REASONING SHIFT: HOW CONTEXT SILENTLY SHORTENS LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) exhibiting test-time scaling behavior, such as extended reasoning traces and self-verification, have demonstrated remarkable performance on complex, long-term reasoning tasks. However, the robustness of these reasoning behaviors remains underexplored. To investigate this, we conduct a systematic evaluation of multiple reasoning models across three scenarios: (1) problems augmented with lengthy, irrelevant context; (2) multi-turn conversational settings with independent tasks; and (3) problems presented as a subtask within a complex task. We observe an interesting phenomenon: reasoning models tend to produce much shorter reasoning traces (up to 74%) for the same problem under different context conditions compared to the traces produced when the problem is presented in isolation. A finer-grained analysis reveals that this compression is associated with a decrease in self-verification and uncertainty management behaviors, such as double-checking. Importantly, we show that even when additional self-checks are forced, their efficiency depends not only on the content of the reasoning traces, but also on the presence of redundant context. We hope our findings draw additional attention to both the robustness of reasoning models and the problem of context management for LLMs.

1 INTRODUCTION

Recently, test-time scaling has emerged as a promising strategy for improving the performance of Large Language Models by allocating more compute during inference, allowing adaptation to input complexity without retraining (OpenAI, 2024; DeepSeek-AI et al., 2025). A prominent example of test-time scaling is Chain-of-Thought (CoT) (Kojima et al., 2022; Wei et al., 2022). With recent advancements in Reinforcement Learning (RL) and other post-training methods, LLMs are now equipped with a thinking mode, which enables them to produce long reasoning trajectories before providing an answer to the user's query. Importantly, modern advancements in reasoning LLMs are often tied to the models' ability to self-reflect on how they solve problems, as demonstrated by the presence of high-level patterns in their CoT, such as backtracking, uncertainty management, and self-verification (Gandhi et al., 2025; Venhoff et al., 2025). These patterns typically accompany increased reasoning trace length.

In parallel, recent years have seen the rise of long-context language models, with context windows expanding to hundreds of thousands or even millions of tokens (Dubey et al., 2024; Anthropic, 2024; Google DeepMind, 2025b). These advances provide ample space for test-time scaling, allowing models to explore, reflect, and summarize within a single context, thereby enabling complex reasoning (Liu et al., 2025a), multi-stage prover-verifier pipelines (Shao et al., 2025), and sophisticated agent workflows (Kimi Team, 2025). However, multiple works have demonstrated significant limitations associated with longer contexts, such as reduced ability to retrieve relevant data from long contexts (Kamradt, 2023), to learn in context (Li et al., 2024), to recover from wrong assumptions in multi-turn conversations (Laban et al., 2026), and to perform multi-step reasoning over long inputs (Ling et al., 2025).

With the rise of multiple test-time strategies and agents working on long-term tasks, natural questions arise: "How do context length and content affect the reasoning capabilities of the models?" In particular, "If a model faces an isolated subproblem with irrelevant data in the context, will it solve it similarly to working on it in isolation?"

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In this work, we study a surprising phenomenon: we observe a significant distribution shift in how models solve the same problems under different context conditions. We explore how reasoning quality and performance change under simple distracting conditions: (1) problems augmented with lengthy, irrelevant context; (2) multi-turn conversational settings with independent tasks; and (3) problems presented as subtasks within a complex task. We find that reasoning models tend to produce significantly fewer reasoning tokens when solving problems under non-isolated context conditions. Interestingly, we found that the part of the reasoning traces after finding the first candidate answer is disproportionately shortened compared to the time needed to find the first candidate answer. An analysis of the reasoning chains shows that this compression is associated with a decrease in self-verification and uncertainty management behaviors, such as double-checking. Importantly, we show that even when additional self-checks are forced, their efficiency depends not only on the content of the reasoning traces, but also on the presence of redundant context. Finally, we investigate several mitigation strategies and show that prompting and targeted supervised fine-tuning have limited effect on improving robustness to irrelevant context.

2 BACKGROUND

Chain-of-Thought reasoning. Test-time scaling has brought a paradigm shift that enables long Chain-of-Thought reasoning and induces sophisticated reasoning behaviors, making models superior in competitive math and coding tasks. The central technique driving this revolution is large-scale RL, which elicits complex reasoning behaviors such as self-verification and iterative refinement (OpenAI, 2024; DeepSeek-AI et al., 2025; Shao et al., 2024; Yu et al., 2025; Kumar et al., 2024). Subsequent works have explored additional approaches to achieve test-time scaling and strong reasoning performance (Muennighoff et al., 2025; Ye et al., 2025).

Additional thinking budget introduces new challenges in adjusting reasoning effort according to input problem complexity (Yang et al., 2025a). Su et al. (2025) analyzed the relationship between reasoning length and answer correctness, finding that LLMs tend to overthink on simpler problems and underthink on harder ones, indicating that models may fail to calibrate their reasoning length accordingly. Aggarwal et al. (2025) proposed a unified benchmark that jointly evaluates overthinking and underthinking in LLMs.

To analyze how LLMs tackle hard reasoning problems within their long CoT traces, Venhoff et al. (2025) define distinct reasoning functions within a reasoning trace and use an LLM judge to classify each sentence according to its functional role. Adopting their framework, Bogdan et al. (2025) propose analyzing long reasoning traces by identifying steps that guide the trajectory of reasoning, organizing sentences into different categories and measuring their causal impact, highlighting the importance of planning or uncertainty management sentences. Muennighoff et al. (2025) also demonstrated that enforcing self-reflection by intervening in the reasoning trace when the model attempts to stop may lead to improved performance. Subsequent works have investigated the relationship between accuracy and high-level characteristics of reasoning traces, such as length, review ratio, and others (Wu et al., 2025; Hassid et al., 2025; Feng et al., 2025).

Long context and context management. Recent years have witnessed remarkable growth in model context length (Dubey et al., 2024; Anthropic, 2024; Google DeepMind, 2025b). Despite this impressive scaling of context extension, a significant gap remains between the context length these models claim to support and the actual context length they can process effectively (Liu et al., 2025b). Common limitations typically involve retrieval-based evaluations (Kamradt, 2023) and multi-hop tracing and aggregation (Hsieh et al., 2024). However, Du et al. (2025) show that long contexts may degrade performance despite perfect retrieval, even in a synthetic setting where models are forced to attend only to relevant tokens. Laban et al. (2026) demonstrate significant performance drops in multi-turn settings using sharded simulation - a set of smaller instructions that collectively convey the same information as the original instruction. In this setting, models tend to over-rely on incorrect assumptions they made in earlier turns.

Multiple methods have been proposed to overcome the limitations of long contexts and unlock further scaling of test-time compute, including context compaction, summarization, and external memory modules (Anthropic, 2025; Liu et al., 2025a; Yan et al., 2025; Tekparmak & Ömer Kaya, 2025). A parallel line of work takes advantage of problems that can be split into isolated subproblems by delegating them to recursive self-calls, which may improve efficiency and naturally allow for maintaining compact context representations (Yang et al., 2025b; Jin et al., 2025; Ning et al., 2024; Zheng et al., 2025).

3 EXPERIMENTS

3.1 SETUP

This section presents experiments designed to answer a key question: Can a model solve an isolated subproblem as effectively when surrounded by irrelevant context as it does in isolation? This inquiry is motivated by two observations. First, complex reasoning tasks can often be decomposed into independent subtasks that do not require global context. Second, long-running agents increasingly operate within broad, general contexts that inevitably contain details irrelevant to specific user queries or subtasks.

To evaluate the model's ability to reason about the same problems under different context conditions, we compare the following setups:

- **Baseline**: a single user message containing a problem with a standard prompt.
- **Subtask**: a single user message containing two independent problems to solve.
- **Long Input**: a single user message containing a long chunk of irrelevant data (Shakespeare's plays from the Karpathy (2015)) followed by a problem with a standard prompt.
- **Multi-turn**: a multi-turn chat history, where each user message asks for solving a new problem with the Baseline prompt. We only evaluate the second turn.

These synthetic context conditions are controlled experimental probes designed to isolate and characterize the effect of redundant context. Please refer to Appendix B for the discussion on the setup design.

