Think Deep, Think Fast: Investigating Inference-Time Scaling And The Reasoning Floor

Junlin Wang [1 2] **Shang Zhu** [2] **Jon Saad-Falcon** [3] **Ben Athiwaratkun** [2] **Qingyang Wu** [2] **Jue Wang** [2]
Shuaiwen Leon Song [2] **Ce Zhang** [2 4] **Bhuwan Dhingra** [1] **James Zou** [2 3]

Abstract

There is intense interest in investigating how inference time compute (ITC) (e.g. repeated sampling, refinements, etc) can improve large language model (LLM) capabilities. While breakthroughs like DeepSeek-R1 highlight the power of reinforcement learning for reasoning, the interaction between ITC and reasoning-optimized weights remains poorly understood. This work conducts a comprehensive analysis of inference-time scaling methods for both reasoning and non-reasoning models on challenging reasoning tasks. While prior work suggests that scaling test-time compute can optimally substitute for model parameter scaling, we identify a fundamental limit to this compute-equivalence: the Reasoning Floor. We demonstrate that general-purpose models fail to match the accuracy of reasoning-optimized models even with an order of magnitude more inference compute, suggesting that internalizing reasoning protocols is a prerequisite for effective test-time scaling. Within reasoning models, we find that the complexity of the scaling method often yields diminishing returns; simple majority voting consistently outperforms sophisticated sequential revision and mixture-of-agents frameworks. Crucially, we identify a "Linguistic Signal of Correctness"—correct responses are significantly more concise and exhibit a lower density of "hedging" and "thinking" markers. We demonstrate that these intrinsic linguistic features can serve as zero-compute proxies for response quality, providing a pathway to more efficient, self-diagnostic reasoning agents.

[1]Duke University [2]Together AI [3]Stanford University [4]University of Chicago. Correspondence to: Junlin Wang <junlin.wang@duke.edu>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

1. Introduction

Language models have witnessed remarkable advancements in recent years, demonstrating increasingly sophisticated capabilities across various tasks (OpenAI, 2023; Dubey et al., 2024; Bai et al., 2023). Despite these improvements, complex reasoning remains challenging, often requiring additional computational resources and specialized techniques to achieve satisfactory performance (Wang et al., 2024b; Yao et al., 2023). This challenge has motivated the development of inference-time compute (ITC) scaling methods, which allocate additional computational resources during inference to enhance model outputs.

The landscape of language model reasoning has evolved along two primary dimensions. First, approaches like Chain-of-thought (Wei et al., 2023), self-consistency (Wang et al., 2022), tree-structured sampling (Snell et al., 2024), and mixture of agents (Wang et al., 2025) have emerged as effective techniques for boosting reasoning capabilities during inference without requiring model parameter changes. Second, a new class of "reasoning models", explicitly post-trained to solve highly challenging problems, has been introduced, exemplified by models like o1 (OpenAI et al., 2024), Deepseek-R1 (DeepSeek-AI et al., 2025), and QwQ (Team, 2024).

While both approaches show promise, they introduce significant computational overhead. Chain-of-thought (Wei et al., 2023) prompting increases token generation, tree-structured sampling requires exploring multiple solution paths (Liu et al., 2025), and mixture of agents (Wang et al., 2025) demands running several specialized agent configurations simultaneously. This computational burden raises critical questions about efficiency: how can we optimize the trade-off between computational resources and reasoning performance? Which inference-time scaling methods deliver the best results for different model architectures? How do reasoning models compare to conventional models under varying computational budgets?

These questions remain largely unsolved in the current literature. Prior work has primarily focused on fine-tuning strategies (Yu et al., 2025; Xie et al., 2025) or simple inference

evaluations (Brown et al., 2024a), with limited attention to systematic evaluation of inference-time compute scaling across model types. Furthermore, a line of works suggest that inference-time compute can be more effective than model parameter scaling (Snell et al., 2024)—these findings were largely established prior to the widespread availability of specialized reasoning architectures like DeepSeek-R1. Consequently, there is a lack of systematic evidence regarding whether this equivalence holds across different model classes. Furthermore, existing research often relies on resource-heavy process-reward models (PRMs), leaving the potential of reward-model-free (RM-free) scaling methods relatively misunderstood.

Our work bridges this gap by conducting a comprehensive analysis of RM-free inference-time scaling across both reasoning-optimized and general-purpose architectures. We specifically focus on trained-verifier-free approaches to investigate the upper bounds of intrinsic model reasoning without external supervision. Our results challenge the compute-equivalence narrative, revealing a fundamental **Reasoning Floor**: a performance ceiling where non-reasoning models fail to convert additional search budget into accuracy gains. In contrast, for reasoning-optimized models, we find that increasing method complexity (e.g., sequential revision) often yields diminishing returns compared to simple majority voting. Our research provides the following **key contributions**:

- We compare inference-time scaling methods across reasoning and non-reasoning models, establishing efficiency–performance trade-off curves. We find that complex inference-time strategies yield diminishing returns, with simple majority voting consistently outperforming more elaborate methods.
- We provide empirical evidence that inference-time compute is not a universal substitute for specialized training. We show that non-reasoning models are fundamentally limited by a "Reasoning Floor," where even an order of magnitude more compute cannot match the performance of reasoning-optimized weights.
- We analyze the correlation between response linguistic features (response length, linguistic markers) and task performances, providing practical guidance for improving existing inference-time methods without increasing computational costs.

