SpecCoT: Accelerating Chain-of-Thought Reasoning through Speculative Exploration

Junhan Shi [1] **Yijia Zhu** [2] **Zhenning Shi** [1] **Dan Zhao** [3] **Qing Li** [3] **Yong Jiang** [4 3]

Abstract

Large Reasoning Models (LRMs) demonstrate strong performance on complex tasks through chain-of-thought (CoT) reasoning. However, they suffer from high inference latency due to lengthy reasoning chains and the substantial computational requirements of large models. In this paper, we propose SpecCoT, a collaborative framework that combines large and small models for effective yet efficient reasoning. Unlike traditional speculative decoding, which operates at the token level, SpecCoT adopts a step-level verification strategy: the large model first establishes the reasoning direction, and for each intermediate step, the small model generates multiple candidate drafts in parallel. The large model then verifies these drafts, either selecting the most suitable one or rejecting them all and generating its own response. This SpecCoT approach balances reasoning quality with inference efficiency through fine-grained model cooperation. Experiments across diverse tasks show that SpecCoT reduces inference latency by 1.7-4.1× while maintaining comparable accuracy to standard large model inference.

1. Introduction

Large Reasoning Models (LRMs) leveraging chain-of-thought (CoT) (Wei et al., 2022b) offer powerful reasoning but suffer from high inference latency, hindering their application in time-sensitive scenarios (Jaech et al., 2024; Team, 2025; DeepSeek-AI et al., 2025). Current acceleration methods, such as model distillation or CoT shortening (Wang et al., 2025b; Liu et al., 2024; Xia et al., 2025), face a critical trade-off: large models are inefficient for simple tasks, while smaller or compressed models fail on complex ones, motivating adaptive frameworks (Xia et al., 2025; Zhang et al., 2025; Liu et al., 2024).

Inspired by the success of speculative decoding (Leviathan et al., 2023; Chen et al., 2023; Xia et al., 2024) in model collaboration, we propose SpecCoT (Speculative Chain-of-Thought). This framework employs a small model to generate multiple candidate reasoning steps in parallel for each intermediate stage. A large model then performs step-wise verification of these drafts, intervening to generate its own reasoning step only when all candidates are deemed unacceptable. This strategic delegation enhances efficiency by reserving the large model primarily for validation and critical reasoning tasks, thereby preserving overall accuracy while reducing computational overhead.

Experimental evaluations demonstrate that SpecCoT reduces inference latency by up to 4.1× while maintaining comparable accuracy to standard approaches. The effectiveness of SpecCoT stems from its collaborative framework that combines parallel candidate generation with dynamic verification, where the large model intervenes only when necessary, effectively balancing reasoning quality with computational efficiency.

Our key contributions include:

- We propose SpecCoT, a collaborative reasoning framework that strategically delegates reasoning steps between large and small models to achieve an optimal balance between reasoning quality and computational efficiency.
- We develop a novel parallel verification mechanism with dynamic fallback, enabling efficient step-wise evaluation of multiple reasoning candidates while preserving accuracy through selective large model intervention.
- Comprehensive experiments across diverse reasoning benchmarks demonstrate that SpecCoT reduces inference latency by 1.7-4.1× while maintaining comparable accuracy to standard large model inference.

2. Motivation

Complex reasoning tasks present unique challenges and opportunities for models of different sizes. This section ex-

[1]Tsinghua University [2]Xidian University [3]Peng Cheng Laboratory [4]Tsinghua Shenzhen International Graduate School. Correspondence to: Yong Jiang <jiangy@sz.tsinghua.edu.cn>, Qing Li <liq@pcl.ac.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

plores their inherent trade-offs in reasoning versus efficiency and identifies key optimization opportunities.

2.1. Model Size vs. Reasoning Efficiency

Small models generate content faster but with reduced reasoning reliability. Small models offer substantial generation speed advantages (3-4× faster token generation), but this does not guarantee faster overall reasoning performance. Their limited reasoning capabilities result in verbose solutions (up to 4× more tokens, Table 1), yielding similar end-to-end latency compared to large models. This verbosity often indicates underlying reasoning deficiencies, such as incoherent logic or circular arguments.

Table 1: Performance and efficiency tradeoffs in DeepSeek R1 distilled Qwen models in GSM8K.

