What Recent Research on Large Reasoning Models Reveals About AI Limitations and Computing Education*

Chris Alvin[1], Lori Alvin[2]
[1]Computer Science Department
[2]Mathematics Department
Furman University
Greenville, SC 29613
{ chris.alvin†, lori.alvin }@furman.edu

Abstract

Recent developments in Large Reasoning Models (LRMs) such as OpenAI's o1/o3 series and Claude Thinking have generated some considerable interest from the educational community. However, new research reveals some fundamental limitations in the reasoning capabilities of these models that have important implications for computing and computer science education. This paper considers findings from controlled puzzle environments that demonstrate three distinct performance 'regimes' and systematic reasoning failures in state-of-the-art LRMs. Our goal is to consider the educational implications of these limitations for computer science pedagogy, particularly with respect to assessment design, pedagogical strategies, and skill development priorities. We believe that, rather than replacing human reasoning instruction, these limitations highlight the continued importance of foundational computational thinking, algorithmic reasoning, and problem-solving skills in computer science education.

†Corresponding author

1 Introduction

Large Language Models (LLMs) and Large Reasoning Models (LRMs) have sparked debate about their potential impact on computer science (CS) education, education as a whole, and industry. These sophisticated AI systems, such as OpenAI's o1/o3 series [9], DeepSeek-R1 [7], and Claude 3.7 Sonnet Thinking [1], are designed to generate detailed reasoning traces before providing answers to complex problems. Unlike traditional Large Language Models (LLMs), LRMs explicitly demonstrate step-by-step thinking processes, leading to speculation about their potential for a paradigm shift in education and assessment.

However, recent research by Shojaee et al. [10] provides interesting insight into fundamental limitations of these current reasoning models. Through systematic evaluation using easily understood, controllable puzzle environments, their work reveals that despite sophisticated self-reflection mechanisms, LRMs exhibit predictable failure patterns and scaling limitations that have not been discussed in educational contexts.

This paper examines the educational implications of these findings for CS pedagogy. We consider how the documented limitations of LRMs should inform pedagogical decisions, assessment strategies, and curriculum design in CS programs, particularly at small and liberal arts institutions where close faculty-student relationships enable greater discussion among students and faculty. We believe that, rather than representing temporary technical challenges, these limitations may actually strengthen the case for traditional CS education approaches while suggesting new opportunities for meaningful human-AI collaboration in learning environments.

2 Background

Understanding Large Reasoning Models. LRMs represent an evolution of traditional LLMs, incorporating explicit reasoning mechanisms designed to tackle complex problem-solving tasks. Unlike standard LLMs that generate responses directly, LRMs produce detailed "thinking" processes: traces of intermediate reasoning steps that can be analyzed.

This approach builds on **Chain of Thought (CoT)** prompting [11], where models show step-by-step reasoning rather than jumping to conclusions. The technique has become foundational to reasoning model design, enabling researchers to examine not just the final answers but also the intermediate problem-solving process. These internal reasoning steps, called **thinking tokens** or reasoning traces, represent the solving work that LRMs generate before providing final answers.

The behavior of these models can be tuned through parameters like **temperature**, which controls the randomness (or creativity) of model responses. Lower temperature values yield more deterministic responses while higher temperatures promote creative generation [8]. However, all models are limited by their **token budget**: the maximum number of tokens (roughly word-like units) they can process or generate per interaction, directly constraining the depth and complexity of reasoning.

Evaluating these models requires more sophisticated metrics than simple accuracy measurements. `Pass@k` **evaluation** [6], commonly used in assessing generated code, measures the probability that at least one correct solution appears in k attempts, providing a more nuanced view of the capabilities of the model compared to the accuracy of single attempts. This approach recognizes that in real-world applications, users often generate multiple attempts when working with AI systems.

Assessment Challenges in AI Era. The integration of AI tools in education has created significant challenges for assessment design. Traditional evaluation methods face threats from AI capabilities, leading to concerns about academic integrity and the validity of student evaluation [2]. However, understanding specific AI limitations can inform more robust assessment strategies.

Data contamination represents a critical issue where AI models have been exposed to test problems during training, making standard benchmarks unreliable for evaluation [5, 12]. This contamination problem has particular relevance for CS education, where many traditional programming and algorithmic problems have inevitably been included in training datasets.

