REASONING WITHOUT SELF-DOUBT: MORE EFFICIENT CHAIN-OF-THOUGHT THROUGH CERTAINTY PROBING

Yichao Fu*
University of California, San Diego
yif034@ucsd.edu

Junda Chen*
University of California, San Diego
juc049@ucsd.edu

Yonghao Zhuang
Carnegie Mellon University
yhzhuang@cmu.edu

Zheyu Fu
University of California, San Diego
z7fu@ucsd.edu

Ion Stoica
University of California, Berkeley
istoica@cs.berkeley.edu

Hao Zhang
University of California, San Diego
haozhang@ucsd.edu

ABSTRACT

Recent large language models with chain-of-thought reasoning capabilities exhibit poor token efficiency due to self-doubt — spending excessive tokens verifying already-correct answers. Using our Probe-In-The-Middle technique to analyze model states during reasoning, we propose Dynasor-CoT, a certainty-based approach for dynamic reasoning termination. Our training-free method efficiently achieves up to 29% token reduction while maintaining accuracy across mathematical reasoning tasks like AMC23, AIME24, and MATH500.

1 INTRODUCTION

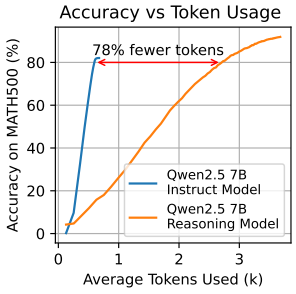

Figure 1: The token efficiency curve for the traditional model is much steeper than reasoning model.

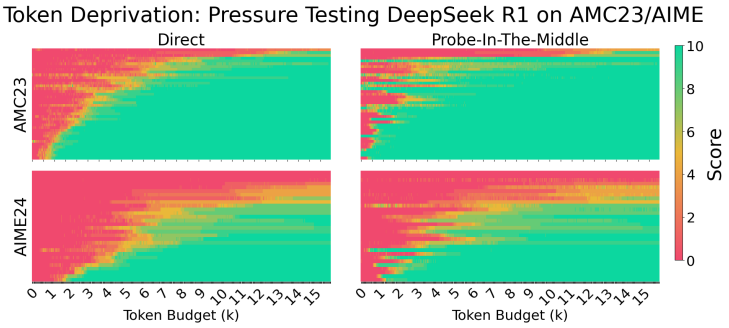

Figure 2: DeepSeek R1 performance on AMC23 (upper) and AIME24 (lower) at varying token budgets (scoring lowest to highest over 10 attempts). (Left) Standard reasoning with late answer outputs. (Right) Early answer extraction using Probe-In-The-Middle technique.

Recent advances in large language models (LLMs) with long Chain-of-Thought (CoT) reasoning capabilities, such as DeepSeek r1 (Guo et al., 2025) and OpenAI o1/o3 (Jaech et al., 2024; OpenAI, 2025), have demonstrated remarkable performance on complex tasks (e.g., math and code). However, compared to previous LLMs, these models exhibit markedly lower token efficiency—requiring more tokens to achieve comparable accuracy—as shown in Figure 1.

*Equal contribution.

One major source of this inefficiency stems from what we identify as self-doubt: models often reach the correct answer early but engage in extended verification behaviors such as double-checking, reassessment, reverification, and so on. Such self-doubt patterns can lead to significantly increased token consumption. For instance, Figure 3 compares traditional Qwen-7B (Yang et al., 2024) model with a reasoning Deepseek-distilled Qwen-7B model on a simple question. While the traditional model reaches its answer in 180 tokens, the reasoning model expends 1K tokens on iterative verification steps but already got the correct answer at token 340.

To systematically investigate this phenomenon, we developed a "Probe-In-The-Middle" technique (or "Probe" for short) that extracts the model's intermediate thinking by appending specific prompts such as *"Oh, I suddenly got the answer to the whole problem, Final Answer: boxed{"*. Figure 2 shows the analysis of the accuracy comparing directly asking vs probing the model. Taking AMC23 as an example, reasoning models frequently arrive at correct answers early (median: 830 tokens), but continue generating unnecessary tokens due to self-doubt (median: 2.7K tokens). This self-doubt phenomenon significantly impacts token efficiency, as models continue reasoning despite having internal confidence in their answers. Our key insight is that LLMs exhibit detectable levels of certainty during their reasoning process, which can be leveraged to determine effective stopping points.