2. Related Work

Scaling compute during inference offers a promising alternative to costly model pretraining. Brown et al. (2024b) show a log-linear relationship between problem-solving coverage and sample count across reasoning tasks, while inference-time architectures combine generation, ranking, and fusion to exceed individual model performance, including Mixture-of-Agents (Wang et al., 2024c), LLM Blender (Jiang et al., 2023), and orchestration frameworks like DSPy (Khattab et al., 2023). Even with single models, techniques like chain-of-thought (Wei et al., 2023) and branch-solve-merge (Saha et al., 2024) enhance reasoning capabilities. Our work extends this literature by focusing specifically on trained-verifier-free inference-time scaling methods that do not require additional reward models, and by systematically constructing efficiency–performance trade-off curves across both reasoning-optimized and general-purpose architectures. A more detailed discussion of related work on inference efficiency (repeated sampling refinements, MCTS-based methods) is provided in Section B.

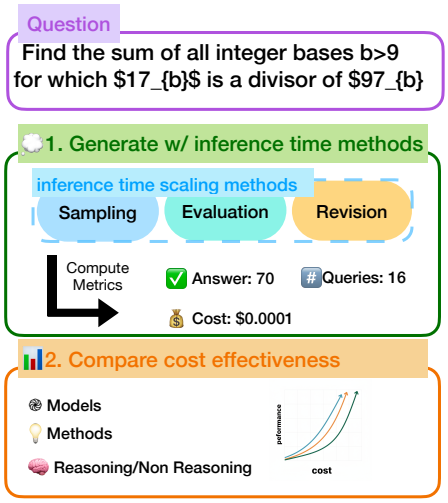

Figure 1. We analyze the efficacy of various reasoning methods and models.

3. Methodology

3.1. Models

The study evaluates a diverse set of models to cover a wide range of model sizes and architectures, crucial for understanding the effectiveness of ITC methods across different capabilities. The models are categorized into non-reasoning and reasoning models, reflecting their primary strengths and training focus.

Non-reasoning models Non-reasoning models are general-purpose LLMs optimized for tasks like text generation and dialogue, but they may lack specialized training for complex reasoning. The selected models include: GPT-4o-mini, Qwen2.5-7B-Instruct, Qwen2.5-72B-Instruct (Bai et al., 2023), Llama-3.3-70B-Instruct, and Llama-3.1-8B-Instruct (Dubey et al., 2024). This selection

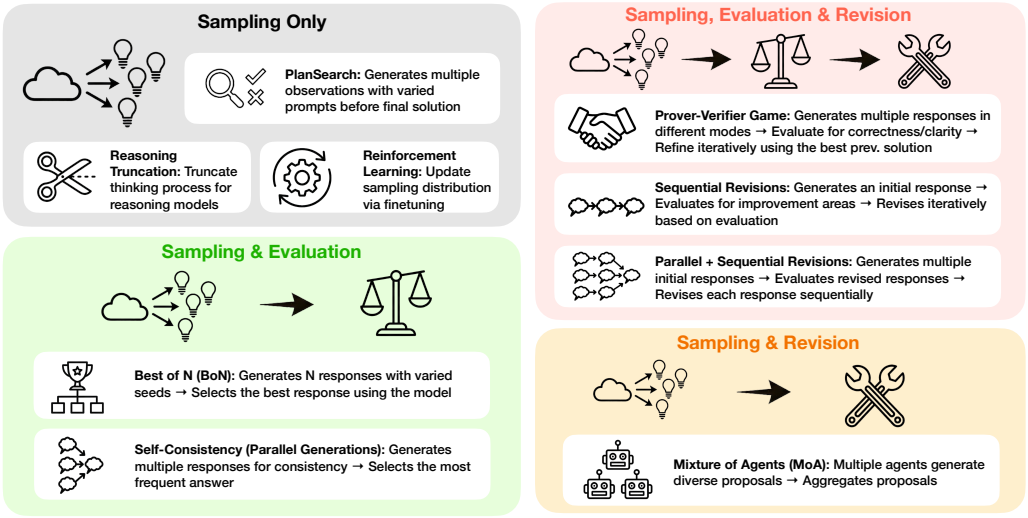

Figure 2. Inference-Time Scaling Methods: Sampling, Evaluation, and Revision approaches. Note that we view reinforcement learning and reasoning truncation as an approach to change the sampling distribution.

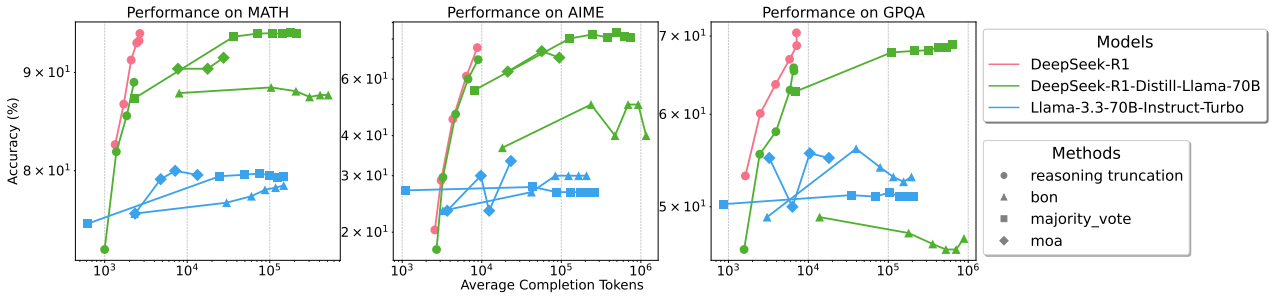

Figure 3. The overview of inference-time-compute methods for reasoning and non-reasoning models. Even though inference-time scaling method improves Llama-3.3-70B, it still struggles to beat the R1-distilled version of Llama 70B. However, with very limited compute, non-reasoning model with inference method can be at the trade-off curve.