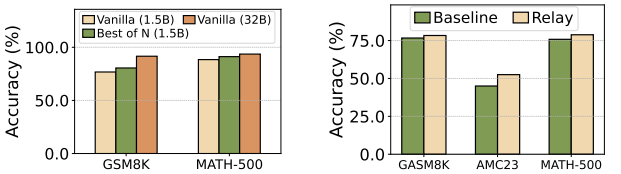
Distill-Qwen	1.5B	7B	14B	32B
Accuracy (%)	75.51	87.49	90.83	91.58
Avg Time (s)	10.94	5.20	6.86	10.33
Avg Token	2736.46	974.19	758.51	634.11
Speed (token/s)	256.38	200.41	120.39	67.90

Large models demonstrate superior reasoning capabilities with built-in reflection mechanisms. Large models, despite slower token generation rates, exhibit advanced reasoning abilities beyond computational scale. Their key advantage lies in intrinsic reflection capabilities—monitoring reasoning processes, detecting errors, and self-correcting—maintaining logical consistency. Empirically, large models achieve higher accuracy (91% vs 75%) while using fewer tokens (less than 1/4 compared to small models) to reach solutions.

2.2. Collaborative Reasoning Principles

Complex reasoning can be decomposed into sub-tasks of varying difficulty. Complex reasoning involves sub-tasks of varying difficulty. Pivotal stages like initial analysis or strategy formulation require sophisticated reasoning capabilities. However, many intermediate steps—calculations, logical deductions, case analyses—are more straightforward and can be effectively handled by small models despite their end-to-end limitations. This suggests opportunities for efficient model deployment through strategic task allocation.

Reasoning quality depends on logical correctness rather than exact expression. Unlike tasks requiring precise wording, reasoning quality hinges on logical correctness rather than exact expression. This differs fundamentally from speculative decoding, which demands precise token-level replication. Since multiple valid formulations can convey the same logical step, reasoning naturally accommodates collaboration—allowing small models to contribute effectively despite variations in their outputs.



((a)) Performance of Best-of-N on GSM8K and MATH-500.

((b)) Impact of initial guidance on reasoning accuracy.

Figure 1: Performance analysis of Best-of-N sampling and initial guidance impact on reasoning tasks.

2.3. Optimization Potential for Small Models

High-quality initial guidance impacts subsequent reasoning. High-quality initial guidance critically impacts subsequent reasoning. A well-structured initial analysis decomposes complex problems, establishes a clear framework, and crucially helps prevent error accumulation, as early mistakes often cascade. Leveraging more capable models at this initial stage offers substantial benefits for the entire solution.

Parallel candidate generation enables efficient exploration. Parallel candidate generation enables efficient exploration. While individual small model generations may lack reliability, generating multiple candidates in parallel significantly increases the chance of obtaining high-quality reasoning steps. Optimized inference frameworks like vLLM(Kwon et al., 2023) make this multi-candidate generation's overhead manageable, allowing parallel solution space exploration to overcome small model limitations.

3. Method

Speculative Chain-of-Thought (SpecCoT) is a collaborative framework leveraging the complementary strengths of large and small language models. It decomposes CoT reasoning into speculative iterations: large models provide initial guidance and verification, while small models generate multiple diverse intermediate reasoning steps. This approach maintains reasoning quality with efficiency gains from parallel candidate generation.

3.1. Formulation

Given problem x, SpecCoT generates a reasoning chain $C = \{c_1, ..., c_n\}$ for answer y. The target model generates initial guidance g to provide a strong foundation. For each subsequent step i, the draft model generates N candidate continuations $\{\hat{c}_i^j\}_{j=1}^N$ based on prior steps $c_{1:i-1}$. The target model then verifies and selects the best candidate or regenerates the step if needed. Formally:

$$x \xrightarrow{C} y, \quad \text{where } C = \{g\} \cup \{c_1, c_2, ..., c_n\}. \tag{1}$$

