The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity

Parshin Shojaee*† Iman Mirzadeh* Keivan Alizadeh
Maxwell Horton Samy Bengio Mehrdad Farajtabar

Apple

Abstract

Recent generations of frontier language models have introduced Large Reasoning Models (LRMs) that generate detailed thinking processes before providing answers. While these models demonstrate improved performance on reasoning benchmarks, their fundamental capabilities, scaling properties, and limitations remain insufficiently understood. Current evaluations primarily focus on established mathematical and coding benchmarks, emphasizing final answer accuracy. However, this evaluation paradigm often suffers from data contamination and does not provide insights into the reasoning traces' structure and quality. In this work, we systematically investigate these gaps with the help of controllable puzzle environments that allow precise manipulation of compositional complexity while maintaining consistent logical structures. This setup enables the analysis of not only final answers but also the internal reasoning traces, offering insights into how LRMs "think". Through extensive experimentation across diverse puzzles, we show that frontier LRMs face a complete accuracy collapse beyond certain complexities. Moreover, they exhibit a counterintuitive scaling limit: their reasoning effort increases with problem complexity up to a point, then declines despite having an adequate token budget. By comparing LRMs with their standard LLM counterparts under equivalent inference compute, we identify three performance regimes: (1) low-complexity tasks where standard models surprisingly outperform LRMs, (2) medium-complexity tasks where additional thinking in LRMs demonstrates advantage, and (3) high-complexity tasks where both models experience complete collapse. We found that LRMs have limitations in exact computation: they fail to use explicit algorithms and reason inconsistently across puzzles. We also investigate the reasoning traces in more depth, studying the patterns of explored solutions and analyzing the models' computational behavior, shedding light on their strengths, limitations, and ultimately raising crucial questions about their true reasoning capabilities.

1 Introduction

Large Language Models (LLMs) have recently evolved to include specialized variants explicitly designed for reasoning tasks—Large Reasoning Models (LRMs) such as OpenAI's o1/o3 [1, 2], DeepSeek-R1 [3], Claude 3.7 Sonnet Thinking [4], and Gemini Thinking [5]. These models are new artifacts, characterized by their "*thinking*" mechanisms such as long Chain-of-Thought (CoT) with self-reflection, and have demonstrated promising results across various reasoning benchmarks. Their

*Equal contribution.

†Work done during an internship at Apple.

{p_shojaee, imirzadeh, kalizadehvahid, mchorton, bengio, farajtabar}@apple.com

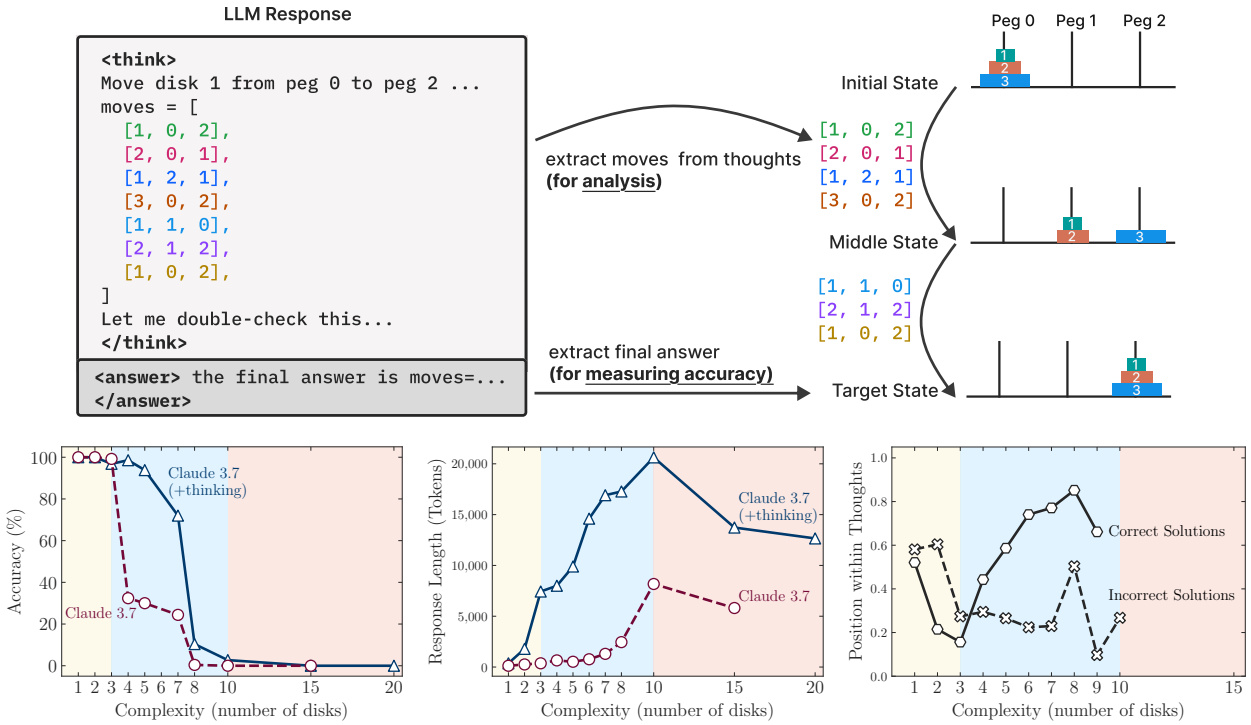


Figure 1: **Top**: Our setup enables verification of both final answers and intermediate reasoning traces, allowing detailed analysis of model thinking behavior. **Bottom left & middle**: At low complexity, non-thinking models are more accurate and token-efficient. As complexity increases, reasoning models outperform but require more tokens—until both collapse beyond a critical threshold, with shorter traces. **Bottom right**: For correctly solved cases, Claude 3.7 Thinking tends to find answers early at low complexity and later at higher complexity. In failed cases, it often fixates on an early wrong answer, wasting the remaining token budget. Both cases reveal inefficiencies in the reasoning process.