Table 1: Model performance on IMOAnswerBench and GPQA-Diamond. Accuracy and average number of generated reasoning tokens are shown. Background color represents the relative change from the baseline values.

Model	Baseline		Subtask		Long Input		Multi-turn	
	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens
IMOAnswerBench								
Qwen3.5-27B	74.5	28,771	62.4	20,165	67.8	16,415	67.0	17,404
Gemma 4 31B	72.7	9,240	52.8	5,635	67.0	6,252	71.0	7,530
gpt-oss-120b	73.8	24,180	64.0	17,408	64.0	11,876	69.3	19,831
Gemini 3 Flash Preview	82.8	23,090	67.0	13,653	80.3	19,879	82.5	21,693
Kimi K2 Thinking	74.8	29,615	65.0	19,630	70.8	23,380	72.8	30,421
GPQA-Diamond								
Qwen3.5-27B	85.1	12,364	83.6	7,279	82.1	6,939	83.8	4,722
Gemma 4 31B	84.1	3,638	79.8	2,219	82.6	2,047	78.5	1,353
gpt-oss-120b	78.3	10,762	64.0	7,466	57.6	3,711	72.5	4,476
Gemini 3 Flash Preview	89.6	12,927	89.4	7,918	89.6	10,922	88.7	3,334
Kimi K2 Thinking	81.6	11,121	70.7	7,635	78.0	7,619	73.2	4,669

For our main experiments, we evaluate the following models on IMOAnswerBench (Luong et al., 2025) and GPQA-Diamond (Rein et al., 2023): Qwen3.5-27B (QwenTeam, 2026), gpt-oss-120b (Agarwal et al., 2025), Gemma 4 31B (Google DeepMind, 2026b), Gemini 3 Flash Preview (Google DeepMind, 2025a), and Kimi K2 Thinking (Moonshot AI, 2025). For each model, we report both accuracy and the number of reasoning tokens generated. IMOAnswerBench consists of 400 Olympiad-level math problems with short verifiable answers (including mathematical expressions): we use Gemini 3.1 Pro Preview (Google DeepMind, 2026a) with the original prompt for Answer-AutoGrader(Luong et al., 2025) as a judge for automatic evaluation. For GPQA-Diamond, we report results averaged across 2 runs.

For Subtask scenario, model solves two independent problems within the same query. We report average accuracy of solving both problems and the number of reasoning tokens divided by two, representing the average number of reasoning tokens spent on a single task. Please refer to Appendix A

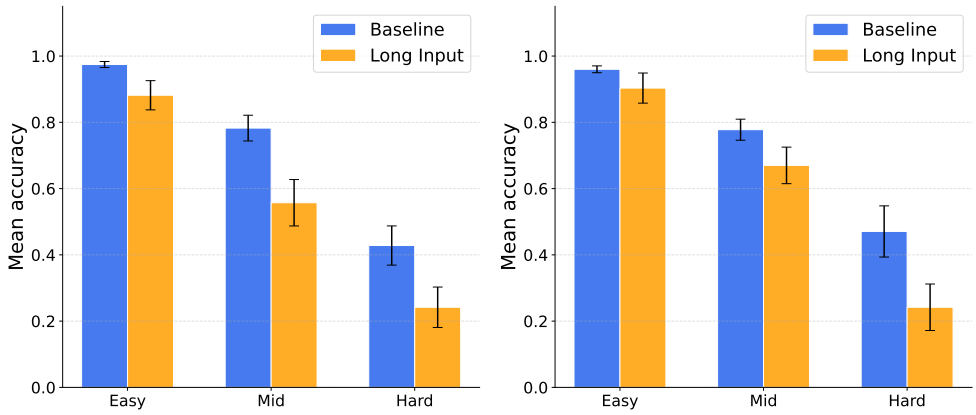


Figure 1: Accuracy comparison between Baseline and Long Input conditions across task difficulty levels (Easy, Mid, Hard) on a subset of IMOAnswerBench. **Left:** Gemma 4 31B, **right:** gpt-oss-120b.

for additional details on the experimental setup and to Appendix D for additional evaluation on code generation tasks.

3.2 Results

Results are presented in Table 1. Interestingly, we observe a slight performance drop in all Subtask and Long Input scenarios for IMOAnswerBench: degradation of 12% for Qwen3.5-27B, 9% for gpt-oss-120b, 15% for Gemini 3 Flash Preview and 9% for Kimi K2 Thinking. We report a detailed evaluation of the Subtask scenario in Appendix E.

Importantly, all models covered produce much shorter reasoning traces under different non-baseline context conditions for both benchmarks, generating up to 74% fewer reasoning tokens on average for the same problems ($p < 10^{-10}$ with paired Wilcoxon signed-rank test, for all models under the Long Input condition on IMOAnswerBench). Please see Figure 5 for task-wise comparison of the number of generated tokens.

Overall, we observe a significant distributional shift that causes symptoms such as shorter reasoning and degraded performance.

3.3 Additional Experiments

To further examine how task difficulty interacts with the observed performance degradation, we sampled 50 runs per task for 30 selected problems from IMOAnswerBench under both Baseline and Long Input conditions, grouping tasks into easy, medium, and hard buckets based on Baseline accuracy. We find that the accuracy drop under Long Input is disproportionately concentrated on

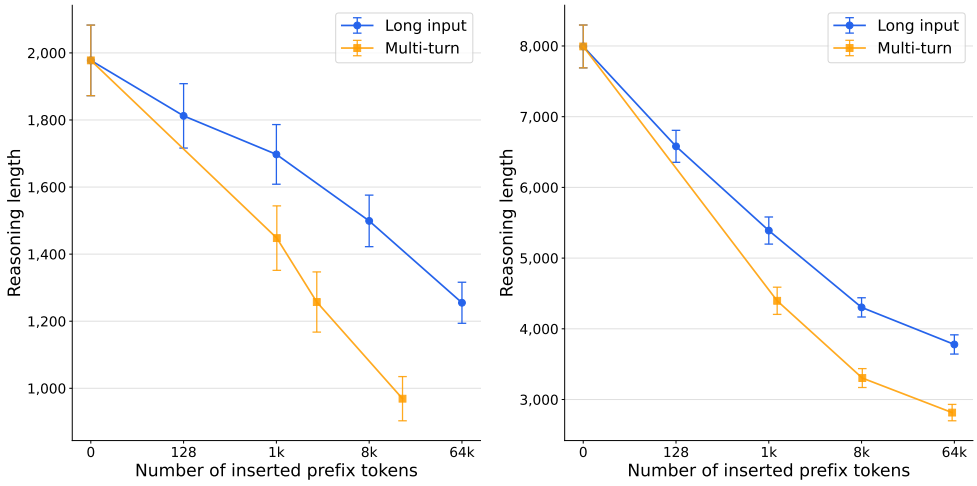
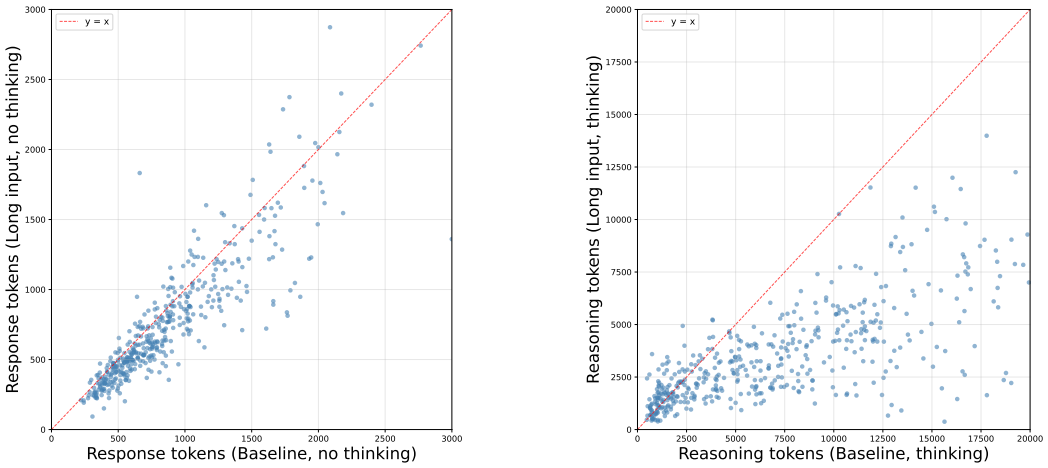


Figure 2: Average reasoning length on MATH-500 under varying number of inserted tokens in Long Input and Multi-turn setups. **Left:** Gemma 4 31B, **right:** Qwen3.5-27B.