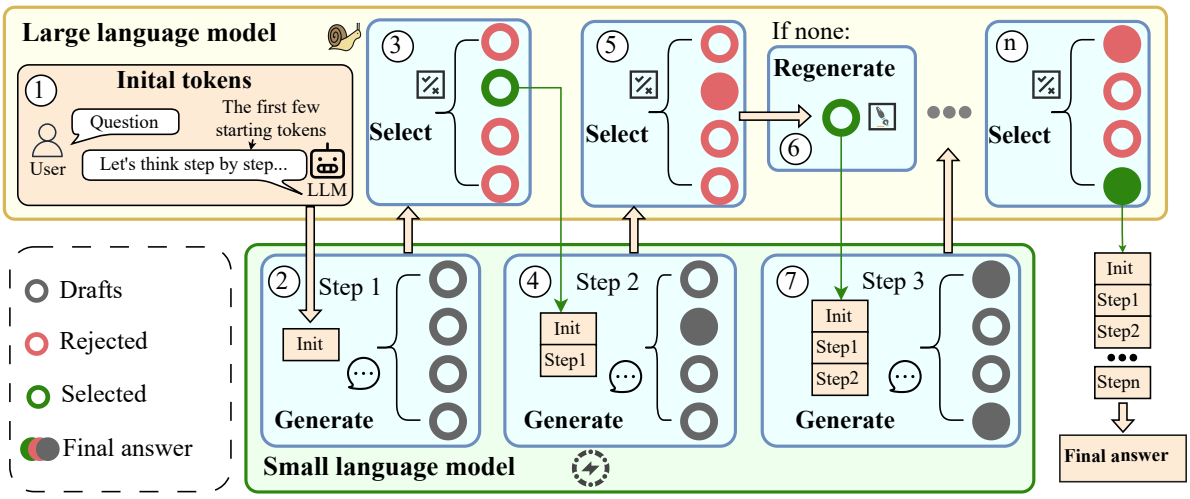


Figure 2: Overview of SpecCoT. The process begins with an LLM generating initial guidance tokens (1), followed by a small language model generating multiple candidate continuations for each reasoning step (2,4,7). For each step, the LLM verifies and selects the most promising candidate (3,5,8) or regenerates the content itself if no candidate meets quality standards (6). This collaborative approach combines the reasoning quality of large models with the computational efficiency of small models, culminating in a final answer after multiple reasoning steps.

This approach maintains quality through target model guidance and verification while reducing overhead via parallel candidate generation by the draft model.

3.2. Initial Guidance

SpecCoT starts with a large target model M_T generating an initial k-token guidance g from input x:

$$g \sim M_T(\cdot|x), \quad |g| = k \tag{2}$$

where g forms the reasoning chain foundation. A strong initial direction is crucial; M_T's superior reasoning establishes a sound starting point, constraining the draft model's solution space and minimizing overhead.

3.3. Speculative Exploration

After initial guidance, SpecCoT uses a lightweight draft model M_D for efficient exploration at each step i, generating N parallel candidate continuations simultaneously:

$$\hat{c}_i^j \sim M_D(\cdot|g, c_{1:i-1}), \quad j = 1, ..., N \tag{3}$$

This parallel generation mitigates M_D's limitations and leverages batch sampling efficiency. Diversity from multiple candidates increases the chance of finding a high-quality continuation.

3.4. Efficient Verification and Fallback

An efficient verification mechanism evaluates all candidates via a single target model forward pass. Augmenting candidates with a `reject` option allows M_T to select a candidate or signal the need for its own generation:

$$s = V_T(\{\hat{c}_i^j\}_{j=1}^N \cup \{\texttt{reject}\}), \tag{4}$$

where V_T represents the verification function based on M_T that assesses logical coherence and reasoning progress. This requires only one forward pass with a single token output.

The `reject` option provides key quality control. If all candidates are unsuitable (logical errors, insufficient progress), M_T rejects them and generates the step itself:

$$c_i \sim M_T(\cdot|g, c_{1:i-1}). \tag{5}$$

This verification and fallback approach ensures reasoning quality with minimal computational cost, balancing efficiency and reliability through selective intervention.

3.5. Relation to Speculative Decoding

SpecCoT shares conceptual similarities with speculative decoding, as both leverage draft models for efficiency gains. While speculative decoding operates at the token level, SpecCoT extends this concept to reasoning steps. This alignment suggests potential for a unified framework operating at different granularities—using the same draft model for both token-level and step-level speculation could enable hierarchical efficiency, with target model verification ensuring quality at both levels.