Computational Thinking and Problem Solving. CS education has long emphasized computational thinking through four components [4]. **Decomposition** involves breaking problems into smaller, more manageable parts. **Pattern recognition** identifies similarities across problems. **Abstraction** involves ignoring irrelevant details while focusing on essential features. Last, **algorithm design** develops step-by-step solutions to problems. Understanding how LRMs perform on these fundamental cognitive tasks provides crucial insights for curriculum design and pedagogical strategy.

3 Overview of Findings by Shojaee et al.

Shojaee et al. [10] used controllable puzzle environments to systematically assess LRM reasoning at varying levels of complexity. Unlike traditional benchmarks prone to data contamination, these environments, including Blocks World puzzles, River Crossing, Checker Jumping, and Tower of Hanoi, allow exact control over problem complexity while preserving consistent logical structures.

Three Reasoning Regimes. The research suggested three different performance regimes based on the complexity of the problem, each revealing important differences in how models respond to varying levels of challenge. In the **low complexity regime**, standard LLMs outperformed LRMs, achieving better accuracy with greater token efficiency. As is sometimes the case with our students, this result suggests that advanced reasoning mechanisms might cause models to 'overthink' simple problems, adding unnecessary complexity. As complexity increased, the **medium complexity regime** emerged where LRMs demonstrated clear advantages over standard models. Their explicit reasoning capabilities provided measurable benefits that justified the additional computational costs for moderately complex tasks. However, in the **high complexity regime**, both LLMs and LRMs experienced complete performance collapse despite models operating well below their token budget limits.

Reasoning Collapse and Scaling Limits. Perhaps most significantly, the research revealed systematic reasoning collapse beyond model-specific complexity thresholds. As problems grew more difficult, models actually reduced their reasoning effort (measured in thinking tokens) despite having ample computational resources. Even when provided with complete algorithms for solving problems, LRMs failed to act on them reliably, suggesting fundamental limitations in logical processing. Performance also varied dramatically across puzzle types of similar complexity, succeeding on problems requiring 100+ sequential moves in one domain while failing on 11-move problems in another domain. These findings highlight fundamental limitations in current model scalability but offer pedagogical hope—like our students, even advanced AI systems struggle when complexity increases, reinforcing the value of guided reasoning in education.

Analysis of Reasoning Traces. Detailed examination of LRM thinking processes revealed complexity affects their thinking patterns, exposing basic flaws in current reasoning approaches and notable weaknesses in their ability to self-correct. For simple problems, models exhibited an *overthinking phenomenon*, often identifying correct solutions early but continuing to explore incorrect alternatives, wasting computational resources. At moderate complexity levels, correct solutions showed *late convergence*, emerging only after extensive exploration of incorrect paths. Beyond certain complexity thresholds, models experienced *complete failure*, unable to generate any correct solutions regardless of the allowed reasoning length.

4 Educational Implications for CS

The systematic limitations identified by Shojaee et al. [10] have direct implications for CS education. In this section, we analyze key conclusions through a ‘finding-implication’ framework, where findings represent specific limitations and implications explore how these findings should inform pedagogical decision, curriculum design, and assessment strategies. This approach attempts to bridge the gap between bleeding-edge AI research and practical educational applications.

4.1 Assessment Design and Academic Integrity

- **Finding**: Providing complete algorithms to LRMs does not improve their performance on execution tasks.
  Implication: This finding implies that educators (currently) should not hesitate to provide algorithmic guidance, pseudocode, or detailed specifications in assignments. In this case, discovering algorithms is not the main challenge, the real difficulty lies in executing them reliably through clear and sequential reasoning. This continues to be a foundational skill for CS students, particularly given that, as educators can attest, following instructions is not always students’ top priority.
- **Finding**: LRMs exhibit complete accuracy collapse beyond certain complexity thresholds, regardless of available computational resources.
  Implication: Educators can design assessments with appropriate complexity levels that reliably differentiate between student work and AI assistance. Programming assignments that involve more than 15 logical steps, require managing complex states, or demand deep algorithmic reasoning may naturally be resistant to current AI capabilities. This suggests that well-designed capstone projects, complex data structure implementations, and multi-phase algorithm development remain viable assessment approaches.
- **Finding**: LRMs show inconsistent performance across different problem domains of similar complexity.
  Implication: Students relying heavily on AI tools would likely exhibit similarly inconsistent performance patterns. This supports the use of varied assessment formats including domain-specific applications or cross-disciplinary programming projects. Thus, using diverse problem types within assignments can expose gaps in student understanding.