(a) Number of response tokens, **no thinking** mode.

(b) Number of reasoning tokens, **thinking** mode.

Figure 3: Number of generated tokens for Qwen3.5-27B for each MATH-500 task. X-axis: Baseline, Y-Axis: Long Input.

medium and hard problems, see Figure 1. This suggests that the reasoning shift induced by additional context is relatively benign for straightforward problems but becomes increasingly costly as task difficulty rises.

Importantly, as shown in Figure 2, increasing the number of input context leads to a gradual and consistent reduction in reasoning length in both Long Input and Multi-turn scenarios. For example, for Qwen3.5-27B, even short distractions (hundreds of tokens) may be enough to reduce the average reasoning length by 18%, while further increasing the prompt size reduces reasoning by 50%. In the Multi-turn setup, adding more turns to the chat history also gradually reduces reasoning length.

Further, we additionally test whether the domain relevance of inserted prefix under Long Input scenario would affect the observed phenomenon. Results are presented in Appendix C.

3.4 Relation to thinking mode

We also conduct additional experiments to determine whether this effect is specific to reasoning mode. For this purpose, we evaluate Qwen3.5-27B, a model with integrated thinking and non-thinking modes, on the MATH-500 benchmark (Lightman et al., 2024) using the Long Input scenario (see Figure 3 for a task-wise comparison of the number of generated tokens). For non-thinking mode, we report the response length. Qwen3.5-27B demonstrates a 19% reduction in response length in non-thinking mode (1664 tokens for Baseline vs. 1348 tokens for Long Input) and a 53% reduction in reasoning length (8003 tokens for Baseline vs. 3762 tokens for Long Input). While both thinking and non-thinking modes exhibit statistical changes under different context conditions, we observe that the described phenomenon is markedly more pronounced in thinking mode.

Focusing on reasoning mode, we analyze checkpoints from different stages of post-training to examine how they affect reasoning shifts under varying context conditions. For this purpose, we

Table 2: Model performance with accuracy and average number of generated reasoning tokens, MATH-500. For Instruct model, the average number of response tokens is reported. Background color represents the relative change from the baseline values.

Model	**Baseline**		**Subtask**		**Long Input**		**Multi-turn**	
	Acc.	**Tokens**	**Acc.**	**Tokens**	**Acc.**	**Tokens**	**Acc.**	**Tokens**
Olmo-3-7B-Instruct	91.2	925	89.1	1,139	89.8	1,162	90.5	1,089
Olmo-3-7B-Think-SFT	94.5	2,671	90.2	1,829	91.4	1,908	89.1	1,611
Olmo-3-7B-Think-DPO	93.5	3,051	89.7	2,286	91.0	2,437	88.7	2,395
Olmo-3-7B-Think	95.2	3,664	92.1	2,851	93.2	2,859	92.2	2,788

evaluate different checkpoints of the Olmo3 model (Team Olmo, 2025). In short, we observe the described phenomenon across all reasoning checkpoints. The results are presented in Table 2.

4 ANALYSIS

Our experiments in Section 3 demonstrate a significant difference in the number of reasoning tokens generated for the same problems under different context conditions. In this section, we analyze the differences between these traces. Please refer to Appendix F for the details on experimental setup.

We inspect reasoning traces produced by Qwen3-32B, Qwen3.5-27B and Gemma 4 31B for MATH-500 problems. For clarity of presentation, we report here the results for Qwen3-32B, with references to the additional experiments with other models.

First, we manually inspect the traces to find any evidence of how the context might affect the reasoning - we did not find any indication that the model became confused by the query or failed to understand the task. Specifically, when investigating the reasoning traces produced by Qwen3-32B under Long Input prompts, we find only brief references to the irrelevant prompt part; we report several examples in Appendix H. These are all brief, dismissive acknowledgments: the model notices the prefix, labels it irrelevant, and moves on immediately. While additional context clearly affects the model's internal computations, we found no evidence that the inserted content influenced the model at a semantic level beyond simple acknowledgment.

4.1 DOES THE MODEL ARRIVE AT THE ANSWER FASTER?

We compare whether Baseline traces need more time to arrive at the answer or are longer because they exhibit stronger verification patterns after a candidate answer is found. For this purpose, we identify the position within each trace where the first candidate answer is stated. Results are reported in Table 14. Interestingly, while some differences are observed (-12.4% for Long Input and -5.5% for Multi-turn compared to Baseline), those differences are significantly lower than the average reduction of all reasoning traces, meaning that the parts of reasoning traces from the moment when first candidate answer is found until the end of the reasoning trace were reduced much more intensively (-36.1% for Long Input and -21.1% for Multi-turn compared to Baseline). Other models share the same trend.

4.2 WHAT IS DIFFERENT IN THE REASONING DYNAMICS?

To further investigate the nature of the significant differences in reasoning length, we adopt the framework from (Venhoff et al., 2025) to perform sentence-level analysis of long reasoning traces. We classify each sentence within the traces into the following categories: problem setup, plan generation, fact retrieval, active computation, result consolidation, uncertainty management, and final answer emission. We investigate the transition matrices of these labels within the reasoning traces. Figure 4 demonstrates the difference in transition heatmaps for the Baseline and Long Input setups. We note the absolute largest value in this heatmap: the transition from final answer emission to the end of the thinking trace (57% for Baseline vs. 68% for Long Input), which may indicate a significant behavioral difference: once the final answer is stated, Long Input traces finish more often, whereas Baseline traces have a greater probability of initiating additional self-checks.

Table 3: Resampling from the same reasoning traces but under varying numbers of inserted prompt tokens in the **Long Input** setup. The table presents the ratio of traces containing the end of reasoning or self-verification tokens.

Token	**0 (Baseline)**	**128**	**16k**
</think>	21%	26%	46%
Wait	11%	10%	5%
Alternatively	17%	11%	5%
But	46%	38%	20%
Maybe	23%	17%	9%

4.3 DOES ADDITIONAL CONTEXT SUPPRESS SELF-VERIFICATION?

The changes in self-checks frequency may be influenced by the content of the reasoning traces themselves rather than by the context conditions. To test this, we conduct a resampling experiment: we take the traces produced under Long Input and Multi-turn conditions, remove the final 64 tokens from the reasoning part, and resample the same number of tokens under different context conditions: Baseline, Long Input (128) (a condition similar to Long Input but with only 128 inserted tokens), Long Input (16k) and Multi-turn. We resample each trace with the same sampling parameters as the

original traces - recommended for reasoning tasks. For the resampled traces, we compare the ratio of finished traces. The results are presented in Table 3.

We observe an interesting phenomenon: when resampling from almost finished reasoning samples under different context conditions, all non-baseline context conditions have higher rates of finished reasoning traces and lower frequencies of words using during self-verification and uncertainty management. The ratio of finished traces for Long Input is 46%, compared to 21% for Baseline, which is also accompanied by reduced frequencies of words used during self-verification and uncertainty management, such as "wait," "alternatively," and "but." Please refer to Appendix I for the examples of the resampled samples. In Table 15 we additionally report the results of a similar resampling experiment for other models and Multi-turn setup, which demonstrates the same trend.

4.4 Does additional context affect verbalized self-confidence?

Recent studies demonstrate that language models use confidence to drive their behavior (Kumaran et al., 2026b;c), which raises a question: do different context conditions affect models' confidence in their reasoning? To test this we ask the model to verbalize the confidence in their partial solutions under different context conditions (similar to our resampling experiment). For confidence evaluation, we remove last 64 tokens of the reasoning trace and perform an intervention with a self-confidence prompt from Kumaran et al. (2026a) and use greedy decoding for generating the verbalized confidence score. Results are presented in Table 4. We observe that the same reasoning traces produced from both the Baseline and non-baseline setups receive lower confidence scores under the Baseline setup on average, which is consistent with the changes in rates of self-verification in our resampling experiment. In Appendix F, we report additional results on verbalized confidence, evaluations with different prompts and reasoning prefixes, all leading to similar conclusions. These results suggest that the ability of the model to estimate its confidence might not be robust to OOD context conditions.