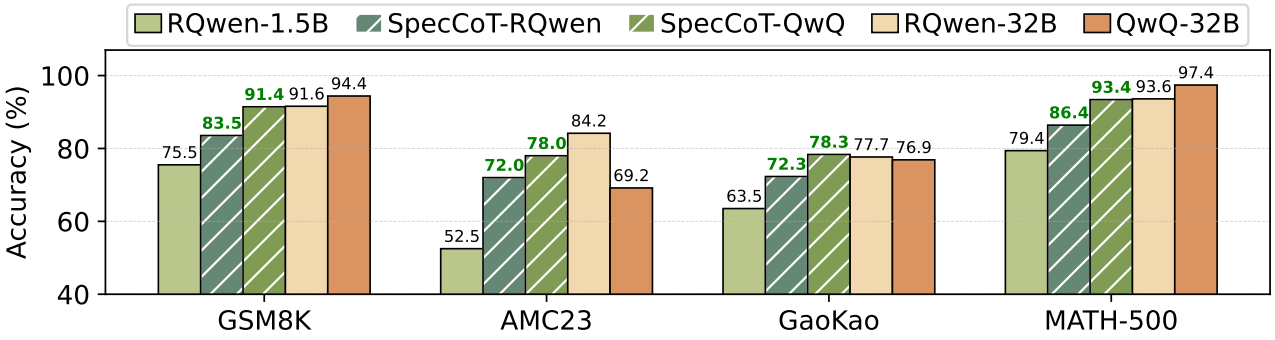


Figure 3: Accuracy comparison (%) of SpecCoT variants against baseline models across four reasoning benchmarks. SpecCoT implementations use Deepseek-R1-Distill-Qwen-1.5B (RQwen-1.5B) as the draft model, with either QwQ-32B or Deepseek-R1-Distill-Qwen-32B (RQwen-32B) as the target model.

Table 2: **Performance across four mathematical reasoning tasks (GSM8K, MATH-500, GaoKao, AMC23).** Metrics include average tokens, latency (Lat, in seconds), and improvement ratio r of SpecCoT over baselines.

Model	GSM8K			MATH-500			GaoKao			AMC23		
	Token	Lat	r	Token	Lat	r	Token	Lat	r	Token	Lat	r
RQwen-1.5B	1096.8	4.5	–	2609.3	10.6	–	3456.6	14.4	–	5298.8	21.64	–
QwQ-32B	2053.8	33.5	–	3086.8	50.8	–	4102.0	68.0	–	5885.8	97.78	–
SpecCoT(ours)	1023.3	10.1	3.3	1495.9	16.2	3.1	2316.5	30.1	2.3	3633.0	48.6	2.0
RQwen-32B	574.3	9.3	–	1998.6	32.6	–	2833.4	46.7	–	4605.5	76.2	–
SpecCoT(ours)	239.1	2.9	3.2	636.9	7.8	4.1	1674.7	24.0	1.9	3031.6	44.7	1.7

4. Experiments

4.1. Setup

SpecCoT is evaluated using RQwen-1.5B (DeepSeek-AI et al., 2025) (Deepseek-R1-Distill-Qwen series) as the draft model, and QwQ-32B (Team, 2025) or RQwen-32B as target models. Evaluation datasets include GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021), GaoKao-En-2023 (Liao et al., 2024), and AMC23 (MAA, 2023), covering diverse reasoning complexities. Experiments were conducted on 4 NVIDIA A100 GPUs using vLLM (Kwon et al., 2023) with 5 runs per configuration, 8192 max tokens, and temperature 0.6. Baselines are vanilla model inference.

4.2. Main Results

Accuracy Improvement. Figure 3 shows SpecCoT substantially outperforms the draft model, approaching target model accuracy on GSM8K and MATH-500. On AMC23 and GaoKao, the QwQ-32B baseline occasionally underperforms SpecCoT due to token needs, yet SpecCoT with QwQ-32B as the target model generally outperforms SpecCoT with RQwen-32B. These improvements stem from the target model's step-wise guidance and verification, which corrects draft model errors early in the reasoning process.

Lower Latency. SpecCoT significantly improves computational efficiency. With QwQ-32B as the target model, it achieves approximately 3× speedup on GSM8K/MATH-500, and 2.0-2.3× on GaoKao/AMC23. With RQwen-32B, speedups range from 1.7× (AMC23) to 4.1× (MATH-500). Diminishing gains on harder tasks are expected due to increased target model intervention, demonstrating the framework's balance between efficiency and quality requirements.