emergence suggests a potential paradigm shift in how LLM systems approach complex reasoning and problem-solving tasks, with some researchers proposing them as significant steps toward more general artificial intelligence capabilities.

Despite these claims and performance advancements, the fundamental benefits and limitations of LRMs remain insufficiently understood. Critical questions still persist: Are these models capable of generalizable reasoning, or are they leveraging different forms of pattern matching [6]? How does their performance scale with increasing problem complexity? How do they compare to their non-thinking standard LLM counterparts when provided with the same inference token compute? Most importantly, what are the inherent limitations of current reasoning approaches, and what improvements might be necessary to advance toward more robust reasoning capabilities?

We believe the lack of systematic analyses investigating these questions is due to limitations in current evaluation paradigms. Existing evaluations predominantly focus on established mathematical and coding benchmarks, which, while valuable, often suffer from data contamination issues and do not allow for controlled experimental conditions across different settings and complexities. Moreover, these evaluations do not provide insights into the structure and quality of reasoning traces. To understand the reasoning behavior of these models more rigorously, we need environments that enable controlled experimentation.

In this study, we probe the reasoning mechanisms of frontier LRMs through the lens of problem

complexity. Rather than standard benchmarks (e.g., math problems), we adopt controllable puzzle environments that let us vary complexity systematically—by adjusting puzzle elements while preserving the core logic—and inspect both solutions and internal reasoning (Fig. 1, top). These puzzles: (1) offer fine-grained control over complexity; (2) avoid contamination common in established benchmarks; (3) require only the explicitly provided rules, emphasizing algorithmic reasoning; and (4) support rigorous, simulator-based evaluation, enabling precise solution checks and detailed failure analyses. Our empirical investigation reveals several key findings about current Language Reasoning Models (LRMs): First, despite their sophisticated self-reflection mechanisms learned through reinforcement learning, these models fail to develop generalizable problem-solving capabilities for planning tasks, with performance collapsing to zero beyond a certain complexity threshold. Second, our comparison between LRMs and standard LLMs under equivalent inference compute reveals three distinct reasoning regimes (Fig. 1, bottom). For simpler, low-compositional problems, standard LLMs demonstrate greater efficiency and accuracy. As problem complexity moderately increases, thinking models gain an advantage. However, when problems reach high complexity with longer compositional depth, both model types experience complete performance collapse (Fig. 1, bottom left). Notably, near this collapse point, LRMs begin reducing their reasoning effort (measured by inference-time tokens) as problem complexity increases, despite operating well below generation length limits (Fig. 1, bottom middle). This suggests a fundamental inference time scaling limitation in LRMs' reasoning capabilities relative to problem complexity. Finally, our analysis of intermediate reasoning traces or thoughts reveals complexity-dependent patterns: In simpler problems, reasoning models often identify correct solutions early but inefficiently continue exploring incorrect alternatives—an "overthinking" phenomenon. At moderate complexity, correct solutions emerge only after extensive exploration of incorrect paths. Beyond a certain complexity threshold, models completely fail to find correct solutions (Fig. 1, bottom right). This indicates LRMs possess limited self-correction capabilities that, while valuable, reveal fundamental inefficiencies and clear scaling limitations.

These findings highlight both the strengths and limitations of existing LRMs, raising questions about the nature of reasoning in these systems with important implications for their design and deployment. Our key contributions are:

- We question the current evaluation paradigm of LRMs on established math benchmarks and design a controlled experimental testbed by leveraging algorithmic puzzle environments that enable controllable experimentation with respect to problem complexity.
- We show that state-of-the-art LRMs (e.g., o3-mini, DeepSeek-R1, Claude-3.7-Sonnet-Thinking) still fail to develop generalizable problem-solving capabilities, with accuracy ultimately collapsing to zero beyond certain complexities across different environments.
- We find that there exists a scaling limit in the LRMs' reasoning effort with respect to problem complexity, evidenced by the counterintuitive decreasing trend in the thinking tokens after a complexity point.
- We question the current evaluation paradigm based on final accuracy and extend our evaluation to intermediate solutions of thinking traces with the help of deterministic puzzle simulators. Our analysis reveals that as problem complexity increases, correct solutions systematically emerge at later positions in thinking compared to incorrect ones, providing quantitative insights into the self-correction mechanisms within LRMs.
- We uncover surprising limitations in LRMs' ability to perform exact computation, including their failure to benefit from explicit algorithms and their inconsistent reasoning across puzzle types.