4.5 Would the model benefit from longer thinking under context pressure?

Our results suggest that, for the same reasoning prefixes, different context conditions may suppress high-level reasoning patterns, such as self-verification and uncertainty management, which make reasoning traces significantly shorter. That raises a question: would the model be able to recover the Baseline-level accuracy if forced to think longer?

We took reasoning traces generated under the Long Input condition for IMOAnswerBench problems and, at the end of that reasoning, intervened with "Wait, let's check everything again." Similar to our resampling experiment, we considered interventions under different context conditions: Baseline and Long Input. Note that the interventions were applied to the same reasoning traces, which were produced under the Long Input setup. Results are presented in Table 5.

We can observe that, for both models considered, the "wait" intervention reduces the gap between the Long Input and Baseline conditions in terms of both accuracy and the number of reasoning tokens. Importantly, both models benefit from additional reasoning under both context conditions. However, for both models, the intervention under the Baseline context is more effective for the same reasoning traces: the models tend to review their solutions for longer and produce better final answers.

Overall, we can observe that additional context affects self-confidence reported by the model, suppresses the proportion of traces in which additional self-checks are initiated and limits the effectiveness of these self-verifications.

Table 4: The same reasoning traces receive higher self-confidence scores under non-baseline context conditions. Table represents the ratio of traces with the highest self-confidence scores. Rows represent how traces were generated, columns represent the context used during self-confidence evaluation. Qwen3-32B, MATH-500.

Trace	Baseline	Long Input(128)	Long Input(16k)	Multi-turn
Baseline	62.0%	68.0%	90.4%	78.4%
Long Input	41.2%	48.2%	85.4%	64.4%
Multi-turn	46.8%	55.5%	86.7%	70.5%

Table 5: Effect of a "Wait" intervention on IMOAnswerBench accuracy and average reasoning-token count. The same reasoning traces generated under Long Input are continued after the intervention with the irrelevant context either retained or removed.

	Long Input		"Wait" intervention (Long Input context)		"Wait" intervention (Baseline context)		Baseline	
Model	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens
Gemma 4 31B	67.0	6,252	68.3	8,058	71.8	10,301	72.7	9,240
Qwen3.5-27B	67.8	16,415	68.5	18,191	69.3	19,865	74.5	28,771

5 Mitigation approaches

In this section, we study potential approaches for mitigating the observed phenomenon as an additional diagnostic analysis. We discuss two methods: (1) prompting models to engage in more deliberate reasoning, and (2) fine-tuning models on a mixture of data that includes non-baseline context conditions.

5.1 Prompting

We first investigate whether the observed phenomenon can be mitigated through prompting alone. To test this, we evaluate multiple models under the same conditions as in the main experiment but with a prompt explicitly requesting more deliberate and thorough reasoning. Results are presented in Table 6. We experimented with several prompt formulations and report results for the best-performing one - the maximum reasoning effort prompt used for DeepSeek v4 (Xu et al., 2026), see Appendix A.

Overall, reasoning models show limited sensitivity to prompts requesting additional reasoning effort. In the baseline setting, the max effort prompt produces modestly longer reasoning traces (10–15% more tokens on average) with comparable or slightly improved accuracy. However, under non-baseline context conditions, the same prompt yields a similar relative increase in trace length while preserving the same shortening rate observed without the intervention. For example, gpt-oss-120b generates approximately 14% more tokens with the max effort prompt under the Long Input condition (13,494 vs. 11,876), yet this still represents a 51% reduction from the max effort baseline (27,525 tokens), nearly identical to the 51% reduction observed with the standard prompt (24,180 to 11,876). These results suggest that prompting alone is insufficient to counteract the reasoning shift induced by non-baseline context conditions.

5.2 Fine-tuning

We also explore a simple fine-tuning approach aimed at preventing models from reducing reasoning length when irrelevant information is present in the context. Our analysis in Section 4 indicates that the mechanisms responsible for self-verification are not robust to out-of-distribution inputs, although other explanations remain possible (e.g., general architectural limitations in processing long contexts). The fine-tuning experiment is intended to test whether the observed behavioral shift has a substantial distributional component.

Results in Table 2 suggest that the phenomenon may emerge during the early stages of reasoning-specific post-training. Motivated by this finding, we focus on mitigating the effect at the SFT stage. A straightforward approach would be to augment all SFT training samples with irrelevant context.

Table 6: Effect of max reasoning effort prompt on IMOAnswerBench.

		Baseline		**Long Input**	
Model	**Prompt**	**Acc.**	**Tokens**	**Acc.**	**Tokens**
Qwen3.5-27B	Standard	74.5	28,771	67.8	16,415
	Max effort	74.4	28,755	70.5	15,960
Gemma 4 31B	Standard	72.7	9,240	67.0	6,252
	Max effort	73.8	10,624	70.9	7,198
gpt-oss-120b	Standard	73.8	24,180	64.0	11,876
	Max effort	73.9	27,525	61.8	13,494

Table 7: Model performance with accuracy and average number of generated reasoning tokens under different context conditions on MATH-500.

Model	Baseline		Multi-turn(16)		Multi-turn(32)		Long Input(32)	
	Acc.	Tok.	Acc.	Tok.	Acc.	Tok.	Acc.	Tok.
Olmo-3-7B-Think-SFT	94.5	2,671	92.2	1,781	89.1	1,611	91.4	1,908
Olmo-3-7B-Think-SFT-Ours	96.1	2,869	94.5	2,674	89.8	2,606	93.8	2,806

Instead, we try to adapt an already SFT-trained reasoning model using a relatively small amount of additional data.

Training data. Starting from the `allenai/Dolci-Think-SFT-7B` dataset (Team Olmo, 2025), we augment training samples with additional irrelevant context. We use only 10% of the dataset for our experiment. For each sample, one of several augmentation strategies is selected uniformly at random: (1) keeping the original sample unchanged, (2) prepending a long irrelevant Shakespeare text fragment, (3) prepending a single irrelevant user-assistant interaction, or (4) prepending multiple irrelevant user-assistant turns. We limit the number of inserted irrelevant context to 16k tokens. For multi-turn augmentation, we use user queries from the `allenai/Dolci-Think-DPO-7B` dataset and generate assistant responses using Olmo-3-7B-Think-SFT, producing synthetic prompt-response pairs. Please refer to Appendix J for additional experiment details.

Evaluation. We evaluate the resulting model on MATH-500 under the Multi-turn(16), Multi-turn(32), and Long Input(32) scenarios, containing 16k, 32k, and 32k irrelevant context tokens respectively, to measure whether the reasoning shift effect is reduced after fine-tuning. Notably, the model is trained only with 16k-token augmentations.

Results. Our fine-tuned model demonstrates substantially improved robustness to irrelevant context in terms of reasoning length stability (Table 7). Across all evaluated scenarios, the number of generated reasoning tokens produced by our model remains close to the corresponding baseline values where no irrelevant context is present. However, we note that the performance of our model with irrelevant context still degrades compared to the Baseline values. These results suggest that while robustly imitating high-level patterns might be trivial, maintaining the effectiveness of long reasoning traces under additional context pressure is a more challenging task.

6 Discussion

Limitations We wish to highlight several important limitations of our work. First, while chosen on purpose to be easily controllable and easy to identify as redundant, our context conditions are relatively simple and synthetic: demonstrating and analyzing the described reasoning shift "in the wild" (using more realistic scenarios, including agentic ones) is of great interest for future work. Second, while we provide extensive empirical evidence that context conditions systematically shorten reasoning traces and suppress self-verification behaviors, our work does not offer a mechanistic account of how this effect arises within the model's internal representations. We leave a mechanistic investigation of this effect to future work, given the computational challenges of analyzing long inputs (Sharkey et al., 2025) and the generalization limitations of interpretability techniques across context lengths and domains (Kramár et al., 2026). Lastly, we do not cover or develop any context management methods, including those based on recursive self-calls (Yang et al., 2025b; Jin et al., 2025; Ning et al., 2024; Zheng et al., 2025; Zhang et al., 2025).