Token Efficiency. SpecCoT significantly reduces token consumption. With QwQ-32B as the target model, token usage decreases by approximately 50-52% on GSM8K and MATH-500. Using RQwen-32B, reductions reach 59-68% on these datasets. The gains result from early selection of optimal reasoning paths, preventing wasteful exploration and reducing the need for extensive large model generation, thereby producing concise yet accurate solutions.

5. Conclusion

We presented SpecCoT, demonstrating efficient CoT reasoning through strategic large-small model collaboration. By combining parallel candidate generation with selective verification, SpecCoT achieves 1.7-4.1× speedup while maintaining comparable accuracy across diverse reasoning benchmarks.

References

Burton, F. W. Speculative computation, parallelism, and functional programming. *IEEE Transactions on Computers*, 100(12):1190–1193, 1985.

Cai, T., Li, Y., Geng, Z., Peng, H., and Dao, T. Medusa: Simple framework for accelerating llm generation with multiple decoding heads. *Retrieved December*, 3:2024, 2023.

Chen, C., Borgeaud, S., Irving, G., Lespiau, J.-B., Sifre, L., and Jumper, J. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023. URL https://arxiv.org/abs/2302.01318.

Cheng, J. and Van Durme, B. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*, 2024.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Fu, Y., Bailis, P., Stoica, I., and Zhang, H. Break the sequential dependency of llm inference using lookahead decoding. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Fu, Y., Chen, J., Zhuang, Y., Fu, Z., Stoica, I., and Zhang, H. Reasoning without self-doubt: More efficient chain-of-thought through certainty probing. In *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025.

He, Z., Zhong, Z., Cai, T., Lee, J. D., and He, D. Rest: Retrieval-based speculative decoding. *arXiv preprint arXiv:2311.08252*, 2023.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Leviathan, Y., Kalman, M., and Matias, Y. 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*, pp. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Li, Y., Wei, F., Zhang, C., and Zhang, H. Eagle: speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Li, Y., Wei, F., Zhang, C., and Zhang, H. Eagle-2: Faster inference of language models with dynamic draft trees. *arXiv preprint arXiv:2406.16858*, 2024b.

Liao, M., Li, C., Luo, W., Jing, W., and Fan, K. Mario: Math reasoning with code interpreter output-a reproducible pipeline. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 905–924, 2024.

Liu, T., Guo, Q., Hu, X., Jiayang, C., Zhang, Y., Qiu, X., and Zhang, Z. Can language models learn to skip steps? *arXiv preprint arXiv:2411.01855*, 2024.

Ma, W., He, J., Snell, C., Griggs, T., Min, S., and Zaharia, M. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*, 2025.

MAA. American mathematics competition 2023 dataset. https://www.maa.org/math-competitions/amc-resources, 2023.

Miao, X., Oliaro, G., Zhang, Z., Cheng, X., Wang, Z., Zhang, Z., Wong, R. Y. Y., Zhu, A., Yang, L., Shi, X., Shi, C., Chen, Z., Arfeen, D., Abhyankar, R., and Jia, Z. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, ASPLOS '24, pp. 932–949, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703867. doi: 10.1145/3620666.3651335. URL https://doi.org/10.1145/3620666.3651335.

Pan, R., Dai, Y., Zhang, Z., Oliaro, G., Jia, Z., and Netravali, R. Specreason: Fast and accurate inference-time compute via speculative reasoning. *arXiv preprint arXiv:2504.07891*, 2025.

Spector, B. and Re, C. Accelerating llm inference with staged speculative decoding. *arXiv preprint arXiv:2308.04623*, 2023.

Team, Q. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL https://qwenlm.github.io/blog/qwq-32b/.

Wang, J., Li, J., Wu, L., and Zhang, M. Efficient reasoning for llms through speculative chain-of-thought. *arXiv preprint arXiv:2504.19095*, 2025a.

Wang, J., Li, W.-D., Paliotta, D., Ritter, D., Rush, A. M., and Dao, T. M1: Towards scalable test-time compute with mamba reasoning models. *arXiv preprint arXiv:2504.10449*, 2025b. URL https://arxiv.org/abs/2504.10449.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022a. Curran Associates Inc. ISBN 9781713871088.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022b.