2 Related Works

Reasoning in Language Models. Large Language Models (LLMs) undergo multiple costly training phases using vast amounts of training data. While these LLMs demonstrate promising language understanding with strong compression capabilities, their intelligence and reasoning abilities remain a critical topic of scientific debate [7, 8]. Earlier iterations of LLMs [9, 10, 11] exhibited poor performance on reasoning benchmarks [12, 13, 14, 6]. To address these shortcomings, several approaches have been explored with the common theme among them being *"scaling"* both the training data and test-time computation. For instance, generating a Chain of Thought (CoT) [15, 16, 17, 18] and incorporating self-verification [19, 20, 21] prior to the final answer have been shown to improve model performance. However, obtaining high-quality and scalable CoT data is quite expensive due to its scarcity. Another line of research focuses on compensating for the lack of supervised data by teaching models to think more effectively through supervised learning or reinforcement learning [22, 23, 24, 25, 26, 27]. A notable open-source example of these improvements is Deepseek-R1 [3], which demonstrated that applying RL with verifiable rewards can significantly enhance model performance, matching that of closed models like OpenAI's o1 [2], leading to a new generation of language models referred to as Large Reasoning Models (LRMs) such as Gemini flash thinking [5], Claude 3.7 Sonnet thinking [4], etc.

Understanding Large Reasoning Models. Recent studies have explored various aspects of reasoning behavior: Large Reasoning Models have shown emergent behaviors such as discrepancy between thought traces and final answers [28, 29] as well as efficiency concerns through what researchers term the *"overthinking phenomenon"* [30, 31, 32, 33], where models produce verbose, redundant outputs, even after finding the solution, creating significant inference computational overhead. In this work, we systematically analyze how much model thinks w.r.t task complexity. Recently, Ballon et al. [34] demonstrated that in newer LRMs accuracy generally declines when thinking increases in math problems, in contrast we observe when in controlled puzzle environment difficulty passes a certain level the model starts to think less and opposite corelation of thinking and task complexity only happens up to some threshold. Yue et al. [35] questioned whether reinforcement learning truly elicits novel reasoning patterns and shows pass@k of reasoning vs non-reasoning models converge to the same point. We also observe that in MATH-500 pass@k is close for reasoning versus non-reasoning models but we observed different patterns under medium and high complexity of puzzles, which is not easily observable on established math benchmarks used in common evaluations.

Controllable Evaluation Environments. Unlike earlier studies that focused on mathematical problems to evaluate the reasoning capabilities of language models, this work introduces controllable puzzle environments. These environments allow for precise manipulation of problem complexity while maintaining consistent logical processes, enabling a more rigorous analysis of reasoning patterns and limitations. Controllable environments are not uncommon in the literature [12, 36, 37]. However, our primary aim is not to propose a new benchmark; instead, we use these benchmarks as tools for designing experiments to understand the reasoning capabilities of language models. A closely related study by Valmeekam et al. [38] demonstrated that o1-models show significant performance improvements compared to previous models. Our work offers additional insights, such as examining pairs of thinking/non-thinking models (e.g., DeepSeek-R1/V3, Claude 3.7 Sonnet thinking/non-thinking). Furthermore, we study the reasoning traces of the LRMs in more depth, revealing different behaviors across various complexity levels.

Overall, the promising results from recent LRMs raise a critical question: how much have the previously reported limitations of LLMs been improved? In this work, we move beyond merely measuring the performance of these LRMs. We analyze how well these LRMs tackle problems of varying complexities and examine the properties of their reasoning processes.

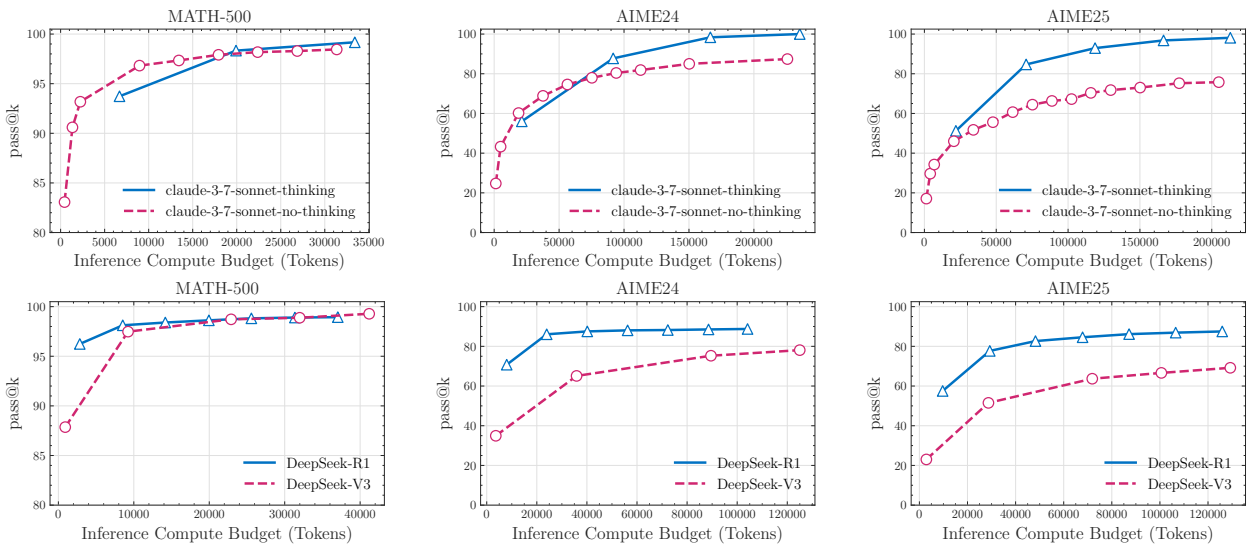


Figure 2: Comparative analysis of thinking versus non-thinking models across math benchmarks reveals inconsistent performance patterns. While results on the MATH-500 dataset show comparable performance between both model types, the thinking models demonstrate superior performance on AIME24 and AIME25 benchmarks. Additionally, the observed performance degradation from AIME24 to AIME25 highlights the vulnerability of these benchmarks to data contamination issues.