In this paper, we find that different context conditions may affect the way reasoning LLMs tackle the same problems. In particular, we demonstrate that the distribution of high-level behavioral patterns, such as uncertainty management and self-verification, is fragile and can be suppressed by non-relevant context in the prompt. The efficiency of these self-checks is also reduced by irrelevant context. While for easier problems this may reduce overthinking, such behavioral shifts degrade performance on more challenging tasks. We also show that prompting has a limited effect on mitigating the described distributional shift and targeted fine-tuning helps to maintain high-level patterns of reasoning traces while having limited effect on the accuracy.

We hope our findings draw additional attention to both the robustness of reasoning models and the problem of context management for LLMs and LLM-based agents.

AI USE STATEMENT

In this work, we used generative AI tools to assist in writing code for experiments and to polish the language and readability of author-written text. As part of the research methodology, we used Olmo-3-7B-Think-SFT to generate synthetic prompt-response pairs for training-data augmentation (Section 5.2), Gemini 3.1 Pro Preview to evaluate answer correctness (Section 3), and Gemini 3 Flash Preview to label reasoning-trace sentences for behavioral analysis (Section 4.2). We have not used generative AI tools to draft manuscript text from scratch or for other research tasks. We have reviewed all AI-assisted work: AI-assisted experimental code was reviewed and tested for correctness by humans, and writing edits were reviewed to ensure that they preserved the intended meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The paper focuses on analyzing reasoning behavior in LLMs, and does not introduce a deployed system, dataset with sensitive attributes, or application setting with direct societal use. As such, we do not expect the work itself to have broad immediate societal impacts.

REPRODUCIBILITY STATEMENT

We provide all the details needed to reproduce experimental results, including source code used to run main experiments, models, datasets and other experimental details.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Pranjal Aggarwal, Seungone Kim, Jack Lanchantin, Sean Welleck, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. Optimalthinkingbench: Evaluating over and underthinking in llms. *arXiv preprint arXiv:2508.13141*, 2025.

Anthropic. Introducing the next generation of claude, March 2024. URL https://www.anthropic.com/news/claude-3-family. Accessed: 2025-05-18.

Anthropic. Effective context engineering for ai agents, 2025. URL https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents.

P. C. Bogdan, U. Macar, N. Nanda, and A. Conmy. Thought anchors: Which LLM reasoning steps matter? *arXiv preprint arXiv:2506.19143*, jun 2025. Preprint.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, and Xiao Bi et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A Huerta, and Hao Peng. Context length alone hurts llm performance despite perfect retrieval. *arXiv preprint arXiv:2510.05381*, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yunzhen Feng, Julia Kempe, Cheng Zhang, Parag Jain, and Anthony Hartshorn. What characterizes effective reasoning? revisiting length, review, and structure of cot. *arXiv preprint arXiv:2509.19284*, 2025.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

Google DeepMind. Gemini 3 flash: frontier intelligence built for speed, 2025a. URL https://blog.google/products-and-platforms/products/gemini/gemini-3-flash/.

Google DeepMind. Gemini 2.5: Our Newest Gemini Model with Thinking. https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#gemini-2-5-thinking, 2025b. Accessed: 2025-04-07.

Google DeepMind. Gemini 3.1 pro: A smarter model for your most complex tasks, 2026a. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/.

Google DeepMind. Gemma 4, 2026b. URL https://deepmind.google/models/gemma/gemma-4/.

Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *arXiv preprint arXiv:2505.17813*, 2025.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024. URL https://arxiv.org/abs/2403.07974.

Tian Jin, Ellie Y. Cheng, Zack Ankner, Nikunj Saunshi, Blake M. Elias, Amir Yazdanbakhsh, Jonathan Ragan-Kelley, Suvinay Subramanian, and Michael Carbin. Learning to keep a promise: Scaling language model decoding parallelism with learned asynchronous decoding, 2025. URL https://arxiv.org/abs/2502.11517.

Greg Kamradt. Needle in a haystack - pressure testing llms. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023.

Andrej Karpathy. char-rnn. https://github.com/karpathy/char-rnn, 2015.

Kimi Team. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *ArXiv*, abs/2205.11916, 2022. URL https://api.semanticscholar.org/CorpusID:249017743.

János Kramár, Joshua Engels, Zheng Wang, Bilal Chughtai, Rohin Shah, Neel Nanda, and Arthur Conmy. Building production-ready probes for gemini. *arXiv preprint arXiv:2601.11516*, 2026.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

Dharshan Kumaran, Arthur Conmy, Federico Barbero, Simon Osindero, Viorica Patraucean, and Petar Velickovic. How do llms compute verbal confidence. *arXiv preprint arXiv:2603.17839*, 2026a.

Dharshan Kumaran, Nathaniel Daw, Simon Osindero, Petar Velickovic, and Viorica Patraucean. Causal evidence that language models use confidence to drive behavior. *arXiv preprint arXiv:2603.22161*, 2026b.

Dharshan Kumaran, Viorica Patraucean, Simon Osindero, Petar Velickovic, and Nathaniel Daw. How llms detect and correct their own errors: The role of internal confidence signals. *arXiv preprint arXiv:2604.22271*, 2026c.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VKGTGGcwl6.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024.

Zhenwen Liang, Linfeng Song, Yang Li, Tao Yang, Feng Zhang, Haitao Mi, and Dong Yu. Towards solving more challenging imo problems via decoupled reasoning and proving. *arXiv preprint arXiv:2507.06804*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The twelfth international conference on learning representations*, 2024.

Zhan Ling, Kang Liu, Kai Yan, Yifan Yang, Weijian Lin, Ting-Han Fan, Lingfeng Shen, Zhengyin Du, and Jiecao Chen. Longreason: A synthetic long-context reasoning benchmark via context expansion. *arXiv preprint arXiv:2501.15089*, 2025.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*, 2025b.

Minh-Thang Luong, Dawsen Hwang, Hoang H Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo, Junsu Kim, Garrett Bingham, Jonathan Lee, Swaroop Mishra, et al. Towards robust mathematical reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 35406–35430, 2025.

Moonshot AI. Introducing kimi k2 thinking, 2025. URL https://www.kimi.com/blog/kimi-k2-thinking.

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.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. Skeleton-of-thought: Prompting LLMs for efficient parallel generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mqVgBbNCm9.

OpenAI. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

QwenTeam. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Zhihong Shao, Yuxiang Luo, Chengda Lu, ZZ Ren, Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, and Xiaokang Zhang. Deepseekmath-v2: Towards self-verifiable mathematical reasoning. *arXiv preprint arXiv:2511.22570*, 2025.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, et al. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*, 2025.

Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms. *arXiv preprint arXiv:2505.00127*, 2025.

Team Olmo. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Atakan Tekparmak and Ömer Kaya. mem-agent: Equipping llm agents with memory using rl, October 2025. URL https://huggingface.co/blog/driaforall/mem-agent-blog. Community Article.

C. Venhoff, I. Arcuschin, P. Torr, A. Conmy, and N. Nanda. Understanding reasoning in thinking language models via steering vectors. In *Workshop on Reasoning and Planning for Large Language Models at ICLR 2025*, 2025. URL https://arxiv.org/abs/2506.18167.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Yuchen Yan, Yongliang Shen, Yang Liu, Jin Jiang, Mengdi Zhang, Jian Shao, and Yueting Zhuang. Inftythink: Breaking the length limits of long-context reasoning in large language models. *arXiv preprint arXiv:2503.06692*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Chenxiao Yang, Nathan Srebro, David McAllester, and Zhiyuan Li. Pencil: Long thoughts with short memory. *arXiv preprint arXiv:2503.14337*, 2025b.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning, 2025. URL https://arxiv.org/abs/2502.03387.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Alex L Zhang, Tim Kraska, and Omar Khattab. Recursive language models. *arXiv preprint arXiv:2512.24601*, 2025.

Tong Zheng, Hongming Zhang, Wenhao Yu, Xiaoyang Wang, Runpeng Dai, Rui Liu, Huiwen Bao, Chengsong Huang, Heng Huang, and Dong Yu. Parallel-r1: Towards parallel thinking via reinforcement learning, 2025. URL https://arxiv.org/abs/2509.07980.