Xia, H., Yang, Z., Dong, Q., Wang, P., Li, Y., Ge, T., Liu, T., Li, W., and Sui, Z. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *arXiv preprint arXiv:2401.07851*, 2024.

Xia, H., Li, Y., Leong, C. T., Wang, W., and Li, W. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*, 2025. URL https://arxiv.org/abs/2502.12067.

Xu, Y., Guo, X., Zeng, Z., and Miao, C. Softcot: Soft chain-of-thought for efficient reasoning with llms. *arXiv preprint arXiv:2502.12134*, 2025.

Yang, W., Yue, X., Chaudhary, V., and Han, X. Speculative thinking: Enhancing small-model reasoning with large model guidance at inference time. *arXiv preprint arXiv:2504.12329*, 2025.

Zhang, J., Wang, J., Li, H., Shou, L., Chen, K., Chen, G., and Mehrotra, S. Draft & verify: Lossless large language model acceleration via self-speculative decoding. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11263–11282, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.607. URL https://aclanthology.org/2024.acl-long.607/.

Zhang, J., Zhu, Y., Sun, M., Luo, Y., Qiao, S., Du, L., Zheng, D., Chen, H., and Zhang, N. Lightthinker: Thinking step-by-step compression. *arXiv preprint arXiv:2502.15589*, 2025. URL https://arxiv.org/abs/2502.15589.

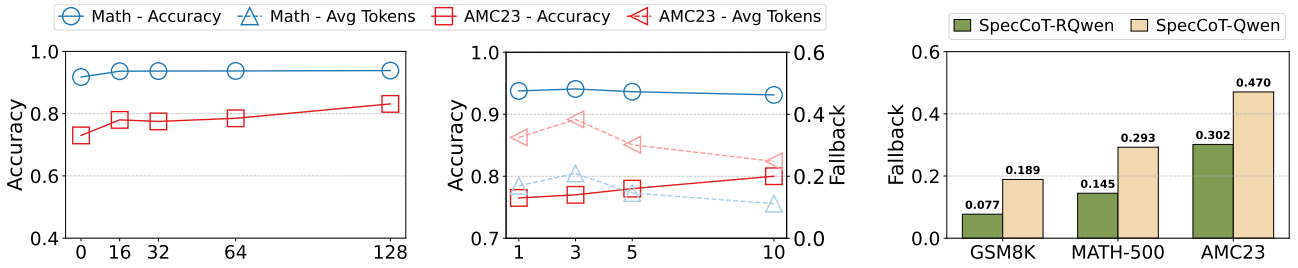
A. Appendix

A.1. Pseudo code of SpecCoT

Algorithm 1 Speculative Chain-of-Thought

Input: Input x, target model M_T, draft model M_D, candidates N, temperature τ
Output: Reasoning chain C, answer y
Reasoning chain $C \leftarrow \emptyset$
Initial guidance: $g \leftarrow M_T(x)$
repeat
  $i \leftarrow |C| + 1$
  candidates $\leftarrow \emptyset$
  for $j = 1$ **to** N **do**
    $\hat{c}_i^j \leftarrow M_D(g, C, \tau)$
    candidates $\leftarrow$ candidates $\cup \{\hat{c}_i^j\}$
  end for
  candidates $\leftarrow$ candidates $\cup \{\texttt{reject}\}$
  $s \leftarrow V_T(\text{candidates})$
  if $s = N + 1$ **then**
    $c_i \leftarrow M_T(g, C)$
  else
    $c_i \leftarrow \text{candidates}[s]$
  end if
  $C \leftarrow C \cup \{c_i\}$
  if reasoning complete **then**
    Extract y from C
    Return C, y
  end if
until reasoning complete =0

A.2. Ablations



((a)) Impact of initial guidance token count on model accuracy across datasets.

((b)) Relationship between number of drafts and accuracy/fallback rate.

((c)) Comparison of fallback rates across different reasoning datasets.

Figure 4: Ablation studies and sensitivity analysis of SpecCoT performance.