3 Math and Puzzle Environments

Currently, it is not clear whether the performance enhancements observed in recent RL-based thinking models are attributable to increased exposure to established mathematical benchmark data, to the significantly greater inference compute allocated to thinking tokens, or to reasoning capabilities developed by RL-based training? Recent studies [35, 39] have explored this question with established math benchmarks by comparing the upper-bound capabilities (pass@k) of RL-based thinking models with their non-thinking standard LLM counterparts. They have shown that under equivalent inference token budgets, non-thinking LLMs can eventually reach performance comparable to thinking models on benchmarks like MATH500 [40] and AIME24 [41]. We also conducted our comparative analysis of frontier LRMs like *Claude-3.7-Sonnet (with vs. without thinking)* and *DeepSeek (R1 vs. V3)*. Our results (shown in Fig. 2) confirm that, on the MATH500 dataset, the pass@k performance of thinking models is comparable to their non-thinking counterparts when provided with the same inference token budget. However, we observed that this performance gap widens on the AIME24 benchmark and widens further on AIME25. This widening gap presents an interpretive challenge. It could be attributed to either: (1) increasing complexity requiring more sophisticated reasoning processes, thus revealing genuine advantages of the thinking models for more complex problems, or (2) reduced data contamination in newer benchmarks (particularly AIME25). Interestingly, human performance on AIME25 was actually higher than on AIME24 [42, 43], suggesting that AIME25 might be less complex. Yet models perform worse on AIME25 than AIME24—potentially suggesting data contamination during the training of frontier LRMs. Given these non-justified observations and the fact that mathematical benchmarks do not allow for controlled manipulation of problem complexity, we turned to puzzle environments that enable more precise and systematic experimentation.

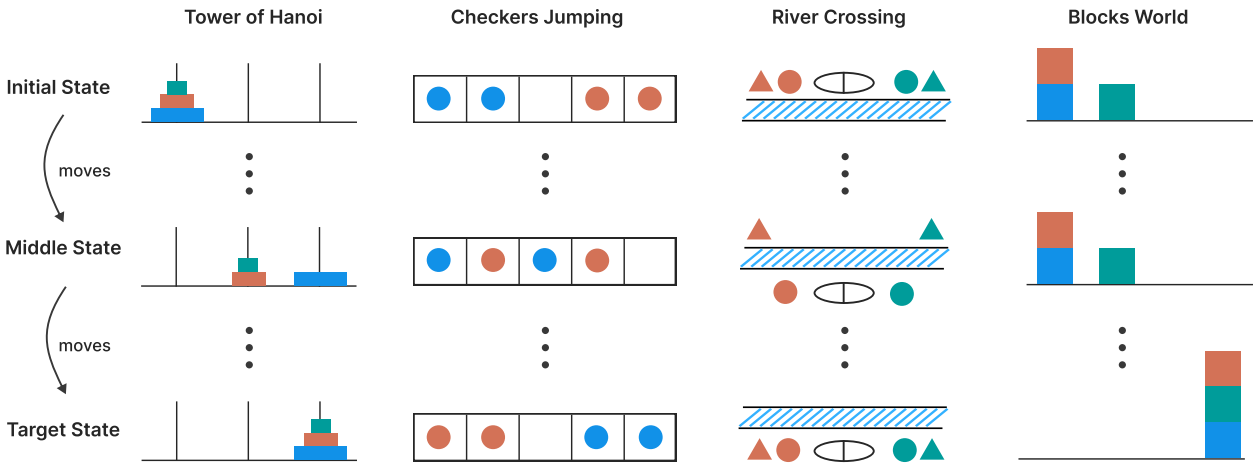


Figure 3: Illustration of the four puzzle environments. Columns show the progression from **initial state (top)** through **intermediate state (middle)** to **target state (bottom)** for puzzles: Tower of Hanoi (disk transfer across pegs), Checkers Jumping (position swapping of colored tokens), River Crossing (transporting entities across a river), and Blocks World (stack reconfiguration).

3.1 Puzzle Environments

We evaluate LRM reasoning on four controllable puzzles spanning compositional depth, planning complexity, and distributional settings. The puzzles are defined below and illustrated in Fig. 3.

Tower of Hanoi is a puzzle featuring three pegs and n disks of different sizes stacked on the first peg in size order (largest at bottom). The goal is to transfer all disks from the first peg to the third peg. Valid moves include moving only one disk at a time, taking only the top disk from a peg, and never placing a larger disk on top of a smaller one. The difficulty in this task can be controlled by the number of initial disks as the minimum number of required moves with n initial disks will be $2^n - 1$. However, in this work we do not grade for optimality of final solution and only measuring the correctness of each move and reaching the target state.

