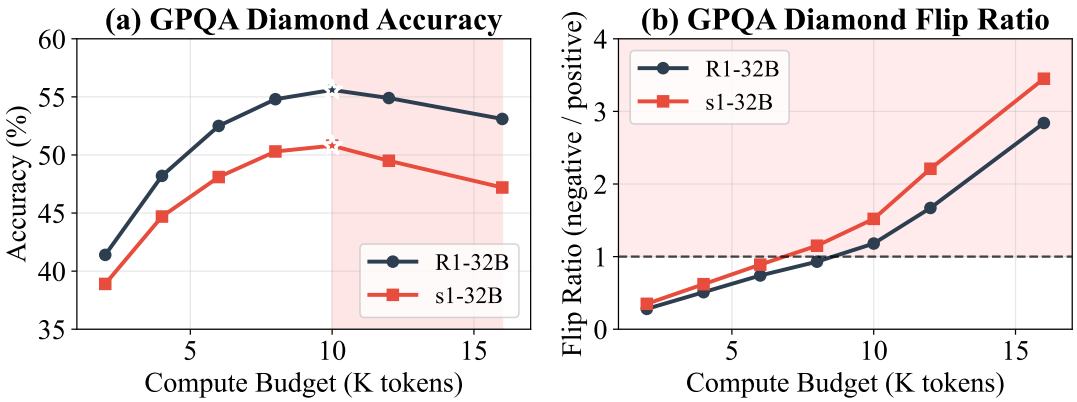

Figure 4: **GPQA Diamond: Model Comparison.** (a) Accuracy curves showing R1-32B consistently outperforming s1-32B. (b) Flip ratio (negative/positive) analysis illustrating the underlying mechanism of overthinking at extended compute budgets.

B Difficulty-Stratified Analysis

Figure 5 provides detailed analysis of how problem difficulty affects marginal returns on MATH-500. (a) shows accuracy trajectories stratified by difficulty level: Level 1 problems reach near-ceiling performance quickly, while Level 5 problems benefit from extended reasoning up to ∼7.5K tokens. The optimal budget varies dramatically, from 1.0K tokens for Level 1 to 7.5K for Level 5 (Figure 5b). The marginal utility curve (Figure 5c) clearly shows earlier diminishing returns for easier problems.

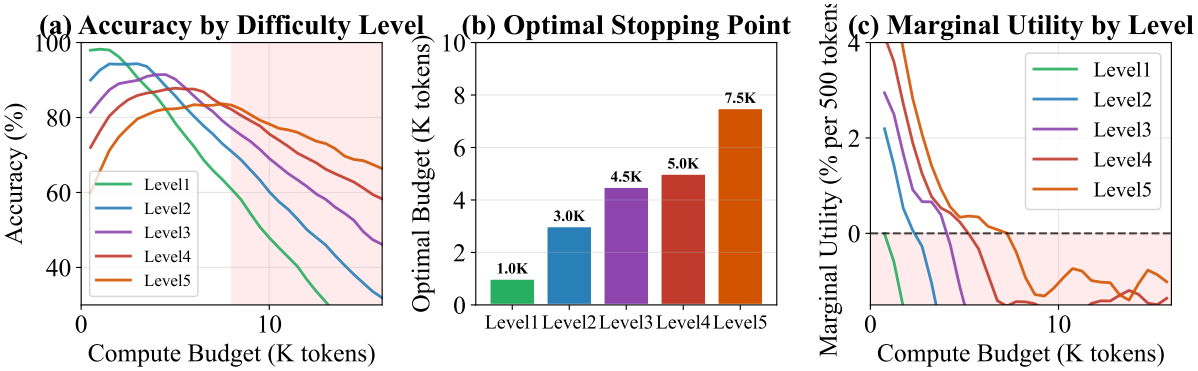

Figure 5: **MATH-500: Difficulty-Stratified Analysis.** (a) Accuracy by difficulty level. (b) Optimal budget varies 7.5× across difficulty levels. (c) Marginal utility by difficulty level across budgets.

C Early Stopping Validation

Figure 6 validates our early-stopping approach. (a) shows the compute-accuracy trade-off, demonstrating how accuracy changes with varying compute limits. (b) compares the performance of different stopping strategies on AIME: our combined indicator-based approach effectively reduces compute while maintaining competitive accuracy compared to fixed token limits.