covers a wide range of sizes (from 8B to 72B parameters) and includes both open-source and closed-source models, ensuring a comprehensive evaluation. The inclusion of GPT-4o-mini as a closed-source model contrasts with the open-source Qwen2.5 and Llama series, highlighting the study's intent to assess performance across different accessibility models.

Reasoning models Reasoning models are specifically trained or designed to handle complex reasoning tasks, such as mathematical problem-solving and code generation, often through methods like reinforcement learning (RL). The selected models include: DeepSeek-R1-Distill-Llama-70B (DeepSeek-AI et al., 2025), DeepSeek-R1-Distill-Llama-8B, DeepSeek-R1-Distill-Qwen-32B, DeepSeek-R1-Distill-Qwen-14B, DeepSeek-R1-Distill-Qwen-7B, and QwQ-32B-Preview (Team, 2024). For some experiments we also evaluate DeepSeek-R1 (DeepSeek-AI et al., 2025), but not all due to the high-cost nature of inference-scaling methods.

3.2. Tasks

We evaluate on four challenging reasoning benchmarks spanning mathematics, science, and code: MATH 500 (Hendrycks et al., 2021), AIME 2024 (30 competition problems), GPQA Diamond (Rein et al., 2023) (a high-quality subset of graduate-level science multiple-choice questions), and the LiveCodeBench code-generation subtask (Jain et al., 2024) (problems from 2024-11-01 to 2025-02-01). Full task descriptions are provided in Section C.

3.3. Inference-Time Scaling Methods

We decompose inference-time scaling methods into three operations (Figure 2): *sampling* (generating one or more responses, possibly with input-level modifications such as chain-of-thought prompts that shift the output distribution), *evaluation* (scoring or ranking responses to concentrate the distribution on higher-quality outputs; in this work all verifiers are LLMs, so the same model performs sampling and