To address self-doubt, we propose Dynasor-CoT, a novel approach for long CoT reasoning. Our method combines certainty-based heuristics with the probe-in-the-middle technique to dynamically determine termination points. This approach efficiently truncates reasoning chains while maintaining accuracy, demonstrating significant improvements over fixed-token-budget baselines. Notably, it achieves up to 29% token reduction without compromising accuracy or requiring additional training, and introduces no extra latency to the critical reasoning path.

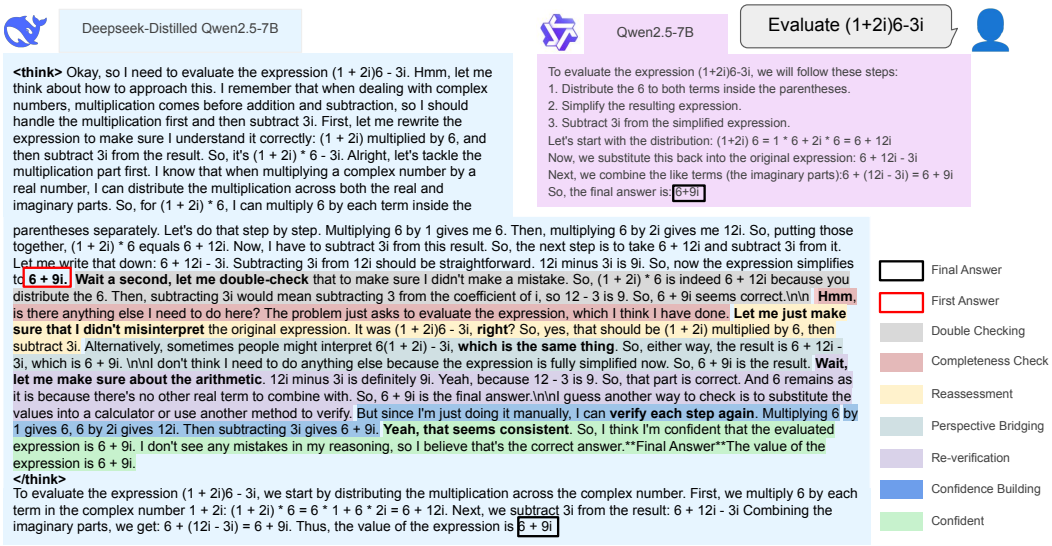

Figure 3: An Example of **Self-Doubt** Comparing a Reasoning Model (Deepseek-distilled Qwen-2.5 7B) vs. a Traditional Model (Qwen-2.5 7B) on a Problem from the MATH500 Dataset

2 Dynasor-CoT: Efficiently Scaling Long Chain-of-Thought Reasoning

We present an efficient reasoning framework Dynasor-CoT for early termination that enhances token-to-accuracy efficiency in long CoT LLM reasoning through three key mechanisms: answer extraction by probe, certainty assessment, and post-generation validation. Figure 4 shows an example of our methods.

2.1 Probe In The Middle

Instead of waiting for complete reasoning chains, we introduce strategic interventions called *Probe-In-The-Middle* (or probe in short) during the generation process. Our approach appends carefully designed guidance at intermediate stages of reasoning to explicitly elicit the model's current answer (e.g., *"Oh, I suddenly got the answer to the whole problem, Final Answer: boxed{"*). This method