A THE MAIN EXPERIMENT DETAILS

A.1 SOURCE CODE

To support reproducibility, we share anonymized code and evaluation scripts at the following link: https://anonymous.4open.science/r/reasoningshift/

A.2 PROMPT DETAILS

Baseline

Please reason step-by-step and put the final answer within \boxed{}.

<Problem>

Subtask

You will be given two independent problems, solve them separately. For each problem please reason step by step, and put your final answer within \boxed{}.

First Problem:
<Problem>

Second Problem:

<Problem>

Long Input

Old data:

<Shakespeare's plays>

Old data ends.

Please reason step-by-step and put the final answer within \boxed{}.

<Problem>

For Long Input setup, we insert 64000 tokens of Shakespeare's plays from the (Karpathy, 2015) to the prompt. We use unrelated text as a simple control to ensure that the model can easily recognize the additional context as irrelevant and do not spend their effort investigating the attached prefix.

Baseline, max reasoning effort

Reasoning Effort: Absolute maximum with no shortcuts permitted. You MUST be very thorough in your thinking and comprehensively decompose the problem to resolve the root cause, rigorously stress-testing your logic against all potential paths, edge cases, and adversarial scenarios. Explicitly write out your entire deliberation process, documenting every intermediate step, considered alternative, and rejected hypothesis to ensure absolutely no assumption is left unchecked. Put the final answer within \boxed{}.

<Problem>

A.3 Inference details

We use OpenRouter API for running the main experiments. To make sure that the results across setups are compatible, we use a single fixed vendor for each model. Here we list the providers and the sampling parameters used in our experiments:

- Qwen3.5-27B: Alibaba Cloud Int.;
- Gemma 4 31B: Parasail;
- gpt-oss-120b: Together AI; reasoning effort xhigh.
- Gemini 3 Flash Preview: Google AI Studio;
- Kimi K2 Thinking: Google Vertex;

For each model, we use default sampling parameters and thinking budget of 80,000 tokens across all context conditions. This compute budget is enough for most setups/models, except rare (less than 2% of generations) cases for Kimi K2 Thinking, we exclude unfinished generations from our analysis, keeping all side-by-side comparisons over the subset of problems where generations were finished naturally by the model.

We report the amount of reasoning tokens reported by the providers, which matches our estimations with manual tokenization where applicable.

For Olmo3 experiments, we use local inference with transformers library, using recommended sampling parameters: temperature 0.6, top_p: 0.95, max new tokens 16384.

A.4 Compute resources

Our main evaluation compute was spent on API-based inference for IMO AnswerBench, GPQA-Diamond, and LiveCodeBench. We accessed the evaluated models through OpenRouter. Across the main experiments, typical reasoning traces contain roughly 5k–30k generated tokens per example, depending on the benchmark, model, and prompting condition. Table 8 summarizes the API pricing used for cost estimation. As a concrete reference point, a full GPQA-Diamond run for the evaluated model set cost approximately $180; the total API cost of the main benchmark evaluations is therefore on the order of several hundred dollars.

A second source of compute was used for the analysis experiments, including the analysis and OLMo evaluations. These experiments were run on local GPU infrastructure. Since these runs include exploratory analyses, failed runs, and intermediate ablations that are not all reported in the paper, we report a conservative upper-bound estimate rather than an exact accounting. Overall, we estimate that these analysis and OLMo evaluation experiments required at most 1000 NVIDIA A100 GPU-hours.

Table 8: API pricing used for estimating evaluation costs. Prices are reported in USD per million tokens, following the OpenRouter model pages at the time of the experiments.

Model	Input	Output
Qwen3.5-27B	$0.195 / M	$1.56 / M
Gemma 4 31B	$0.14 / M	$0.40 / M
gpt-oss-120b	$0.039 / M	$0.18 / M
Gemini 3 Flash Preview	$0.50 / M	$3.00 / M
Kimi K2 Thinking	$0.60 / M	$2.50 / M

B Setup design

In this section, we discuss our motivation behind each of the three setups we consider for our main experiments. Our central research question is: "Can a model solve an isolated subproblem as effectively when surrounded by irrelevant context as it does in isolation?" Although one may ask why a model should operate with irrelevant context when a clean context is available, we believe each condition is worth investigating:

Subtask: In reasoning-intensive tasks, models may encounter independent subtasks that can be stated and solved without the preceding derivation. For example, solving an equation may not require the geometric context from which it was derived. Reproducing such situations exactly is difficult, so combining independent problems from the same domain provides a reasonable controlled approximation.

Multi-turn: Our setup complements Laban et al. (2026): they study related, underspecified conversations where models make early assumptions and fail to recover, whereas we test whether a self-contained problem is solved differently merely because it follows unrelated interactions. Although synthetic, this setup isolates reasoning shifts without confounding them with misleading assumptions or corrections.

Long Input: Long-running scientific and coding agents often accumulate documentation, tool outputs, intermediate results, and previous attempts that are not needed for the current subproblem. We deliberately use Shakespeare as a simple, conservative condition in which the additional text is clearly irrelevant. This minimizes the confound that a model might interpret a distractor as useful and spend additional reasoning trying to incorporate it.

C Long Input with relevant context

We additionally evaluated two models, gpt-oss-120b and Qwen3.5-27B, on IMOAnswerBench under the long-input condition, replacing the irrelevant context with IMO lemmas and their proofs from Liang et al. (2025). The results are presented in Table 9 and show a similar trend: both models generate fewer thinking tokens and achieve lower accuracy under long input, even when the context consists of domain-relevant mathematical content.

Table 9: Model performance on IMOAnswerBench under Long Input scenario with different content used as prefix.

	Baseline		Long Input (Shakespeare)		Long Input (IMO lemmas)	
Model	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens
gpt-oss-120b	73.8	24,180	64.0	11,876	63.5	11,055
Qwen3.5-27B	74.5	28,771	67.8	16,415	68.0	15,465

We note that we do not claim that these particular lemmas are directly relevant or useful for solving the problem in the query. Rather, they provide domain-relevant mathematical context. Determining whether context containing information directly useful for solving each specific problem produces the same effect would require a different experimental setup and is beyond the scope of our main research question, which is stated as follows: "Can a model solve an isolated subproblem as effectively when surrounded by irrelevant context as it does in isolation?"

D Additional experiment on code generation task

To further demonstrate that the observed reduction in reasoning length generalizes beyond mathematical reasoning tasks, we additionally evaluate reasoning models on code generation problems from the *LiveCodeBench* dataset (Jain et al., 2024). Specifically, we use the `code_generation_lite v6` subset, which contains 1055 competitive programming and algorithmic code generation tasks.

For Qwen3.5-27B, we use a maximum generation budget of 80k tokens together with a 32k-token input limit. For gpt-oss-120b, we use a 100k-token generation budget and a 28k-token input limit.

Here, we observe effects similar to those reported in Table 1. Both Qwen3.5-27B and gpt-oss-120b exhibit substantially shorter reasoning traces when irrelevant or additional contextual information is introduced into the prompt. In particular, Qwen3.5-27B demonstrates a dramatic reduction in reasoning length in the multi-turn setting, generating nearly $3.6\times$ fewer reasoning tokens compared to the baseline setup.

Table 10: Model performance with accuracy and average amount of generated reasoning tokens on LiveCodeBench. For the Subtask setup, accuracies are reported separately for the two generated subtasks. Background color represents the relative change from the baseline values.

Model	Baseline		Subtask			Long Input		Multi-turn	
	Acc.	Tokens	Acc_1	Acc_2	Tokens	Acc.	Tokens	Acc.	Tokens
Qwen3.5-27B	87	22,837	78	78	16,302	85	17,429	81	6,222
gpt-oss-120b	89	10,666	51	86	7,675	88	7,173	83	6,156

Interestingly, gpt-oss-120b also exhibits a noticeable degradation in performance in the Subtask setting, where the model is required to solve two tasks within a single response.

E Subtask evaluation details

To evaluate average performance in the **Subtask** scenario, we use two separate judge calls, evaluating the correctness of each subproblem. Results are presented in Table 11. We note that each problem appears once as a first subtask and once as a second subtask.