4.2 Curriculum Design and Skill Prioritization

- **Finding**: The three performance regimes indicate to educators which complexity levels work best for specific learning goals.
  Implication: Curriculum design should strategically leverage these regimes across all course levels. For low-complexity tasks such as syntax practice and basic concept reinforcement as shown in Table 1, courses can safely incorporate AI collaboration regardless of level so that students focus cog-

Table 1: CS competency classification by AI resistance level and recommended pedagogical approaches.

CS Competency	AI Resistance	Priority	Approach
Algorithm Design	High	Critical	Human-focused
System Architecture	High	Critical	Human-led
Debugging & Testing	High	Critical	Human-centered
Code Review	High	Critical	Human expertise
Mathematical Foundations	High	Critical	Traditional
Code Implementation	Medium	High	Guided AI collab
Documentation	Medium	High	AI + review
Syntax Learning	Low	Medium	AI-assisted

nitive effort on higher-level reasoning tasks. Medium-complexity problems (e.g., multi-step algorithms, structured problem-solving, etc.) offer opportunities for guided AI collaboration where students can learn from reasoning demonstrations. Code implementation and documentation, classified as medium AI-resistance skills in Table 1, benefit from this guided collaboration approach where students maintain active oversight roles. High-complexity challenges (e.g., designing novel algorithms, implementing a solution using custom APIs) should prioritize human-centered learning, where students build skills beyond the reach of current AI.

- **Finding**: LRMs demonstrate poor self-correction, fixating on early incorrect solutions.
  Implication: It becomes essential for students to develop and refine debugging methodologies, systematic testing approaches, and analysis of errors and exceptions. CS programs can emphasize metacognitive strategies when problem-solving approaches are not working. This includes learning to identify dead ends early, systematically backtrack to previous decision points, and maintain solution quality during development. Visual problem-solving techniques, such as drawing diagrams, sketching algorithm flows, and creating state representations, become particularly valuable for developing these meta-cognitive skills. These capabilities for reflective thinking and visual reasoning represent areas where current AI falls short, making them essential competencies for students who must learn to guide and complement AI tools effectively.
- **Finding**: Models are poor at sequential planning tasks requiring sustained state tracking and logical consistency.
  Implication: Core CS concepts like algorithm correctness and complexity analysis remain essential (see Table 1). At the implementation level, students must develop precise mental models through hands-on work with

pointers, memory management, and debugging. Medium-complexity skills bridge theory and practice (e.g., recursive algorithm design, data structure trade-offs, and program invariants, etc.). Meanwhile, lab-based experiences in, for example, compiler design, database design, and software engineering offer valuable opportunities to cultivate disciplined, sequential thinking and uniquely human problem-solving abilities.

4.3 Pedagogical Strategies

- **Finding**: LRMs can generate verbose, inefficient reasoning traces, continuing work after finding correct solutions and fixating on incorrect approaches.
  Implication: Teaching students to communicate clearly and write concisely becomes increasingly important. In programming contexts, they must learn to distinguish genuine reasoning from AI-generated patterns. This reinforces the value of code documentation, algorithm explanation, and technical presentation as essential components of a quality CS education.
- **Finding**: Different reasoning models break down at different levels of complexity and show varying strengths across domains.
  Implication: Human teams can overcome individual limitations in ways current AI collaboration cannot. Thus, collaborative learning approaches become increasingly valuable, as group programming projects, code review processes, and peer debugging sessions cultivate complementary reasoning skills that AI cannot replicate.
- **Finding**: `Pass@k` evaluation shows that while multiple attempts boost AI success, the gains diminish and vary by problem type and complexity.
  Implication: When integrating AI tools into learning, limit the number of attempts to promote thoughtful engagement over trial-and-error. This encourages deliberate practice and deeper problem analysis, reducing reliance on brute-force computation.

4.4 Research Methods and Critical Evaluation

- **Finding**: The research shows that data contamination in benchmarks skews conclusions about AI capabilities.
  Implication: CS programs should prioritize experimental design, control methods, and critical evaluation to equip students with the skepticism needed to assess the strength and limitations of AI in a tech-driven workplace.
- **Finding**: Shojaee et al. [10] found that controllable experimental environments provided more reliable insights than traditional benchmarks.
  Implication: Courses teaching research methodologies should teach fair evaluation design and bias detection in AI assessments, preparing students to make informed decisions about AI adoption and limitations in their careers.