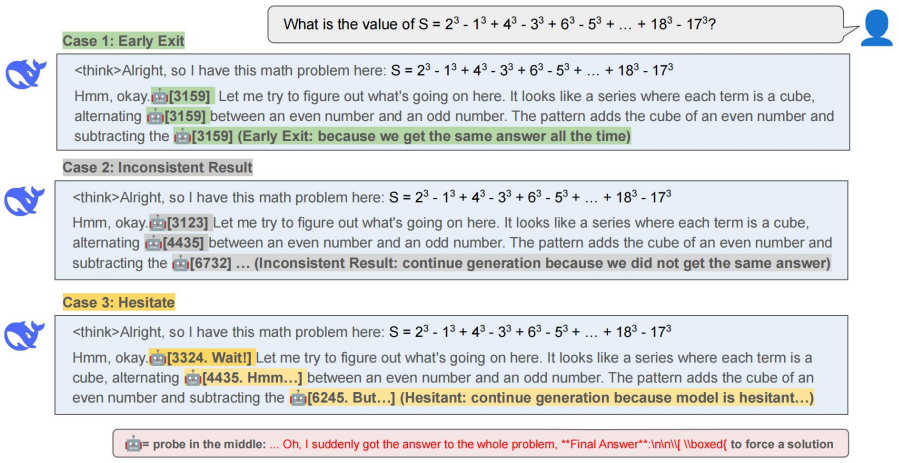

Figure 4: Illustration of Dynasor-CoT: (1) Probe-In-The-Middle for answer extraction, (2) early exit based on certainty (case 1), (2) post-generation validation for hesitation words (e.g., “wait”) (case 3), and (4) continue if not certain enough (case 2)

capitalizes on our observation that reasoning LLMs often reach the correct solution before completing their full reasoning chain. When the LLM has already reached its conclusion internally, this early extraction technique significantly reduces computational costs.

2.2 Certainty Assessment through Answer Consistency

We implement a dynamic certainty assessment mechanism that monitors the model’s outputs at regular intervals (e.g., every 32, 64, or 128 tokens). At each interval, we probe the model to extract and store the current answer, then allow the LLM to continue its generation. Importantly, the subsequent generation remains unaffected by the probing tokens, enabling parallel execution of answer extraction and original generation. When the model produces consistent answers across multiple intervals, we interpret this pattern as an indicator of certainty, following the certaindex approach (Fu et al., 2024). This methodology provides a quantitative measure of the model’s certainty.

2.3 Post-generation Validation

We empirically observed DeepSeek-R1 and DeepSeek-Distill models’ generations, and identified that they generate specific words like ”wait” or ”hmm” when lacking certainty in their previous generations. Based on this finding, we specifically monitor for these uncertainty indicators following probed answers. Responses containing these indicators are automatically discarded. This validation mechanism works in conjunction with the certainty assessment to create a comprehensive certainty metric. Figure 4 shows an example.

These three components operate synergistically to optimize the token-to-accuracy trade-off. At regular intervals, the framework injects probe words after the current generation to extract the model’s answer at that reasoning stage. It then discards answers that exhibit low certainty indicators. Finally, it terminates the process early if answers remain consistent across several consecutive intervals. This approach leverages the model’s ability to reach conclusions during intermediate stages while maintaining robust safeguards against premature or uncertain responses. Our method requires no additional training or model changing, making it readily applicable to existing LLM deployments.

3 EVALUATION

3.1 MAIN RESULTS

We evaluate our certainty-based early termination method Dynasor-CoT against baseline uniform token allocation across multiple scales of distilled DeepSeek models (7B, 14B, and 32B) (Guo et al., 2025) on mathematical reasoning benchmarks AIME24 and AMC23 (Yang et al., 2024), and MATH500 (Lightman et al., 2023). Unlike the baseline approach that uniformly increases token budgets, our method enables early termination by monitoring model certainty at various intervals. As illustrated in Figure 5, we evaluate variable probing intervals (32, 64, and so on) represented by distinct colored lines, with a maximum token budget of 16K. For each interval, we vary the early termination parameter N (the required number of consecutive consistent answers), generating different points along each line. All configurations achieve significant token savings, with our approach reducing token usage by up to 29% while maintaining comparable accuracy to the baseline. For fair comparison, appropriate accuracy thresholds were calibrated to model scale - with 32B models evaluated against stricter thresholds above QwQ (Team, 2024) levels and reduced thresholds for smaller models - while setting higher targets for simpler tasks where greater accuracy is achievable. For the 10% of problems where our method achieves the highest token reduction, we observe savings of 34% on AIME and 53% on MATH500. This extends further for the top 1% of problems, where we achieve even more substantial reductions of 53% on AIME and 81% on MATH500. These results, particularly the substantial token savings on certain problems (up to 81% reduction), demonstrate our method's ability to adapt token allocation to different problem types. This variable performance shows the advantage of our dynamic approach over fixed token budgets, as problems vary in their token requirements for reaching solutions.