Number of initial tokens. Initial tokens are guiding tokens generated by the target model at the start of collaboration (zero means the draft model begins). Results show guiding tokens improved accuracy across three datasets, with a slight, acceptable inference time increase (still substantially lower than the base model). This accuracy gain stems from high-quality target model guidance preventing early draft model mistakes and effectively steering its generation.

Number of Drafts. The number of drafts dictates how many candidate continuations the draft model generates per step. Increasing drafts (Fig. 4(b)) doesn't guarantee higher accuracy but substantially lowers the fallback rate (notably at 10 drafts), as the target model more often finds acceptable candidates. While multiple drafts improve the chance of producing

a suitable continuation, they don't fundamentally boost the draft model's inherent problem-solving ability. For complex junctures beyond the draft model's capacity, target model intervention remains necessary.

Table 3: **SpecCoT performance on MATH-500 and GSM8K.** Metrics include accuracy (%), tokens, time (s), and fallback rate (%) using two draft models (RQwen-1.5B and Qwen2.5-1.5B-Instruct) with three 32B target models (RQwen for RQwen-32B, Qwen-Inst. for Qwen2.5-32B-Instruct, and QwQ for QwQ-32B).

Draft Model	Metric	MATH-500			GSM8K		
		RQwen	Qwen-Inst.	QwQ	RQwen	Qwen-Inst.	QwQ
RQwen-1.5B	Accuracy	83.6	80.7	91.4	86.8	83.5	93.5
	Tokens	242.2	265.1	1021.9	638.6	319.3	1495.9
	Time	2.9	2.9	9.8	7.5	3.2	16.2
	Fallback	8.6	2.5	6.5	16.4	2.7	14.5
Qwen2.5-1.5B Instruct	Accuracy	75.7	77.0	81.8	68.5	67.0	74.4
	Tokens	254.1	247.3	347.8	324.8	292.2	536.3
	Time	5.5	2.9	5.0	6.3	3.4	8.6
	Fallback	5.6	4.7	18.9	13.5	4.9	29.3

A.3. Analysis

A.3.1. Impact of Model Capabilities

To analyze how model capabilities affect collaboration, we tested various combinations of draft and target models. Target models ranged from basic instruction-following (Qwen-32B-Instruct) to enhanced reasoning (RQwen-32B) and sophisticated chain-of-thought reasoning (QwQ-32B), while draft models included reasoning-enhanced RQwen-1.5B and standard Qwen2.5-1.5B. Results show that stronger target models achieve higher accuracy but require more computational resources, with QwQ-32B reaching 93.5% accuracy but consuming 1495.9 tokens on average. When paired with RQwen-1.5B as the draft model, even basic Qwen-32B-Instruct maintains good efficiency (265.1 tokens, 2.9s inference time) while achieving 80.7% accuracy. Draft model capability significantly impacts performance - RQwen-1.5B enables notably higher accuracy across all target models (91.4% vs 81.8% with QwQ-32B on MATH-500) with lower fallback rates (6.5% vs 18.9%). However, even with weaker Qwen2.5-1.5B drafts, QwQ-32B still achieves 74.4% accuracy on GSM8K, demonstrating the robustness of our collaborative approach.

A.3.2. Analysis of Fallback Rate

The fallback rate represents the proportion of the draft model's intermediate steps rejected by the target model. On relatively simpler datasets like GSM8K, we observe low fallback rates of approximately 0.08, indicating that the draft model produces acceptable steps for most reasoning stages(Fig. 4(c)). However, this rate increases dramatically to 0.47 on challenging benchmarks like AMC23, reflecting the draft model's diminished capability to generate reliable continuations for complex problems. Our analysis reveals that dataset difficulty serves as the primary determinant of fallback rates. While increasing the number of draft candidates moderately improves acceptance, this effect is less pronounced than the impact of inherent problem complexity. Interestingly, we find that stronger target models tend to exhibit higher fallback rates, as their enhanced reasoning capabilities enable more stringent evaluation of the draft model's proposals.