Checker Jumping is a one-dimensional puzzle arranging red checkers, blue checkers, and a single empty space in a line. The objective is to swap the positions of all red and blue checkers, effectively mirroring the initial configuration. Valid moves include sliding a checker into an adjacent empty space or jumping over exactly one checker of the opposite color to land in an empty space. No checker can move backward in the puzzle process. The complexity of this task can be controlled by the number of checkers: with $2n$ checkers, the minimum number of moves required will be $(n+1)^2 - 1$.

River Crossing is a constraint satisfaction planning puzzle involving n actors and their corresponding n agents who must cross a river using a boat. The goal is to transport all $2n$ individuals from the left bank to the right bank. The boat can carry at most k individuals and cannot travel empty. Invalid situations arise when an actor is in the presence of another agent without their own agent present, as each agent must protect their client from competing agents. The complexity of this task can also be controlled by the number of actor/agent pairs present. For $n = 2, n = 3$ pairs, we use boat capacity of $k = 2$ and for larger number of pairs we use $k = 3$.

Blocks World is a block-stacking puzzle requiring rearrangement of blocks from an initial configuration into a specified goal configuration. The objective is to find the minimum number of moves needed for this transformation. Valid moves are restricted to the topmost block of any stack, which can be placed either on an empty stack or on top of another block. The complexity in this task can be controlled by the number of blocks present.

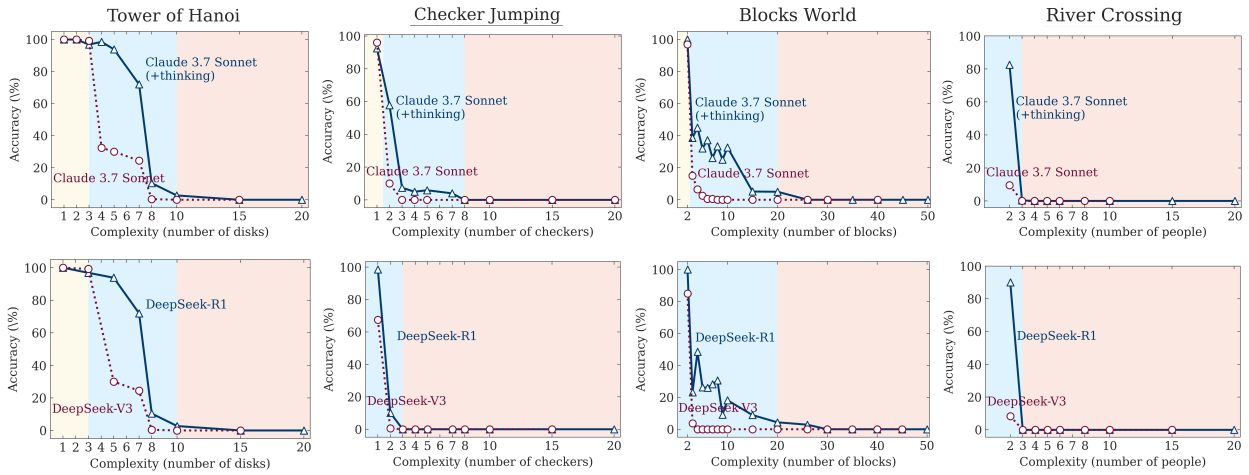


Figure 4: Accuracy of thinking models (Claude 3.7 Sonnet with thinking, DeepSeek-R1) versus their non-thinking counterparts (Claude 3.7 Sonnet, DeepSeek-V3) across all puzzle environments and varying levels of problem complexity.

4 Experiments & Results

4.1 Experimental Setup

Most of our experiments are conducted on reasoning models and their non-thinking counterparts, such as Claude 3.7 Sonnet (thinking/non-thinking) and DeepSeek-R1/V3. We chose these models because they allow access to the thinking tokens, unlike models such as OpenAI's o-series. For experiments focused solely on final accuracy, we also report results on the o-series models. For Claude 3.7 Sonnet models, we allow the maximum token budget (64k). Similarly, for DeepSeek-R1/V3 models on local servers, we allow the maximum length to be up to 64k tokens. For each puzzle instance, we generate 25 samples and report the average performance of each model across them. Comprehensive details of our experimental setup and results are provided in the Appendix.

4.2 How Does Complexity Affect Reasoning?

4.2.1 Three Regimes of Complexity

Motivated by the observations in Fig. 2, to systematically investigate the impact of problem complexity on reasoning behavior, we conducted experiments comparing **thinking** and **non-thinking** model pairs across our controlled puzzle environments. Our analysis focused on matched pairs of LLMs with identical model backbones, specifically *Claude-3.7-Sonnet (w. vs. w/o thinking)* and *DeepSeek (R1 vs. V3)*. In each puzzle, we vary the complexity by manipulating problem size N (representing disk count, checker count, block count, or crossing elements).