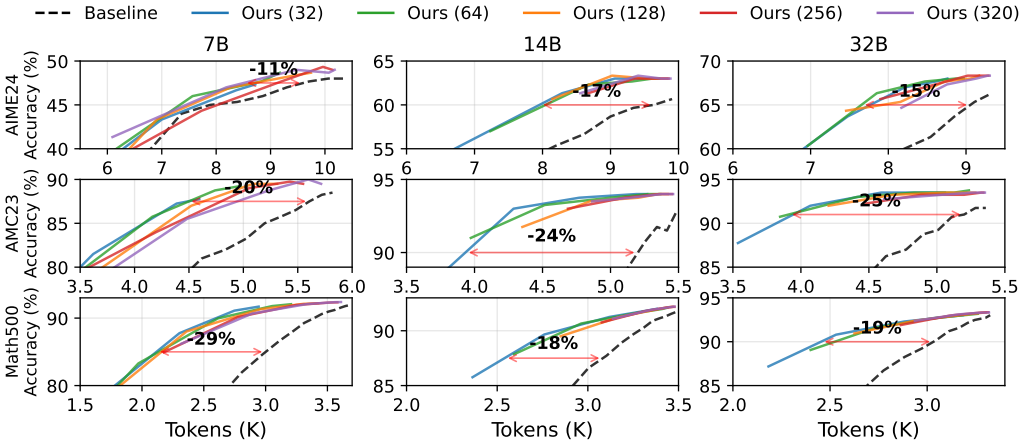

Figure 5: Comparing Dynasor-CoT Performance Across Model Scales and Datasets

To validate scalability, we extended our experiments to the larger DeepSeek-R1 model on AIME and AMC datasets (Figure 6). The results align with our findings from smaller distill models, demonstrating consistent efficiency gains: DeepSeek-R1 achieves 12% token savings on AIME problems and 24% on AMC problems while maintaining baseline accuracy levels.

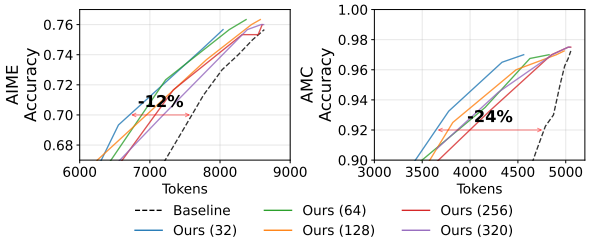

Figure 6: Apply Dynasor-CoT on DeepSeek-R1

3.2 ABLATION STUDY

We conduct ablation studies across MATH500, AIME24, and AMC23 using DeepSeek Distill 32B to evaluate our framework's components. Our analysis compares four configurations: (1) baseline (uniform token budget), (2) baseline + probing, (3) our certainty-based early exit without post-generation validation, and (4) our full Dynasor-CoT framework. Results (Figure 7) demonstrate

that both the basic probe implementation and the version without validation achieve lower token efficiency compared to our complete framework across all settings.

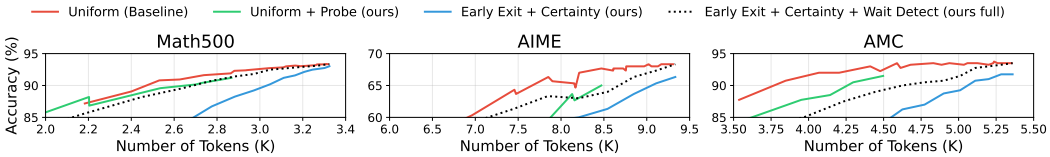

Figure 7: Effectiveness of Components using DeepSeek Distill 32B on mathematic datasets

4 Conclusion

In this paper, we present Dynasor-CoT, a technique that addresses self-doubt in long chain-of-thought reasoning to enhance LLMs' token efficiency. Our evaluation shows that Dynasor-CoT improves token-to-accuracy efficiency by up to 29% compared to baselines across various model configurations and mathematical reasoning tasks.

REFERENCES

Amey Agrawal, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav S Gulavani, and Ramachandran Ramjee. Sarathi: Efficient llm inference by piggybacking decodes with chunked prefills. *arXiv preprint arXiv:2308.16369*, 2023.