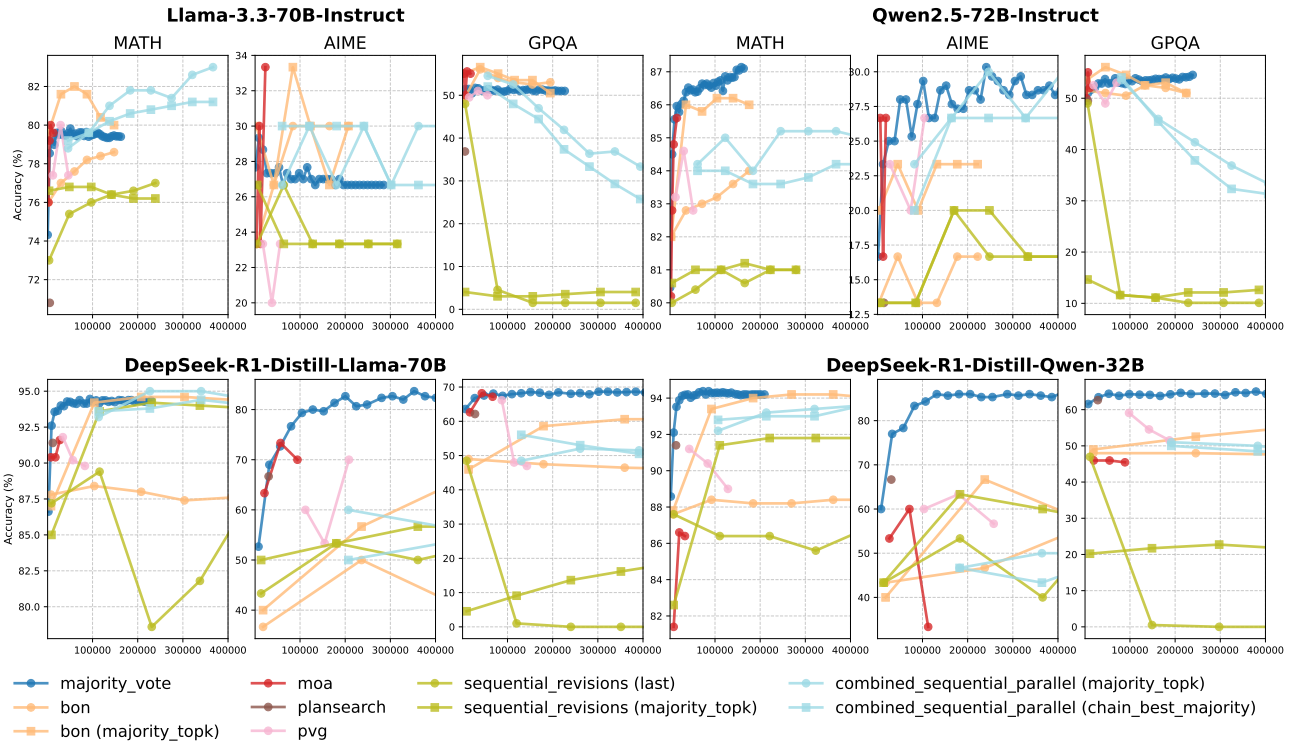

Figure 4. Performance of various inference-time scaling methods for four models across MATH, AIME and GPQA. For some methods we have multiple eval metrics. For *bon* (best-of-N approach), we pick the highest scored response. *majority topk* means we use top k scored resposnes to do majority voting (we always set k as half of total samples). *last* means we pick the last revisioned sample. *chain best majority* indicates that we use the best scored sample from each chain and then take majority. Due to high cost of inference, methods like sequential revisions and combined sequential parallel are only sampled once, which may seem volatile when plotted. The results for other models can be found in the Appendix.

evaluation), and *revision* (iteratively modifying responses based on evaluation feedback). We study seven representative methods spanning these operations: PlanSearch (Wang et al., 2024a), Prover-Verifier Game (PVGame) (Kirchner et al., 2024), Mixture of Agents (MoA) (Wang et al., 2025), Best of N (BoN), Self-consistency (Wang et al., 2022), Sequential Revisions, and Parallel + Sequential Revisions (Snell et al., 2024). Per-method parameterizations are provided in Section D.

4. The Limits of Inference-Time Scaling

In this section, we will give an overview of various inference-time scaling methods and present the current trade-off curve for those methods. We aim to provide a comprehensive view and provide guidelines for what model or method should be use with a given inference budget.

4.1. The Reasoning Floor: Between Test-Time Scaling and Train-Time Scaling

While prior work has suggested that increased inference-time compute can substitute for model parameters (Snell et al., 2024), Figure 3 identifies a distinct **Reasoning Floor**: a performance ceiling below which general-purpose models plateau, regardless of the search budget. Specifically, we find that reasoning-optimized models (e.g., R1-Distill-Llama-70B) demonstrate superior token efficiency, achieving peak accuracy with orders of magnitude less compute than their non-reasoning counterparts like Llama-3.3-70B-Instruct (Similar trend is observed in Figure 4). Even when general models are provided with an expansive inference budget ($N = 256$), they fail to bridge the performance gap to reasoning-optimized weights. This suggests that internalizing reasoning protocols via reinforcement learning is a more effective test-time scaling; without these internalized weights, external search methods yield diminishing returns.

To further isolate the value of these internalized protocols, we introduced a "Reasoning Truncation" baseline. By force-terminating the reasoning process (within the ⟨think⟩ blocks) and prompting for immediate sequence completion, we observed a catastrophic degradation in response quality. Interestingly, as the reasoning budget is aggressively truncated, the performance of reasoning models eventually regresses to the "floor" occupied by non-reasoning models. This

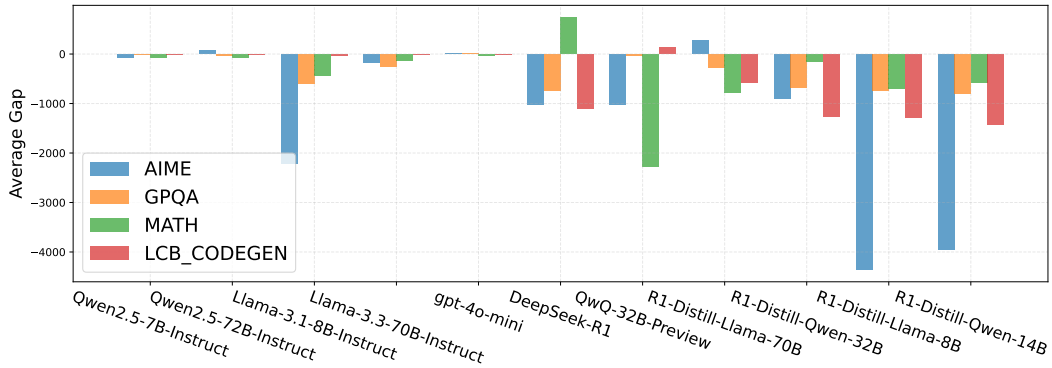

Figure 5. The average response length gap for each model tasks across four tasks. The average response length gap is computed by: 1) calculating mean length difference between correct and incorrect responses within each question and 2) averaging these differences across the entire dataset. LCB_CODEGEN represents the code generation subtask in the LiveCodeBench benchmark.