Fig. 4 presents the accuracy of both model types as a function of problem complexity across all puzzle environments. Complementing this, Fig. 5 shows the upper bound performance capabilities (pass@k) of these model pairs under equivalent inference token compute (averaged across all puzzles), extending earlier analyses from mathematical benchmarks (Fig. 2) to the controlled puzzle environments. Results from both these figures demonstrate that, unlike observations from math, there exists *three regimes* in the behavior of these models with respect to complexity. In the first regime where problem complexity is low, we observe that non-thinking models are capable to obtain performance comparable to, or even better than thinking models with more token-efficient inference. In the

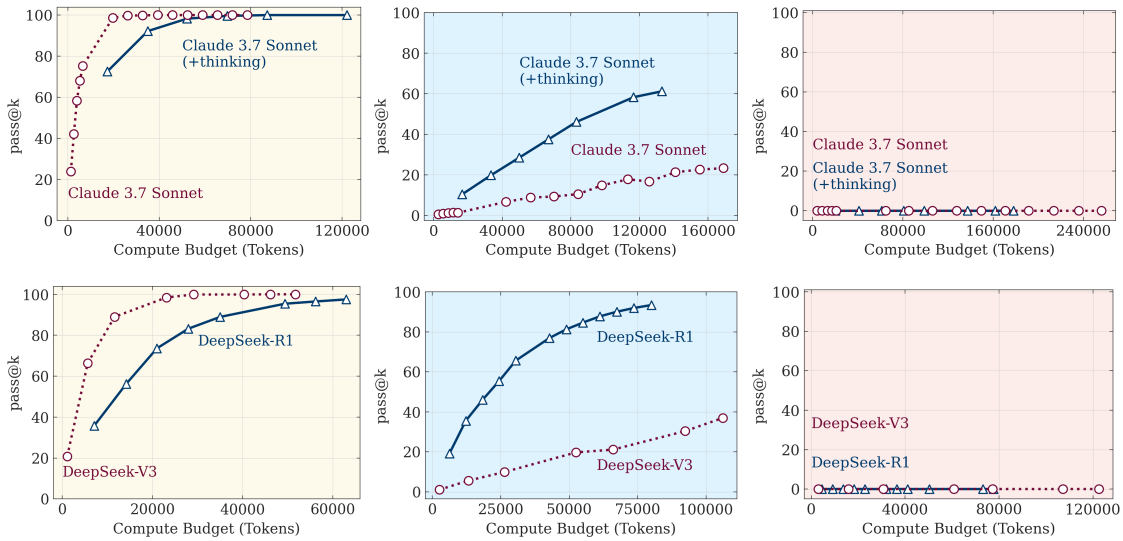


Figure 5: Pass@k performance of thinking vs. non-thinking models across equivalent compute budgets in puzzle environments of low, medium, and high complexity. Non-thinking models excel in simple problems, thinking models show advantages at medium complexity, while both approaches fail at high complexity regardless of compute allocation.

second regime with medium complexity, the advantage of reasoning models capable of generating long chain-of-thought begin to manifest, and the performance gap between model pairs increases. The most interesting regime is the third regime where problem complexity is higher and the performance of both models have collapsed to zero. Results show that while thinking models delay this collapse, they also ultimately encounter the same fundamental limitations as their non-thinking counterparts.

4.2.2 Collapse of Reasoning Models

We next examine how different specialized reasoning models equipped with thinking tokens respond to increasing problem complexity. Our experiments evaluate five state-of-the-art thinking models: *o3-mini* (medium and high configurations), *DeepSeek-R1*, *DeepSeek-R1-Qwen-32B*, and *Claude-3.7-Sonnet (thinking)*. Fig. 6 demonstrates these models' performance in terms of accuracy (top) and thinking token usage (bottom) across varying complexity levels. Results show that all reasoning models exhibit a similar pattern with respect to complexity: accuracy progressively declines as problem complexity increases until reaching complete collapse (zero accuracy) beyond a model-specific complexity threshold. Analysis of inference thinking token compute also reveals an intriguing pattern in thinking token allocation learned by these models. We observe that reasoning models initially increase their thinking tokens proportionally with problem complexity. However, upon approaching a critical threshold—which closely corresponds to their accuracy collapse point—models counterintuitively begin to reduce their reasoning effort despite increasing problem difficulty. This phenomenon is most pronounced in o3-mini variants and less severe in the Claude-3.7-Sonnet (thinking) model. Notably, despite operating well below their generation length limits with ample inference budget available, these models fail to take advantage of additional inference compute during the thinking phase as problems become more complex. This behavior suggests a fundamental scaling limitation in the thinking capabilities of current reasoning models relative to problem complexity.