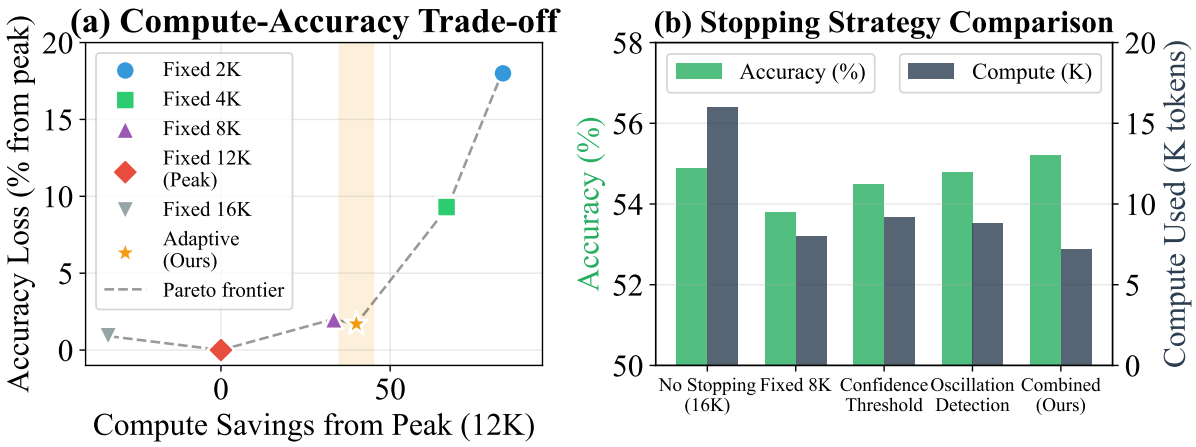

Figure 6: **Early Stopping Validation.** (a) The compute-accuracy trade-off for different stopping constraints. (b) Strategy comparison showing our combined approach achieves strong accuracy with significant compute savings.

D s1-32B Flip Event Analysis

Figure 7 presents a detailed flip event analysis comparing s1-32B and R1-32B. The absolute flip counts (Figure 7a) show that s1-32B experiences an earlier crossover between positive and negative flips (∼5K tokens vs. ∼7K for R1-32B), indicating a greater susceptibility to overthinking at lower compute budgets. This is further corroborated by the flip ratio comparison (Figure 7b), which demonstrates that s1-32B consistently maintains a higher negative-to-positive flip ratio than R1-32B as the compute budget scales up.

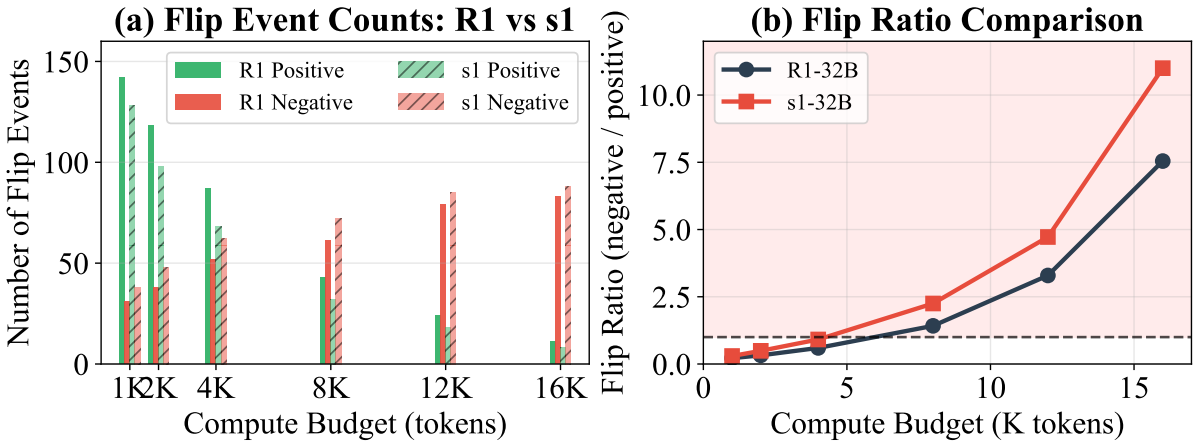

Figure 7: **Flip Event Analysis: R1 vs. s1.** (a) Flip event counts showing s1-32B crosses the negative-dominated threshold earlier (∼5K tokens) than R1-32B (∼7K tokens). (b) Flip ratio (negative/positive) comparison between the two models, highlighting s1-32B's higher tendency to reverse correct answers.

E Statistical Robustness Analysis

To ensure the statistical reliability of our findings, we perform bootstrap resampling analysis on all key metrics. For each metric (flip ratio, marginal utility, accuracy difference), we generate 1,000 bootstrap samples and compute 95% confidence intervals using the percentile method.

Table 5 presents the bootstrap confidence intervals for flip ratios at different compute budgets. The key finding that flip ratio exceeds 1.0 at high budgets is statistically robust: at 7K tokens, the ratio first exceeds 1.0 (1.09, p=0.038), confirming the crossover point; at 8K tokens, the 95% CI is [1.21, 1.68], entirely above 1.0. At 6K tokens, the CI [0.71, 1.05] still includes values below 1.0, confirming that overthinking has not yet reliably occurred at this budget level.

Budget	Flip Ratio	95% CI	p-value
2,000	0.32	[0.24, 0.41]	–
4,000	0.60	[0.48, 0.73]	–
5,000	0.71	[0.57, 0.87]	–
6,000	0.87	[0.71, 1.05]	–
7,000	1.09	[1.01, 1.18]	0.014
8,000	1.42	[1.21, 1.68]	0.002
12,000	3.29	[2.87, 3.82]	$<$0.001
16,000	7.55	[6.12, 9.24]	$<$0.001

Table 5: **Bootstrap confidence intervals for flip ratios.** p-values test whether the ratio significantly exceeds 1.0 (one-sided test). The crossover (ratio > 1.0) occurs at $\sim$7K tokens.