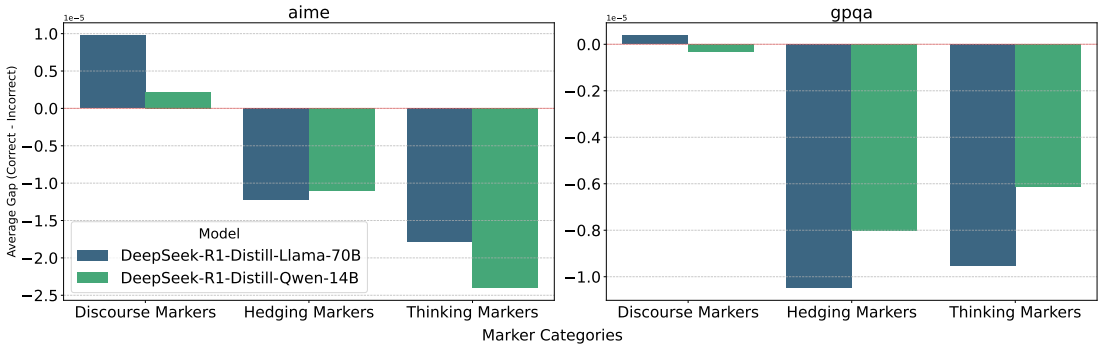

Figure 6. Average gaps between correct and incorrect responses for DeepSeek-R1-Distill-Llama-70B and DeepSeek-R1-Distill-Llama-14B. The average gaps are first computed by using computing the mean difference of thinking token frequency of correct and incorrect responses within one question and then average over the entire dataset. The frequency is weighted by response length. Refer to Table 1 for the definition of different marker categories.

suggests that the latent "thought" tokens are not merely a byproduct of scale, but the primary vehicle through which compute is converted into logic.

4.2. Internalized Reasoning vs. Externalized Search Saturation

Training-free, verifier-free inference-time scaling methods offer minimal improvements for reasoning models. As shown in Figure 4, these sophisticated methods fail to significantly enhance the performance of reasoning models. Almost all the methods are underperforming majority voting for both DeepSeek-R1-Distill-Llama-70B and DeepSeek-R1-Distill-Qwen-32B. We hypothesized excessive external revision often introduces *reasoning drift*, where the model's high-probability internal path is distracted by sub-optimal external feedback.

Conversely, non-reasoning models (e.g., Llama-3.3-70B-Instruct) show greater receptivity to externalized search, occasionally surpassing majority voting. This suggests that for models without specialized weights, external loops can partially simulate the reasoning protocols established in specialized architectures (Snell et al., 2024). However, as established by the Reasoning Floor, these gains are ultimately capped by the model's underlying weights. We conclude that for reasoning-optimized models, the most compute-efficient strategy is to prioritize sample diversity over search complexity.

4.3. Majority Voting as the Compute-Optimal Baseline

Across both reasoning and non-reasoning models, majority voting consistently demonstrates superior performance. The method's simplicity belies its effectiveness, as illustrated in Figure 4, where it outperforms more complex inference-time scaling approaches such as mixture-of-agents, best-of-N, sequential revisions and combined parallel sequential. The success of majority voting can be attributed to its fundamental approach of leveraging multiple model outputs. By aggregating responses, the method effectively mitigates individual model biases and captures a more robust representation of the underlying reasoning.

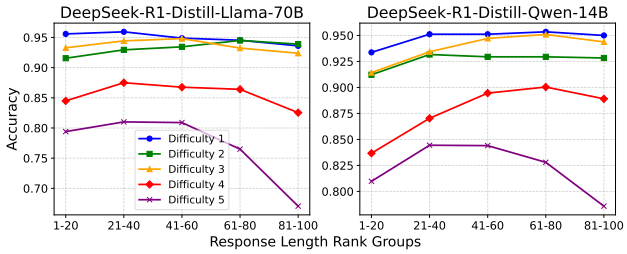

Figure 7. Accuracy of responses in different length groups for various difficulty. For each question, we generate 100 samples and then we bin those samples into five bins. Then average accuracy is computed for each bin across the dataset.

4.4. Scaling Behavior vs. Problem Difficulty

Figure 5 reveals a critical insight into model performance across varying task complexities. As problem difficulty increases, the gap between correct and incorrect responses becomes more pronounced, particularly for reasoning models. The AIME dataset, known for its challenging nature, exemplifies this trend, with all reasoning models demonstrating a wider correctness gap. To systematically investigate this phenomenon, we analyze response length differences using the MATH dataset, which offers a natural difficulty gradient ranging from level one to five. We stratify samples for each question into five bins ranked from shortest to longest responses. We find that for high-difficulty problems (level 5), where an inverse relationship between response length and correctness became evident. Specifically, Figure 7 demonstrates that as problem complexity increases, reasoning models tend to generate more accurate responses with shorter lengths.

5. Diagnostic Signals in Reasoning Traces

5.1. The Impact of Response Length to Performance

Our analysis reveals a distinct divergence between model types. Figure 5 plots the average response length gap between correct and incorrect responses (computed per question—using 100 samples for reasoning models and 256 for non-reasoning models—then averaged across the dataset, discarding questions that are entirely correct or incorrect to avoid bias from trivial or unsolvable items).

Non-reasoning models show no correlation. The length gap is consistently small across all four datasets, with the only notable exception being Llama-3.1-8B-Instruct on AIME.

Reasoning models show an inverse correlation. Shorter, more precise responses are markedly more likely to be correct. This suggests that for reasoning-optimized weights, correctness is tied to efficient reasoning paths, while incorrect outputs often involve redundant logic, circular thinking, or failed self-correction cycles that inflate response length—making verbosity a diagnostic signal for reasoning failure.

5.2. Linguistic Markers and Word Frequency Analysis

Reasoning models tend to use certain linguistic markers, especially thinking tokens such as “alternatively” or “however”. In this section, we investigate the relationships between such linguistic markers and correctness.