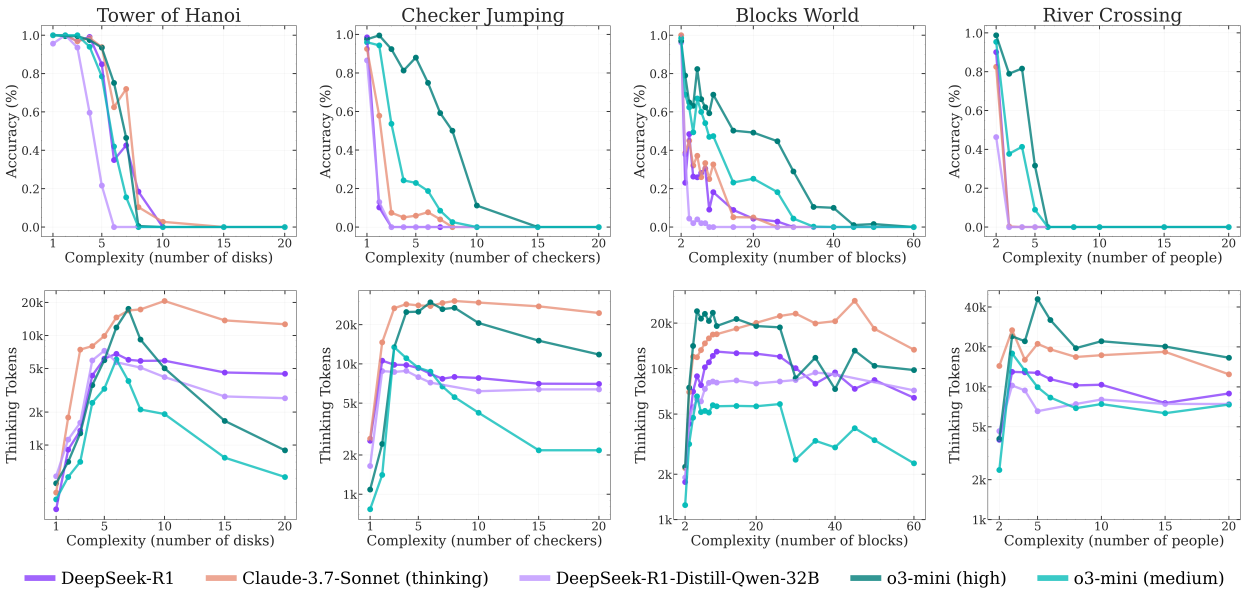
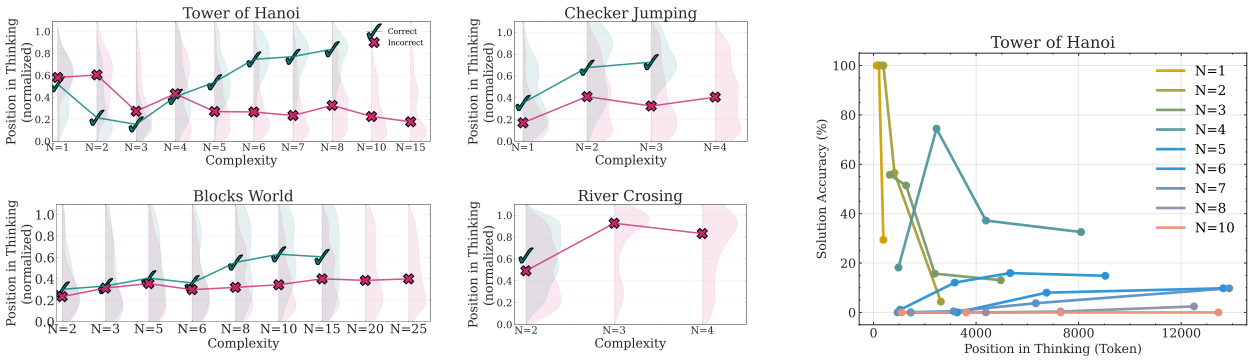


Figure 6: Accuracy and thinking tokens vs. problem complexity for reasoning models across puzzle environments. As complexity increases, reasoning models initially spend more tokens while accuracy declines gradually, until a critical point where reasoning collapses—performance drops sharply and reasoning effort decreases.

4.3 What Happens Inside the Thoughts of Reasoning Models?

To gain deeper insights into the thinking processes of reasoning models, we conducted a fine-grained analysis of their reasoning traces. As shown in Fig. 1, our setup with puzzle environments allows us to look beyond final answer and obtain more detailed insight into the reasoning traces ("thoughts") produced by these models. We extract and analyze the intermediate solutions explored *within the thoughts* of a model with the help of puzzle simulators. Our investigation examines the patterns and characteristics of these intermediate solutions, their correctness relative to their sequential position in the reasoning process, and how these patterns evolve with increasing problem complexity. For this analysis, we focus on the reasoning traces generated by *Claude-3.7-Sonnet-Thinking* across our puzzle suite. For each intermediate solution identified within the traces, we recorded: (1) its relative position within the reasoning trace (normalized by total thought length), (2) its correctness as validated by our puzzle simulators, and (3) the complexity of the corresponding problem. This allows to characterize the progression and accuracy of solution development throughout the reasoning process.

Fig. 7a demonstrates the relation between the position of intermediate solutions within thoughts, their correctness, and problem complexity across all puzzle environments. Our analysis from reasoning traces also further validates three regimes of complexity discussed above. For simpler problems, reasoning models often find the correct solution early in their thinking but then continue exploring incorrect solutions. Note the distribution of incorrect solutions (red) is shifted more upward towards end of thinking compared to correct solutions (green). This phenomenon, referred to as "overthinking" in the literature, leads to the waste of compute. As problems become moderately more complex, this trend reverses: models first explore incorrect solutions and mostly later in thought arrive at the correct ones. This time the distribution of incorrect solutions (red) is shifted more downward compared to correct ones (green). Finally, for the problems with higher complexity, collapse emerges,



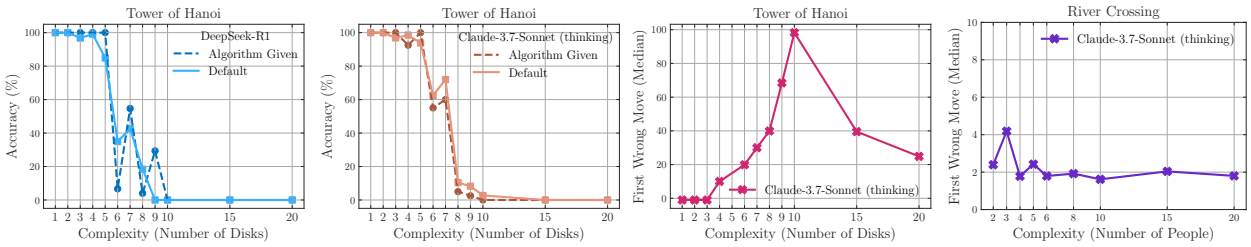
(a) (b)