A.3.3. Impact of resource configurations

To examine SpecCoT's performance under different resource configurations, we evaluated the performance using vLLM with 2-way and 4-way tensor parallelism on 2 and 4 A100 GPUs, respectively. The results demonstrate consistent performance improvements across both datasets. In terms of inference time, SpecCoT achieves the fastest processing speed in both hardware settings, completing inference in 12.49s and 10.08s on GSM8K, and 20.55s and 16.17s on MATH-500 under 2-way and 4-way configurations. The throughput analysis reveals that while all models benefit from increased parallelism, the efficiency gains vary. Large models like RQwen-32B and QwQ-32B show substantial throughput improvements (¿50%) when scaling from 2 to 4 GPUs, while SpecCoT exhibits moderate gains of 23.52% on GSM8K and 24.95% on MATH-500. This difference is attributed to the fact that larger target models benefit more from multi-GPU tensor parallelism compared to the smaller draft model used in SpecCoT.

Table 4: **Inference efficiency comparison on GSM8K and MATH-500.** Time (seconds), throughput (tokens/second), and their improvements when scaling from 2×A100 to 4×A100 GPU configurations.

Dataset	Model	2×A100 Time (s)	4×A100 Time (s)	Time Speedup (%)	2×A100 Token/s	4×A100 Token/s	Throughput Gain (%)
GSM8K	RQwen-1.5B	12.51	10.94	12.55%	236.23	256.38	8.53%
	RQwen-32B	14.71	10.33	29.78%	44.56	67.90	52.38%
	QwQ-32B	51.86	33.57	35.27%	41.17	63.31	53.78%
	SpecCoT(ours)	**12.49**	**10.08**	19.30%	82.17	101.50	23.52%
MATH-500	RQwen-1.5B	34.28	33.13	3.35%	233.25	244.82	4.96%
	RQwen-32B	69.20	43.96	36.47%	39.88	61.59	54.44%
	QwQ-32B	81.54	53.96	33.83%	40.55	62.03	52.97%
	SpecCoT(ours)	**20.55**	**16.17**	21.31%	74.00	92.46	24.95%

A.4. Related Works

A.4.1. Efficient Reasoning

Recent LLMs have adopted chain of thought (CoT) reasoning(Wei et al., 2022a), enhancing problem-solving capabilities while introducing longer outputs and increased computational costs. Research on improving reasoning efficiency follows two main approaches: length compression methods like TokenSkip(Xia et al., 2025), SoftCoT(Xu et al., 2025), and Compressed CoT(Cheng & Van Durme, 2024) reduce verbose outputs while maintaining quality; early termination techniques such as Dynasor(Fu et al., 2025) and NoThinking(Ma et al., 2025) optimize reasoning by identifying when to stop processing. These advances enable LLM applications in resource-constrained environments without sacrificing quality. Several recent works combine speculative decoding with CoT reasoning, including SCoT(Wang et al., 2025a) using LoRA-tuned draft models, SpecReason(Pan et al., 2025) decomposing CoT into discrete speculative steps, and Speculative Thinking(Yang et al., 2025) triggering large model intervention at critical junctures identified by reflection keywords.

Our approach differs fundamentally from these concurrent works in two critical aspects. First, unlike these methods, which allow the small model to initiate reasoning, we use the large model to generate initial tokens, preventing the small model from starting on a flawed reasoning path. Second, rather than operating on complete reasoning trajectories like SCoT, our approach applies best-of-n selection and fallback at intermediate CoT steps, enabling the large model to guide the reasoning process at multiple points. This ongoing involvement allows for earlier intervention when the small model shows signs of error, leading to more accurate reasoning while maintaining efficiency.

A.4.2. Speculative Decoding

Speculative decoding mitigates inference latency in autoregressive language models by enabling parallel token generation without compromising output quality. Inspired by speculative execution in computing (Burton, 1985), seminal works by Leviathan et al.(Leviathan et al., 2023) and Chen et al.(Chen et al., 2023) demonstrated this approach's effectiveness. Implementation strategies vary widely: from leveraging smaller models(Leviathan et al., 2023; Chen et al., 2023; Spector & Re, 2023) and target model components(Cai et al., 2023; Zhang et al., 2024) to utilizing n-gram tables(Fu et al., 2024) and retrieval systems(He et al., 2023). Verification techniques have advanced from basic token-level checks(Leviathan et al., 2023) to sophisticated tree-structured methods(Miao et al., 2024). Recent breakthroughs include feature-level processing in EAGLE(Li et al., 2024a) and adaptive draft trees in EAGLE-2(Li et al., 2024b), delivering enhanced acceleration while preserving or improving quality.