Markers appear more often in incorrect responses. Figure 6 shows that both hedging and thinking markers (defined in Table 1) are markedly more prevalent in incorrect responses. We compute marker frequency gaps as the mean per-question difference between correct and incorrect responses, normalized by response length, then averaged across the dataset. This proliferation of markers in incorrect responses appears to signal cognitive imprecision rather than reasoning depth, providing a promising diagnostic for identifying low-quality outputs.

Markers are a good predictor of correctness. To test the predictive power of these markers, we generated 100 responses per question and trained a logistic regression classifier on marker features with a 0.6/0.2/0.2 train-validation-test split. The classifier achieved a test F1 of 0.7469 on DeepSeek-R1-Distill-Llama-70B and 0.8637 on DeepSeek-R1-Distill-Llama-14B, suggesting linguistic markers offer a nuanced, zero-compute signal for assessing response quality beyond traditional length metrics.

6. Conclusion

Our work thoroughly assesses trained-verifier-free inference-time scaling methods for LLMs, emphasizing their efficiency and effectiveness in reasoning tasks. We identify a fundamental **Reasoning Floor**: a performance ceiling where general-purpose models, despite leveraging advanced scaling techniques and significant computational resources, fail to match the baseline accuracy of specialized reasoning architectures For reasoning models, simpler strategies such as majority voting often surpass more intricate methods like best-of-N or sequential revisions in performance. Furthermore, our analysis reveals that correct reasoning traces are characterized by higher efficiency—typically appearing as shorter responses with a lower density of linguistic markers such as hedging and thinking tokens. These intrinsic features serve as zero-compute diagnostic signals for accuracy. Leveraging these linguistic markers to develop self-correcting agents and more efficient aggregation methods represents a promising frontier for advancing stable, complex reasoning.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., and Mirhoseini, A. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv: 2407.21787*, 2024a.

Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., and Mirhoseini, A. Large language monkeys: Scaling inference compute with repeated sampling, 2024b. URL https://arxiv.org/abs/2407.21787.

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv: 2501.12948*, 2025.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Garcia, L. and Martinez, M. Adaptive branching mcts for multi-turn refinement in language generation. In *Proceedings of the 2024 International Conference on Learning Representations (ICLR)*, 2024.

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.

Jain, N., Han, K., Gu, A., Li, W.-D., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., and Stoica, I. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Jiang, D., Ren, X., and Lin, B. Y. Llm-blender: Ensembling large language models with pairwise comparison and generative fusion. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, 2023.

Khattab, O., Singhvi, A., Maheshwari, P., Zhang, Z., Santhanam, K., Vardhamanan, S., Haq, S., Sharma, A., Joshi, T. T., Moazam, H., Miller, H., Zaharia, M., and Potts, C. Dspy: Compiling declarative language model calls into self-improving pipelines, 2023. URL https://arxiv.org/abs/2310.03714.

Kirchner, J. H., Chen, Y., Edwards, H., Leike, J., McAleese, N., and Burda, Y. Prover-verifier games improve legibility of llm outputs. *arXiv preprint arXiv: 2407.13692*, 2024.

Lee, J. and Kim, K. Mct-self-refine: Integrating monte carlo tree search with self-evaluation in language models. *arXiv preprint arXiv:2406.98765*, 2024.

Liu, D. and Chen, E. Divsampling: Enhancing diversity in language model inference. *arXiv preprint arXiv:2405.13579*, 2024.

Liu, R., Gao, J., Zhao, J., Zhang, K., Li, X., Qi, B., Ouyang, W., and Zhou, B. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv: 2502.06703*, 2025.

OpenAI. Gpt-4 technical report, 2023.

OpenAI, :, Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., Iftimie, A., Karpenko, A., Passos, A. T., Neitz, A., Prokofiev, A., Wei, A., et al. Openai o1 system card. *arXiv preprint arXiv: 2412.16720*, 2024.

Patel, N. and Sharma, P. Memory-augmented language models with mcts planning for text-based game agents. *arXiv preprint arXiv:2407.54321*, 2024.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv: 2311.12022*, 2023.

Saha, S., Levy, O., Celikyilmaz, A., Bansal, M., Weston, J., and Li, X. Branch-solve-merge improves large language model evaluation and generation, 2024. URL https://arxiv.org/abs/2310.15123.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv: 2408.03314*, 2024.

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

Wang, A. and Li, B. Self-consistency decoding for large language models. *arXiv preprint arXiv:2403.12345*, 2024.

Wang, E., Cassano, F., Wu, C., Bai, Y., Song, W., Nath, V., Han, Z., Hendryx, S., Yue, S., and Zhang, H. Planning in natural language improves llm search for code generation. *arXiv preprint arXiv: 2409.03733*, 2024a.

Wang, J., Jain, S., Zhang, D., Ray, B., Kumar, V., and Athiwaratkun, B. Reasoning in token economies: Budget-aware evaluation of LLM reasoning strategies. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 19916–19939, Miami, Florida, USA, nov 2024b. Association for Computational Linguistics. URL https://aclanthology.org/2024.emnlp-main.1112.

Wang, J., Wang, J., Athiwaratkun, B., Zhang, C., and Zou, J. Mixture-of-agents enhances large language model capabilities, 2024c. URL https://arxiv.org/abs/2406.04692.