Figure 7: **Left & Middle:** Position and correctness of intermediate solutions within reasoning traces across four puzzles at varying complexity levels. ✓ indicates correct solutions, ✗ indicates incorrect solutions, with distribution density shown by shading; **Right:** Solution accuracy versus position in thinking for Tower of Hanoi at different complexity levels. Simple problems (N=1-3) show early accuracy declining over time (overthinking), moderate problems (N=4-7) show slight improvement in accuracy with continued reasoning, and complex problems (N≥8) exhibit consistently near-zero accuracy, indicating complete reasoning failure.

meaning that the model fails to generate any correct solutions within the thought.
Fig. 7b presents a complementary analysis of solution accuracy within sequential segments (bins) of the thoughts in the Tower of Hanoi environment. It can be observed that for simpler problems (smaller N), solution accuracy tends to decrease or oscillate as thinking progresses, providing further evidence of the overthinking phenomenon. However, this trend changes for more complex problems, where solution accuracy increases with thinking progression—up to a certain threshold. Beyond this complexity threshold, in the "collapse mode", accuracy is zero.

4.4 Open Questions: Puzzling Behavior of Reasoning Models

In this section, we present surprising results concerning the limitations of reasoning models in executing exact problem-solving steps, as well as demonstrating different behaviors of the models based on the number of moves.
As shown in Figures 8a and 8b, in the Tower of Hanoi environment, even when we provide the algorithm in the prompt—so that the model only needs to execute the prescribed steps—performance does not improve, and the observed collapse still occurs at roughly the same point. This is noteworthy because finding and devising a solution should require substantially more computation (e.g., for search and verification) than merely executing a given algorithm. This further highlights the limitations of reasoning models in verification and in following logical steps to solve a problem, suggesting that further research is needed to understand the symbolic manipulation capabilities of such models [44, 6].
Moreover, in Figures 8c and 8d, we observe very different behavior from the Claude 3.7 Sonnet thinking model. In the Tower of Hanoi environment, the model's first error in the proposed solution often occurs much later, e.g., around move 100 for (N=10), compared to the River Crossing environment, where the model can only produce a valid solution until move 4. Note that this model also achieves near-perfect accuracy when solving the Tower of Hanoi with (N=5), which requires 31 moves, while it fails to solve the River Crossing puzzle when (N=3), which has a solution of 11 moves. This likely suggests that examples of River Crossing with N>2 are scarce on the web, meaning LRMs may not have frequently encountered or memorized such instances during training.



(a) (b) (c) (d)

Figure 8: **(a) & (b)** Despite providing the solution algorithm in the prompt, execution failure occurs at similar points, highlighting reasoning model limitations in logical step execution. **(c) & (d)** Notably, the Claude 3.7 Sonnet model demonstrates much longer error-free sequences in the Tower of Hanoi compared to early errors in the River Crossing scenario.

5 Conclusion

In this paper, we systematically examine frontier Large Reasoning Models (LRMs) through the lens of problem complexity using controllable puzzle environments. Our findings reveal fundamental limitations in current models: despite sophisticated self-reflection mechanisms, these models fail to develop generalizable reasoning capabilities beyond certain complexity thresholds. We identified three distinct reasoning regimes: standard LLMs outperform LRMs at low complexity, LRMs excel at moderate complexity, and both collapse at high complexity. Particularly concerning is the counterintuitive reduction in reasoning effort as problems approach critical complexity, suggesting an inherent compute scaling limit in LRMs. Our detailed analysis of reasoning traces further exposed complexity-dependent reasoning patterns, from inefficient "overthinking" on simpler problems to complete failure on complex ones. These insights challenge prevailing assumptions about LRM capabilities and suggest that current approaches may be encountering fundamental barriers to generalizable reasoning. Finally, we presented some surprising results on LRMs that lead to several open questions for future work. Most notably, we observed their limitations in performing exact computation; for example, when we provided the solution algorithm for the Tower of Hanoi to the models, their performance on this puzzle did not improve. Moreover, investigating the first failure move of the models revealed surprising behaviors. For instance, they could perform up to 100 correct moves in the Tower of Hanoi but fail to provide more than 5 correct moves in the River Crossing puzzle. We believe our results can pave the way for future investigations into the reasoning capabilities of these systems.

Limitations

We acknowledge that our work has limitations. While our puzzle environments enable controlled experimentation with fine-grained control over problem complexity, they represent a narrow slice of reasoning tasks and may not capture the diversity of real-world or knowledge-intensive reasoning problems. It is notable that most of our experiments rely on black-box API access to the closed frontier LRMs, limiting our ability to analyze internal states or architectural components. Furthermore, the use of deterministic puzzle simulators assumes that reasoning can be perfectly validated step by step. However, in less structured domains, such precise validation may not be feasible, limiting the transferability of this analysis to other more generalizable reasoning.