4.5 Programming and Software Development

- **Finding**: LRMs exhibit fundamental limitations in logical step execution and subsequent verification.
  Implication: Again, emphasis on testing methodologies, formal verification principles (e.g., precondition / postcondition, assertions, model checking, etc.), and systematic debugging becomes more crucial. As shown in Table 1, these debugging and testing skills represent high AI-resistance competencies where human oversight remains essential.
- **Finding**: Shojaee et al. [10] found that models demonstrate poor performance on problems requiring exact computation and algorithmic precision.
  Implication: Mathematical foundations, algorithm analysis, and computational complexity remain core competencies. Classified as critical, high AI-resistance skills in Table 1, these algorithmic principles require deep student understanding to effectively guide, verify, and complement AI tools in professional settings.

4.6 Cross-Curricular Applications

- **Finding**: Reasoning model limitations appear to be fundamental rather than domain-specific.
  Implication: These limitations reflect core reasoning challenges. Thus, the computational thinking skills that CS education develops—logical reasoning, systematic problem decomposition, and algorithmic thinking—represent uniquely human capabilities with broad disciplinary value. This idea supports CS requirements in liberal arts curricula and strengthens the case for interdisciplinary programs that integrate computational thinking.
- **Finding**: The overthinking effect highlights inefficient use of resources in AI reasoning.
  Implication: This demonstrates that teaching resource management, efficiency analysis, and optimization principles becomes increasingly relevant beyond CS. These skills support applications in data science and computational modeling across academic departments.
- **Finding**: Ballon et al. [3] conducted research on mathematical reasoning that motivated the systematic puzzle evaluation by Shojaee et al. [10]. They found that discrete mathematics stands out as a token-intensive domain and that a longer CoT does not improve performance. In contrast, foundational mathematical areas like algebra and calculus consumed fewer tokens.
  Implication: This evidence reinforces the reliability of complexity-based AI limitations identified through puzzle environments. This demonstrates that educators can apply these complexity principles beyond CS. For example, problems with heavier combinatorial or multi-step reasoning load

(e.g., counting problems, set theoretic inclusion/exclusion problems, etc.) are more resistant to AI-assistance than procedural, algorithmic problems (e.g., modulo-based equivalence classes, proof by induction of summation formulae, etc.). By designing assignments that focus on multi-step reasoning, faculty from all disciplines can more accurately assess student knowledge.

5 Limitations and Future Considerations

The research by Shojaee et al. [10] provides valuable insights; however, several limitations affect the generalizability of their findings to educational contexts. The puzzle environments represent a narrow slice of reasoning tasks compared to the breadth of problems in CS education. The deterministic nature of puzzle validation may not capture real-world programming challenges where multiple valid solutions exist and creativity plays a larger role. The research relied primarily on black-box API access, limiting analysis of internal mechanisms relevant for educational applications. While AI models will continue to improve, the systematic nature of these reasoning limitations across multiple state-of-the-art models suggests deeper challenges in current reasoning approaches rather than merely scaling issues. AI research typically advances by addressing such limitations, so these issues may well be resolved with time.

The puzzle environments selected by Shojaee et al. [10] are distinct from the types of problems that are often assigned within undergraduate courses. These puzzles often feel overwhelming to students as they require exploration rather than immediate application of techniques or algorithms they have been explicitly taught. Additionally, the amount of time that students must invest in exploration before they are able to synthesize a solution is often beyond the expectations of prior coursework. The research by Shojaee et al. seems to mirror our experiences in the classroom in introducing more creative problems that are not procedural in nature; students tend to struggle with new problems that require pushing the boundaries of their knowledge. The pitfalls that AI faces are similar in nature to the pitfalls that students face.

The implications drawn from this research assume certain educational contexts that may not apply universally. Small class sizes and close faculty-student relationships, while common at many institutions, enable more nuanced approaches to AI integration than might be feasible in larger educational settings. The focus on controlled problem environments may not fully capture the collaborative and iterative nature of real-world software development, where AI tools may offer distinct value. However, student populations may vary significantly in their prior AI exposure and comfort levels, affecting how these tools integrate into learning processes.