Table 11: Model accuracy on each subproblem in the **Subproblem** setup. Background color represents the relative change from the baseline values.

Model	Baseline	First subproblem	Second subproblem
Qwen3.5-27B	74.5	66.8	58.0
gpt-oss-120b	73.8	63.8	64.3
Gemini 3 Flash Preview	82.8	68.3	65.8
Kimi K2 Thinking	74.8	68.0	62.0

F Analysis experiments additional details

F.1 Verbalized confidence experiment

Table 12: Verbalized confidence experiment with last 64 tokens removed, using numeric scores. The same reasoning traces receive higher self-confidence scores under non-baseline context conditions. Table represents the ratio of traces with the highest self-confidence scores. Rows represent how traces were generated, columns represent the context used during self-confidence evaluation. Qwen3-32B, MATH-500.

Trace	Baseline	Long Input(128)	Long Input(16k)	Multi-turn
Baseline	85.9%	88.4%	90.4%	88.8%
Long Input	82.1%	87.1%	90.2%	86.7%
Multi-turn	83.0%	86.5%	89.0%	86.7%

To generate self-confidence scores, we reuse the prompt from Kumaran et al. (2026a). However, since we aim to estimate the confidence in unfinished reasoning trace, we inject the prompt as an intervention into the reasoning process, not as next turn of the conversation.

Prompt used to generate self-confidence scores

...Wait, let me estimate the confidence in my partial solution.
I will classify my confidence into one of the following classes based on how likely my answer is to be correct (NO REASONING OR EXPLANATION):

- "No chance" (0.0–0.1)

- "Really unlikely" (0.1–0.2)
- "Chances are slight" (0.2–0.3)
- "Unlikely" (0.3–0.4)
- "Less than even" (0.4–0.5)
- "Better than even" (0.5–0.6)
- "Likely" (0.6–0.7)
- "Very good chance" (0.7–0.8)
- "Highly likely" (0.8–0.9)
- "Almost certain" (0.9–1.0)

Each category reflects the probability that the answer is correct.
I will format my confidence as **Confidence**: CLASS where CLASS is one of the names above (without the probability ranges).
Okay, my confidence is **Confidence**:

To obtain more robust results, we also tried an alternative prompt where we ask the model to provide a numerical score from 0 to 99. We observe that the model typically selects between the two highest confidence classes (i.e., "Highly likely" and "Almost certain") in the categorical prompt, and between 98 and 99 in the numerical variant. Thus, we report the ratios of the maximum confidence scores. We did not perform calibration because our analysis focuses on statistical changes in confidence between different context conditions rather than on the absolute accuracy of the reported probabilities. Results for numeric evaluation are presented in Table 12. We observe a similar trend: for all reasoning traces, the proportion of problems assigned the maximum confidence score (99) was significantly lower under the baseline context condition than under each non-baseline condition; with $p < 0.001$ one-sided McNemar tests on paired per-task observations.

We additionally evaluate the confidence scores from shorter reasoning prefixes, keeping 50% of the reasoning trace before making a similar intervention. This allows us to assess whether context-induced confidence inflation manifests early in the reasoning process or only emerges after the model has had sufficient tokens to develop its solution. Results are presented in Table 13.

G TRANSITION HEATMAP

Figure 4 represents the difference of transition probability matrices. Positive values mean that Long Input setup has a higher probability of this transition than the Baseline. For this heatmap, we evaluate 500 traces for each context condition, which gives 110,867 transitions for Baseline and 80,508 transitions for Long Input. We evaluate a statistical significance of increased probability of transition from final answer emission to the end of thinking trace as $p = 0.0011$ using Mann-Whitney U test (one-sided (Long Input > Baseline)).

H LONG INPUT REFERENCES

In this section, we provide all references to the inserted prompt tokens in the **Long Input** setup made by Qwen3-32B on the MATH-500 benchmark. We note that other models may exhibit different

Table 13: Verbalized confidence experiment with the first half of the trace being evaluated. The same reasoning traces receive higher self-confidence scores under non-baseline context conditions. Table represents the ratio of traces with the highest self-confidence scores. Rows represent how traces were generated, columns represent the context used during self-confidence evaluation. Qwen3-32B, MATH-500.

Trace	Baseline	Long Input(128)	Long Input(16k)	Multi-turn
Baseline	38.0%	50.3%	57.8%	48.6%
Long Input	16.8%	28.3%	36.8%	26.8%
Multi-turn	22.5%	33.7%	44.8%	31.8%

Table 14: Average reasoning length and first candidate answer position across models and settings, MATH-500.

Model		Baseline	Long Input	Multi-turn
Qwen3-32B	Total reasoning tokens	3,421	2,375	2,825
	First candidate answer position	797	698	753
Qwen3.5-27B	Total reasoning tokens	4,689	2,545	3,017
	First candidate answer position	574	516	533
Gemma-4-31B	Total reasoning tokens	1,929	1,227	1,391
	First candidate answer position	656	638	649

Table 15: The ratio of finished samples during resampling experiment. Rows represent the source of the trace, column - the context condition during resampling. The same almost finished reasoning traces tend to finish earlier under non-baseline conditions. Multi-turn(4) represents the increased chat history - with 4 interactions.

Model		Baseline	Long Input	Multi-turn	Multi-turn(4)
Qwen3.5-27B	Baseline	12.2	14.8	18.8	28.8
	Long Input	3.1	11.0	11.3	14.1
	Multi-turn	5.3	5.7	11.1	20.1
Gemma-4-31B	Baseline	35.0	35.5	56.8	60.7
	Long Input	16.1	16.1	32.9	34.2
	Multi-turn	16.9	20.7	33.5	39.3

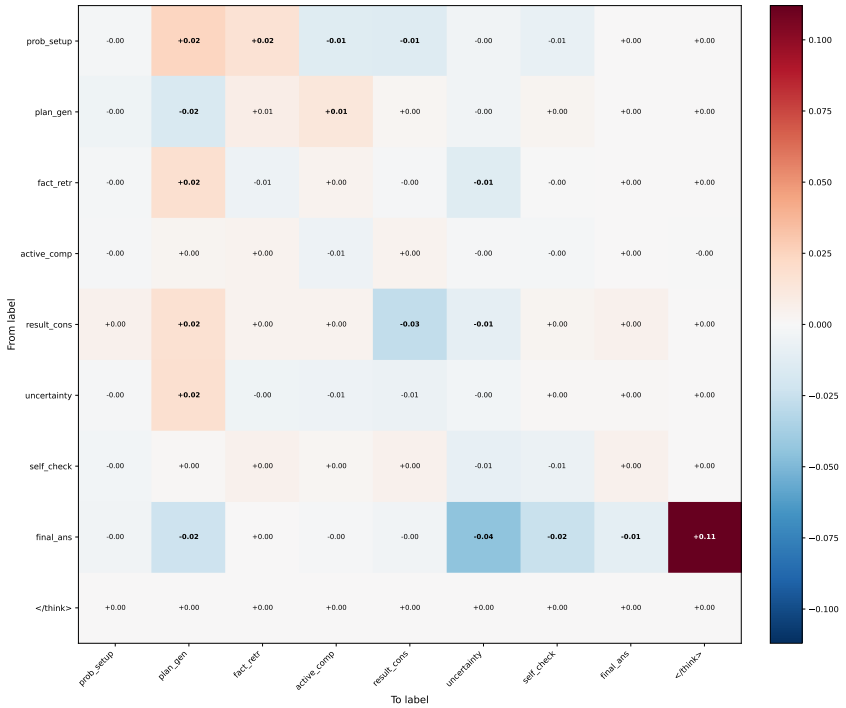


Figure 4: Difference of transition probability matrices (Long Input - Baseline). Qwen3-32B, MATH-500 problems.

patterns of prompt analysis. For example, Qwen3.5-27B tends to first outline the overall structure of the prompt before beginning to solve the task, typically identifying the distractor tokens early on and concluding that they are irrelevant to the user's request.

Task 203

<...>

Therefore, since f(-2) = 12, the remainder is 12. Therefore, the answer is 12.