Wang, J., WANG, J., Athiwaratkun, B., Zhang, C., and Zou, J. Mixture-of-agents enhances large language model capabilities. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=h0ZfDIrj7T.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv: 2203.11171*, 2022.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Xie, T., Gao, Z., Ren, Q., Luo, H., Hong, Y., Dai, B., Zhou, J., Qiu, K., Wu, Z., and Luo, C. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv: 2502.14768*, 2025.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Fan, T., Liu, G., Liu, L., Liu, X., Lin, H., Lin, Z., Ma, B., Sheng, G., Tong, Y., Zhang, C., Zhang, M., Zhang, W., Zhu, H., Zhu, J., Chen, J., Chen, J., Wang, C., Yu, H., Dai, W., Song, Y., Wei, X., Zhou, H., Liu, J., Ma, W.-Y., Zhang, Y.-Q., Yan, L., Qiao, M., Wu, Y., and Wang, M. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv: 2503.14476*, 2025.

Zhang, C. and Kumar, D. Confidence-informed self-consistency for efficient inference. *arXiv preprint arXiv:2404.67890*, 2024.

Category	Examples
Discourse Markers	*on the other hand*, *nevertheless*, *moreover*, *in addition*, *furthermore*, *therefore*, *consequently*, *as a result*
Hedging Markers	*perhaps*, *maybe*, *possibly*, *it seems*, *might*, *could*
Thinking Markers	*however*, *wait*, *alternatively*, *hmm*

Table 1. Categories of Linguistic Markers. Italicized words represent specific markers.

A. Usage of LLM

We utilize LLM to assist some of the paper writing, code development and figure generation.

B. Extended Related Work: Inference Efficiency

Recent work has explored inference-time optimization strategies that trade additional computation for improved model performance. A key approach is repeated sampling and majority voting, leveraging LLMs' generative diversity (Brown et al., 2024b; Wang & Li, 2024). To reduce computational cost, refinements like Confidence-Informed Self-Consistency (CISC) use confidence-weighted voting, cutting required samples by over 40% (Zhang & Kumar, 2024). Another strategy, DivSampling, injects prompt perturbations to increase answer diversity, boosting performance across math, reasoning, and code generation (Liu & Chen, 2024). These methods illustrate a broader trend: sacrificing more inference compute (e.g., multiple forward passes) for superior results.

Another active direction integrates Monte Carlo tree search (MCTS) with LLMs to enhance inference-time exploration. MCT-Self-Refine (MCTSr) combines iterative tree search and self-evaluation, significantly improving math problem-solving (Lee & Kim, 2024). Adaptive Branching MCTS (AB-MCTS) refines search by dynamically deciding to explore new candidates or refine existing ones, outperforming fixed-branch MCTS on coding and engineering tasks (Garcia & Martinez, 2024). Beyond traditional NLP, MCTS has been applied to agentic settings: a text-based game agent equipped with memory-augmented MCTS planning achieved higher scores by learning from past trials (Patel & Sharma, 2024).

C. Task Details

MATH 500 MATH 500 is a subset of challenging competition mathematics problems from the MATH dataset (Hendrycks et al., 2021). The dataset contains complex high school math problems that are often solved with step-by-step reasoning.

AIME AIME consists of problems from the American International Mathematics Examination (AIME), a prestigious mathematics competition for high school students. AIME is known for its difficult and thought-provoking problems and we select the 2024 subset as our evaluation benchmark which contains 30 problems.

GPQA Graduate-Level Google-Proof Q&A Benchmark (GPQA) (Rein et al., 2023), is a dataset of 448 multiple-choice questions in biology, physics, and chemistry, crafted by domain experts. It is designed to be extremely challenging, with experts who have or are pursuing PhDs in the corresponding domains reaching only 65% accuracy. We use GPQA Diamond which is a high quality subset.

LiveCodeBench Livecodebench (Jain et al., 2024) provides a holistic and contamination-free assessment of large language models for code-related tasks. The code generation (codegen) subtask, specifically, is selected in this work to test the models' ability to generate correct code for these problems. We use the code problems from 2024-11-01 to 2025-02-01.

D. Inference-Time Scaling Methods

In this section, we will go into more detils about each inference-time scaling methods and how they are parametrized in our studies.

Majority Vote Majority vote (self-consistency) generate multiple samples and choose the most frequent answer as final solution. Note that this method doesn't quite work for free-form generation problems such as LiveCodeBench. Hence we don't present results for LiveCodeBench. For reasoning model, we sample 100 samples for each query while for non-reasoning models, we sample 256 samples. For reasoning models, 100 samples were used due to higher computational cost and our observation that performance trends stabilized at this point. Non-reasoning models, being less computationally intensive, used 256 samples for greater robustness.

PlanSearch This method prompts LLM to generate a number of observations and derived observations before making a final solution. We set number of generations to be three and number of derived observations to be two across all experiements.

PVGame Prover-Verifier Games involves two main phases: solution generation and solution verification. Solutions are generated in both "helpful" and "sneaky" modes, evaluated for correctness and clarity, and refined iteratively by leveraging the best-scored solutions from prior rounds to guide subsequent attempts. In our setup, we fix the number of solutions each round to be three, and scale the number of rounds from one to three.

Best of N Best of N samples N generations and each generation is evaluted via a judge. We use LLM as a judge and the judge template is shown in LLM evaluator prompt section below. We evaluate three times for each question and then take the mean as the final score.