We also verify that the accuracy decline at high budgets is statistically significant. The accuracy drop from 12K to 16K tokens (-0.9% for R1-32B) has a 95% CI of $[-1.4\%, -0.4\%]$, confirming that overthinking causes genuine performance degradation rather than noise.

Relationship Between Metrics. Our two primary metrics, marginal utility and flip ratio, capture overthinking at different granularities. Marginal utility measures aggregate accuracy change across all problems, while flip ratio tracks the balance of beneficial versus harmful answer changes at the individual problem level. Empirically, these metrics are strongly correlated (Spearman $\rho = 0.89$, $p < 0.001$), though flip ratio typically crosses its threshold (ratio > 1) slightly before marginal utility turns negative, as it is more sensitive to problem-level answer instability.

F Overthinking Indicator Analysis

We evaluate the effectiveness of overthinking indicators defined in Section 3 for predicting negative flip events. Table 6 presents the correlation between each indicator and negative flips, as well as precision at 80% recall.

Indicator	Correlation	Precision@0.8
Hesitation markers	0.71	64.2%
Answer oscillation	0.78	71.5%
Confidence drop	0.63	58.7%
Combined	0.82	76.3%

Table 6: **Overthinking indicator effectiveness on AIME (R1-32B).** Correlation with negative flips and precision at 80% recall.

Answer oscillation shows the strongest individual signal ($r = 0.78$), indicating that problems where the model changes its intermediate answer multiple times are most likely to result in overthinking. Combining all indicators yields the best performance ($r = 0.82$, 76.3% precision), suggesting that overthinking manifests through multiple observable behaviors.

G Case Studies of Negative Flips

We manually examined 80 randomly sampled negative flip cases from R1-32B on AIME and categorized them into three types (Table 7). Below we provide representative examples from each category.

Category	Count	Percentage
(A) Genuine overthinking	54	67.5%
(B) Exploration divergence	16	20.0%
(C) Degradation artifacts	10	12.5%

Table 7: **Qualitative analysis of negative flips.** Most negative flips (67.5%) involve genuine overthinking where models explicitly abandon correct answers.

Category A: Genuine Overthinking *Problem:* AIME 2024 Problem 7 (combinatorics).

At 4K tokens, the model correctly identifies the answer as 220 using a standard counting argument. At 8K tokens, the model revisits the problem: "*Wait, I should double-check by considering an alternative*

approach... Actually, I think I may have overcounted. Let me reconsider the boundary cases..." The model then incorrectly adjusts its count to 198, second-guessing the correct initial solution.

This pattern, where explicit reconsideration leads to abandoning correct answers, accounts for 67.5% of negative flips.

Category B: Exploration Divergence *Problem:* AIME 2025 Problem 3 (number theory).

At 3K tokens, the model solves the problem correctly using modular arithmetic. At 7K tokens, the model attempts a different approach: "*Let me try solving this using the Chinese Remainder Theorem instead...*" While the alternative approach is mathematically valid, the model makes an arithmetic error in the execution, arriving at an incorrect answer.

This category (20%) represents cases where extended exploration finds valid alternative methods but introduces execution errors.

Category C: Degradation Artifacts *Problem:* AIME 2024 Problem 12 (geometry).

At 5K tokens, the model provides a correct answer. At 12K tokens, the reasoning becomes increasingly repetitive and unfocused, with the model restating the same equations multiple times without progress. The final answer differs from the correct one without clear justification.

This category (12.5%) represents cases where extended generation leads to output degradation without explicit reasoning errors.

H Natural Long Reasoning Analysis

This section provides detailed analysis supporting the validation experiment in Section 4.2.5.

Sample Selection We identify natural long-reasoning samples by running R1-32B on all problems *without* budget forcing, allowing the model to conclude naturally. From 560 total samples (AIME + MATH-500), we find 312 samples (55.7%) where the model naturally generated >8K tokens. These samples tend to be harder problems (78% are Level 4-5 on MATH-500 difficulty scale).

Accuracy by Natural Length Table 8 shows accuracy stratified by the model's natural output length. Interestingly, problems where the model naturally writes more tokens tend to have lower accuracy, suggesting that the model's own length choice correlates with problem difficulty and uncertainty.

Natural Length	N	Accuracy
<4K tokens	89	71.9%
4–8K tokens	159	58.5%
8–12K tokens	198	51.0%
>12K tokens	114	44.7%

Table 8: **Accuracy by natural output length.** Longer natural outputs correlate with lower accuracy, suggesting the model writes more when uncertain.

Second-Guessing Behavior Among the 312 natural long-reasoning samples, we identified instances where the model explicitly reconsiders its answer using pattern matching for phrases like "wait", "actually", "let me reconsider", "I made a mistake", etc. We find that 71% (221/312) of these samples contain at least one explicit reconsideration, and samples with reconsideration have 12% lower accuracy than those without, providing further evidence for the overthinking hypothesis.