While the research identifies fundamental limitations in current reasoning architectures, continued technological development may address some identified issues. Educational institutions must balance their preparation for current technological realities with an anticipation of future developments. The emphasis on fundamental reasoning skills suggested by this research appears robust to technological change, as these capabilities remain valuable regardless of AI advancement.

6 Conclusions

The research examining LRM limitations provides crucial guidance for CS education in this AI era. Rather than suggesting wholesale changes to curriculum or pedagogy, the findings support a nuanced approach that leverages current AI capabilities while strengthening uniquely human reasoning skills. The identification of three distinct performance regimes offers a framework for strategic AI integration: utilizing cost-effective standard models for basic concept reinforcement, engaging with reasoning models for intermediate complexity learning, and emphasizing human-centric approaches for advanced problem-solving that exceeds current AI capabilities.

Perhaps most importantly, the documented limitations in algorithmic execution, verification, and logical consistency highlight the continued importance of foundational CS education. Skills in debugging, testing, formal reasoning, and systematic problem decomposition remain not only relevant but essential for effective human-AI collaboration. For CS educators, these findings suggest confidence in traditional pedagogical approaches while identifying specific opportunities for meaningful integration with current AI tools. Assessment strategies can be designed with complexity thresholds in mind, curriculum can be structured to leverage appropriate AI capabilities, and skill development can focus on areas where human reasoning provides irreplaceable value.

The broader implication extends beyond CS to the development of critical thinking and analytical reasoning capabilities. The limitations identified in sophisticated AI systems underscore the value of human reasoning development across disciplines. As AI tools continue to evolve, the emphasis on metacognitive skills, collaborative reasoning, and systematic problem-solving approaches suggested by Shojaee et al. provides a robust foundation for students regardless of technological advancement. Rather than competing with current AI capabilities, effective CS education can prepare students to guide, verify, and complement these powerful but currently limited tools. The illusion of AI reasoning, as revealed through systematic evaluation, ultimately strengthens the case for rigorous CS education focused on developing the reasoning capabilities that current technology cannot replicate or replace.

References

[1] Anthropic. *Claude 3.7 Sonnet*. https://www.anthropic.com. 2025.

[2] Ali Ateeq et al. "Artificial intelligence in education: implications for academic integrity and the shift toward holistic assessment". In: *Frontiers in Education* Volume 9 - 2024 (2024). ISSN: 2504-284X. DOI: 10.3389/feduc.2024.1470979.

[3] Marthe Ballon, Andres Algaba, and Vincent Ginis. "The relationship between reasoning and performance in large language models-03 (mini) thinks harder, not longer". In: (2025). arXiv: 2502.15631 [cs.LG]. URL: https://arxiv.org/abs/2502.15631.

[4] Ünal Çakiroğlu and Volkan Selçuk. "Machine learning meets secondary school classrooms: using hands-on activities to advance computational thinking". In: *Education and Information Technologies* 30.7 (Dec. 2024), pp. 9547–9571. ISSN: 1360-2357. DOI: 10.1007/s10639-024-13196-8.

[5] Nicholas Carlini et al. "Extracting Training Data from Large Language Models". In: (2021). arXiv: 2012.07805 [cs.CR]. URL: https://arxiv.org/abs/2012.07805.

[6] Mark Chen et al. *Evaluating Large Language Models Trained on Code*. 2021. arXiv: 2107.03374 [cs.LG]. URL: https://arxiv.org/abs/2107.03374.

[7] DeepSeek-AI et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL]. URL: https://arxiv.org/abs/2501.12948.

[8] Ari Holtzman et al. "The Curious Case of Neural Text Degeneration". In: (2020). arXiv: 1904.09751 [cs.CL]. URL: https://arxiv.org/abs/1904.09751.

[9] OpenAI. *Introducing OpenAI o1*. https://openai.com/blog/introducing-o1. 2024.

[10] Parshin Shojaee et al. "The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity". In: (2025). arXiv: 2506.06941 [cs.AI]. URL: https://arxiv.org/abs/2506.06941.

[11] Jason Wei et al. "Chain of Thought Prompting Elicits Reasoning in Large Language Models". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh et al. 2022. URL: https://openreview.net/forum?id=_VjQlMeSB_J.

[12] Cheng Xu et al. *Benchmark Data Contamination of Large Language Models: A Survey*. 2024. arXiv: 2406.04244 [cs.CL]. URL: https://arxiv.org/abs/2406.04244.