Sequential Revisions We follow the implementation of (Snell et al., 2024). This method samples solution sequenially and each time it is revising from last solution except the first solution. The revision process involves first prompting feedback and then asking LLM to provide a revision based on those feedbacks. The feedback prompt template and the revision template first prompt is detailed in the LLM revision feedback prompt section and LLM revision prompt below respectively. Each solution is also evaluted using LLM as a judge. This is for choosing the final solution from the samples.

Parallel + Sequential Revisions We again follow the implementation of (Snell et al., 2024). This method samples multiple generations in the first step. For each geneneration, it sequentially revise similar to sequential revision independently. Same prompts are used from sequential revisions.

Reasoning Truncation We truncate the reasoning process (encapsulated in "⟨ think⟩" "⟨/think⟩" tokens) to control for the budget.

E. Templates

E.1. LLM evaluator prompt

LLM evaluator prompt

Question: **{question}**
Response: **{response}**

Analyze this answer strictly and critically. Identify and point out every flaw and imperfection to deduct the appropriate amount of points. Be very harsh and stringent in your assessment to ensure the grades are authoritative and reliable. Never award full marks. Assign a score between -100 and +100.

Response format:
[Analysis] ...
[Score] a single integer between -100 and +100.

E.2. LLM revision feedback prompt

LLM revsion feedback prompt

Since we have a weak Answer, could you provide me with a relection or feedback to correct this answer better? Analyze this Answer Strictly and Critic, point out every flaw for ervery possible imperfect to minus every possible score! Let's think step by step.

Question: **{question}**

Answer to analyze: **{previous response}**

E.3. LLM revision prompt

LLM revsion prompt

Please refine the your answer according to your Reflection or Feedback. The response should begin with [reasoning process]...[Verification]... and end with end with "[Final Answer] The answer is
boxed{answer}"
Let's think step by step.

Question: **{question}**

Previous solution: **{previous response}**

Feedback: **{feedback}**

F. More Inference Time Scaling Results

In this section, we present results for more models: Meta-Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, gpt-4o-mini, DeepSeek-R1-Distill-Qwen-14B (Figure 6).

G. Inference Budget

All runs are conducted on 8xA100 or through Together API.

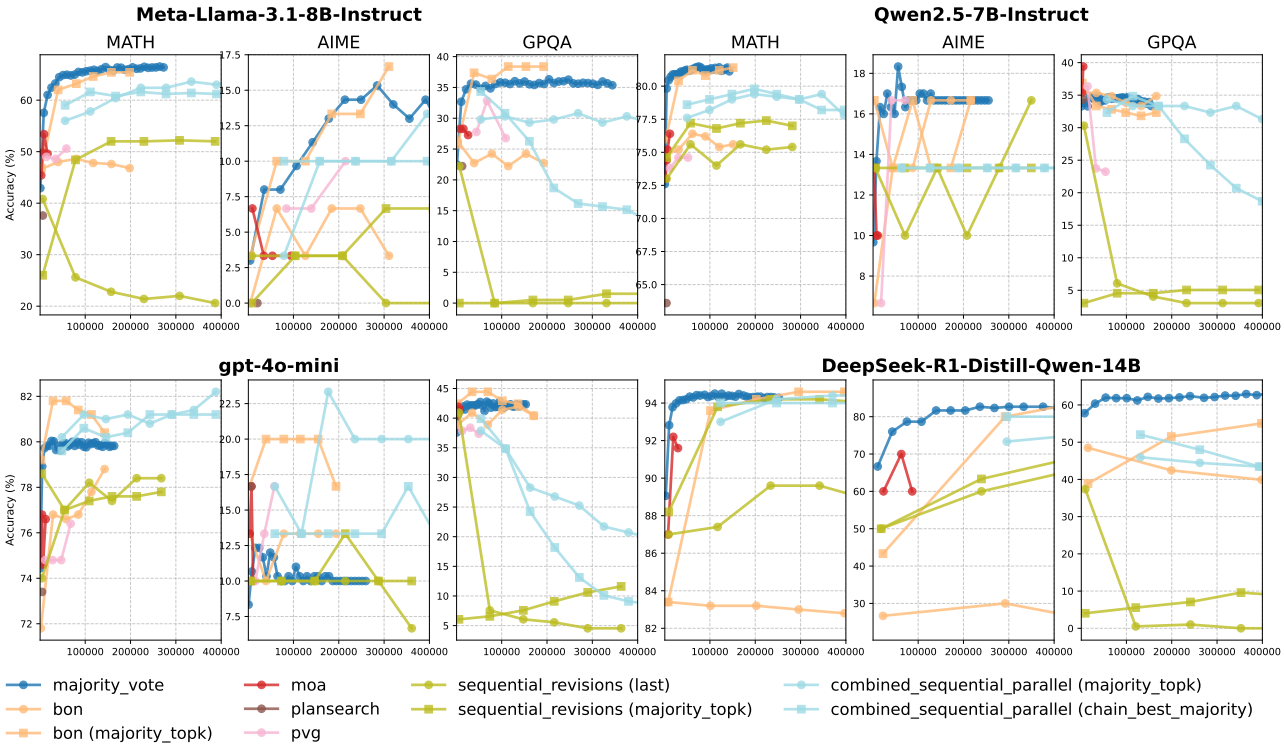

Figure 8. Performance of various inference-time scaling methods for four models across MATH, AIME and GPQA.