But let me think if there's any trick here. The problem is from Shakespeare's *Coriolanus*? Wait, no, the original problem is just a math problem. The "old data" was the play's text, but the new task is a math problem. Maybe that was just to set the context? Anyway, regardless, the math seems straightforward.

Alternatively, perhaps there is a mistake in assuming that the exponents can be evaluated directly? <...>

Task 264

<...>

This suggests that the construction in the Asymptote code might have an error, or "bisectorpoint" is not what I think it is. Given the time I've spent and the complexity, I might need to switch strategies.

Let me recall that in the problem, after the battle, Coriolanus is made consul, and the citizens are angry, leading to his exile, but this is unrelated to the geometry problem. Focus on the geometry.

Given the square CDEG with CD = 3, and rectangle BEFH with BE = 5, find BH. <...>

Task 296

<...>

Therefore, equation B has the largest x. But let me check another angle.

Wait, the problem is from the SAT or similar? Wait, the original problem is from the Coriolanus play? No, the original data is the play, but the new task is a math problem. Anyway, according to my reasoning, equation B should be the answer. But let me check once more.

Wait, equation E: $base_E = 1 + 1/r$. For r approaching 0, this goes to infinity <...>

Task 420

<...> Alternatively, given that the answer requires boxed, maybe they want the scalar? Let me check the problem again.

Wait, in the initial problem statement, the user provided a lot of text from Shakespeare's Coriolanus, but the actual math problem is

separate. So maybe the problem is from a math textbook, and the user just included some old data. Anyway, focusing on the math problem.

Given that the problem says "projection of a onto b," and given that the dot product is given <...>

I EXAMPLES OF RESAMPLED TRACES

In this section, we provide several examples of resampled traces, described in Section 4.

Task 3

Baseline:
<...> That's 9 combinations, which correspond to the 9 divisors. So that's another way to see it. So that confirms it again.

Therefore, I think the answer is 9. But let me just check once more. <...>

Long Input (128 tokens inserted):
<...> That's 3*3=9 divisors, which again gives me 9. So that's consistent. Therefore, I think 9 is the right answer.

But just to be thorough, maybe I can check another way. <...>

Long Input (16k tokens inserted):
<...> That's nine combinations, so nine divisors. Yep, that checks out. So I think the answer is 9.

Final Answer The number of positive whole-number divisors of 196 is $\boxed{9}$. **</think>**

Task 6

Baseline:
<...> multiples of 3, 27 must be the smallest.

But let me think again. Let me check for example negative integers. <...>

Long Input (128 tokens inserted):
<...> multiples of 3, 27 is the smallest.

But wait, just to make sure, let me check if maybe 0 is considered. <...>

Long Input (16k tokens inserted):
<...> multiples of 3, 27 is the smallest.

Therefore, the answer is 27.

Final Answer
The smallest positive perfect cube that can be written as the sum of three consecutive integers is $\boxed{27}$. **</think>**

Table 16: GPQA-Diamond and MMLU-Pro performance of the original Olmo-3-7B-Think-SFT model and the fine-tuned variant. Tokens denote the average number of generated reasoning tokens.

Model	GPQA-Diamond		MMLU-Pro	
	Acc.	Tok.	Acc.	Tok.
Olmo-3-7B-Think-SFT	42.6	9,222	49.2	1779
Olmo-3-7B-Think-SFT-Ours	43.1	9,164	54.2	1912

J ADDITIONAL DETAILS ON FINE-TUNING

We base our experiments on the *open-instruct* and *OLMo-core* repositories (Team Olmo, 2025) (version `v2.5.0`).

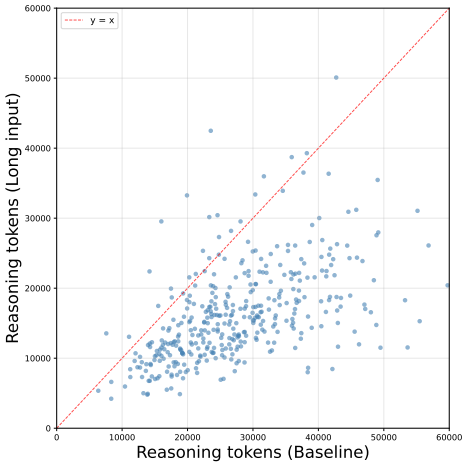
Training. In total, we fine-tune `Olmo-3-7B-Think-SFT` on an augmented 10% subset of the original `allenai/Dolci-Think-SFT-7B` training dataset for 7864 optimization steps, corresponding to 8.24B training tokens and two epochs, i.e., the same number of SFT epochs used in the original training of `Olmo-3-7B-Think-SFT`.

We fine-tune the model using a reduced learning rate of 10^{-6}, which is lower than the learning rates commonly used in OLMo-core SFT training. We intentionally use a conservative optimization setup to reduce the risk of drifting too far from the original checkpoint and degrading the model's overall reasoning quality. Following the recommendations provided in the OLMo-core repository, we set `global_batch_size` to 1048576 and `seq_len` to 32768. Training required approximately one day on 8×H100 GPUs.

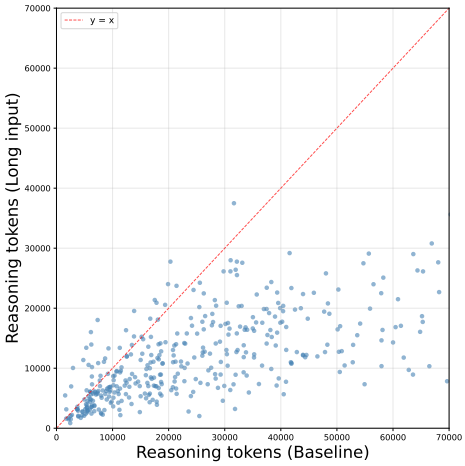
Evaluation. To verify that the additional training does not degrade the model's general reasoning capabilities, we evaluate the resulting model on the GPQA-Diamond benchmark, which consists of 198 PhD-level multiple-choice questions, as well as on MMLU-Pro (Wang et al., 2024) using 500 randomly sampled examples. We use the `Idavidrein/gpqa` and `TIGER-Lab/MMLU-Pro` benchmark versions from Hugging Face. For MMLU-Pro, the mean number of reasoning tokens is computed only over samples whose total sequence length is shorter than 32k tokens.

Table 16 shows that the proposed training procedure preserves both performance and reasoning length on GPQA-Diamond and MMLU-Pro. Additionally, the Baseline column in Table 7 demonstrates that performance on MATH-500 is preserved.

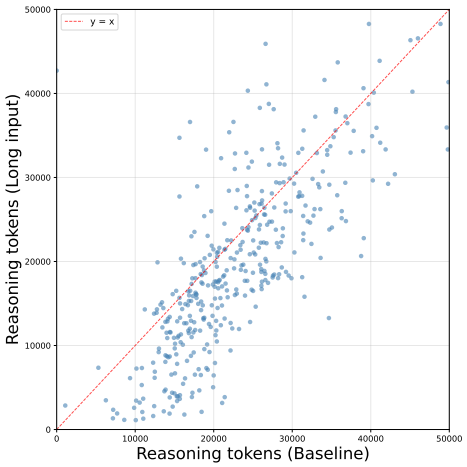
K TASK-WISE COMPARISON OF REASONING LENGTHS



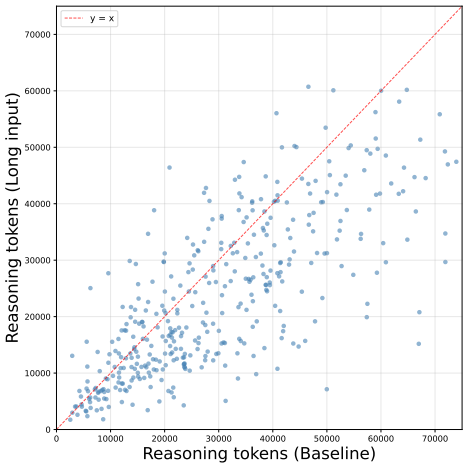
(a) Qwen3.5-27B



(b) GPT-OSS-120B



(c) Gemini 3 Flash Preview



(d) Kimi K2 Thinking

Figure 5: Number of generated tokens for each IMOAnswerBench task. X-axis: Baseline, Y-Axis: Long Input.