Amey Agrawal, Junda Chen, Íñigo Goiri, Ramachandran Ramjee, Chaojie Zhang, Alexey Tumanov, and Esha Choukse. Mnemosyne: Parallelization strategies for efficiently serving multi-million context length llm inference requests without approximations. *arXiv preprint arXiv:2409.17264*, 2024.

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

Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*, 2023.

Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Aurick Qiao, and Hao Zhang. Efficiently serving llm reasoning programs with certaindex. *arXiv preprint arXiv:2412.20993*, 2024.

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

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Cunchen Hu, Heyang Huang, Liangliang Xu, Xusheng Chen, Jiang Xu, Shuang Chen, Hao Feng, Chenxi Wang, Sa Wang, Yungang Bao, et al. Inference without interference: Disaggregate llm inference for mixed downstream workloads. *arXiv preprint arXiv:2401.11181*, 2024.

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

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Rohin Manvi, Anikait Singh, and Stefano Ermon. Adaptive inference-time compute: Llms can predict if they can do better, even mid-generation. *arXiv preprint arXiv:2410.02725*, 2024.

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

OpenAI. Openai o3-mini system card. January 2025. URL `https://cdn.openai.com/o3-mini-system-card.pdf`.

Pratyush Patel, Esha Choukse, Chaojie Zhang, Aashaka Shah, Íñigo Goiri, Saeed Maleki, and Ricardo Bianchini. Splitwise: Efficient generative llm inference using phase splitting. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pp. 118–132, 2024. doi: 10.1109/ISCA59077.2024.00019.

Ruoyu Qin, Zheming Li, Weiran He, Mingxing Zhang, Yongwei Wu, Weimin Zheng, and Xinran Xu. Mooncake: A kvcache-centric disaggregated architecture for llm serving. *arXiv preprint arXiv:2407.00079*, 2024.

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024. URL `https://qwenlm.github.io/blog/qwq-32b-preview/`.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. DistServe: Disaggregating prefill and decoding for goodput-optimized large language model serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 193–210, Santa Clara, CA, July 2024. USENIX Association. ISBN 978-1-939133-40-3. URL https://www.usenix.org/conference/osdi24/presentation/zhong-yinmin.

A RELATED WORK

Chain-of-thought reasoning has emerged as a powerful paradigm for tackling complex tasks such as math, programming, and creative writing, with models like Deepseek-R1 (Guo et al., 2025) and OpenAI O3 (OpenAI, 2025) employing extended reasoning chains to simulate human deliberation. However, this process can produce output sequences that are up to 20 times longer than direct responses (Chen et al., 2024) for simple questions, leading to substantial increases in inference costs and latency.

To mitigate these inefficiencies, several works (Manvi et al., 2024; Chen et al., 2024; Muennighoff et al., 2025) have explored methods to reduce token consumption during reasoning, including mid-generation self-evaluation and adaptive termination strategies. These approaches typically require fine-tuning, additional predictors, or estimations of question difficulty, which can complicate deployment despite their potential to balance accuracy with reduced computational overhead.

While LLM serving systems like Dynasor (Fu et al., 2024) use statistical signals (i.e., Certaindex) to dynamically allocate compute across parallel reasoning branches, they primarily focus on system-level optimization for complex reasoning algorithms (e.g., self-consistency (Wang et al., 2022) and MCTS (Hao et al., 2023; Feng et al., 2023)). In contrast, our work identifies and addresses the hesitation phenomenon in chain-of-thought reasoning for models like DeepSeek-R1, where reasoning is inherently sequential. We leverage probe-in-the-middle techniques and certainty-based heuristics (e.g., post-generation validation) to efficiently truncate reasoning chains when models exhibit sufficient confidence in their answers.

A.1 LIMITATION

Our work did not focus how Dynasor-CoT reduce latency into actual serving system, particularly with large batch size and long context. Future work can explore techniques such as disaggregation (Zhong et al., 2024; Patel et al., 2024; Hu et al., 2024; Qin et al., 2024), chunked prefill (Agrawal et al., 2023; 2024) on the latency-accuracy trade-off in serving systems and real workload.