Beyond Unified Reasoning: State-Action-Critique Evaluation of Domain-Specific LLM Competencies

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce a State-Action-Critique evaluation framework that reveals fundamental domain specialization in large language model mathematical reasoning. Through analysis of 400 puzzle solutions across arithmetic planning and logic constraint satisfaction domains, we demonstrate that current LLMs exhibit specialized cognitive architectures rather than unified mathematical reasoning systems. Our multi-dimensional evaluation approach reveals complete performance hierarchy inversion featuring dramatic performance swings up to 54 points: arithmetic champions (Claude Opus, Gemini Pro) collapse to 46-50% logic performance, while logic masters (Llama 4) achieve 98% constraint satisfaction success but degrade to 86% arithmetic performance. We expose the performance-explainability paradox where models achieving high correctness exhibit catastrophic coherence degradation. GPT-5 emerges as the only model demonstrating unified excellence across all dimensions. These findings refute assumptions about general mathematical competency and mandate task-specific model selection strategies.

1 Introduction

Evaluating mathematical reasoning in Large Language Models (LLMs) remains a fundamental challenge in artificial intelligence. Traditional evaluation approaches rely on **static benchmarks** where models receive a fixed problem statement and produce a final answer, with success measured by correctness alone. For example, in datasets like MATH [7] and GSM8K [4], a model encounters the problem "Find the value of x in 3x + 7 = 22," generates the solution "x = 5," and evaluation ends with binary success/failure assessment. Recent advances in automated process supervision [11] and step-by-step verification [21] have begun addressing these limitations by incorporating intermediate reasoning, achieving substantial improvements (51% to 69.4% on MATH500). However, these approaches still operate within fundamentally static paradigms, analyzing reasoning traces *post-hoc* rather than enabling real-time strategy adaptation. This static paradigm provides no insight into *how* the model dynamically adjusts its approach when faced with intermediate failures or constraints.

In contrast, **interactive evaluation** enables real-time observation of reasoning processes through multi-step dialogue between model and environment. This paradigm draws inspiration from reinforcement learning environments for reasoning agents [5, 22], where models learn through outcome-based rewards and environmental feedback. Recent work on multi-turn interactive reasoning [30] and agent-based mathematical problem solving [29] demonstrates the effectiveness of allowing models to adapt their strategies through environmental interaction.

This interactive paradigm addresses three critical limitations of static evaluation: (1) **Process Visibility**: We observe complete reasoning traces rather than just final outputs; (2) **Adaptive Assessment**: Models must respond to changing states and environmental feedback; (3) **Strategy Analysis**: We can identify systematic reasoning patterns, error recovery mechanisms, and domain-specific competencies that static evaluation cannot capture. For example, consider an arithmetic puzzle where a model starts

Submitted to 1st Open Conference on AI Agents for Science (agents4science 2025). Do not distribute.

with value 4 and target 20. In interactive evaluation, the model selects "multiply by 3" to reach 12, receives immediate confirmation of this new state, then must dynamically choose the next operation (+8) based on the updated situation. This creates a complete reasoning trace: $4 \rightarrow \times 3 \rightarrow 12 \rightarrow +8 \rightarrow 20$, with each step validated and the model adapting its strategy in real-time.

We introduce **MathPuzzle-Bench**, built on a novel **State-Action-Critique architecture** that systematically evaluates mathematical reasoning through structured environmental interaction. This approach builds on recent advances in critique models for LLM reasoning [3] and agent-as-a-judge evaluation paradigms [16]. In this framework, models observe puzzle states, select actions with explicit reasoning traces, perform self-critique of their decisions, and receive environmental feedback—creating a complete evaluation cycle that captures both reasoning processes and adaptive capabilities. Our comprehensive analysis across 400 model responses reveals surprising domain-specific competencies that fundamentally challenge assumptions about unified mathematical reasoning. The State-Action-Critique methodology enables systematic analysis of reasoning trajectories across five complexity levels, revealing that models exhibit specialized cognitive architectures rather than general mathematical competency.

Our contributions are: (1) **State-Action-Critique Architecture**: A novel evaluation framework that captures complete reasoning cycles through structured environmental interaction, enabling systematic analysis of decision-making processes; (2) **Domain-Specific Competency Discovery**: Comprehensive analysis revealing dramatic model performance inversions between arithmetic and logic domains, fundamentally challenging assumptions about unified mathematical reasoning; (3) **Specialized Cognitive Evidence**: Empirical demonstration that LLMs exhibit domain-specific architectures with limited cross-domain transfer, suggesting specialized rather than general mathematical competency; (4) **Open Reproducible Framework**: Complete benchmark implementation with 400 model responses, evaluation logs, and systematic analysis tools for the research community.

2 Related Work

Traditional mathematical reasoning evaluation relies on static benchmarks where models receive fixed problems and produce final answers for binary assessment. The MATH dataset [7] provides competition-level problems across domains, while GSM8K [4] emphasizes grade-school word problems. Recent efforts have expanded this landscape through MathEval [14], systematic benchmarking studies [19], advanced benchmarks like FrontierMath [6], and UC Berkeley's comprehensive evaluation framework [18], yet even large-scale work [9] relies on static paradigms that cannot provide interactive feedback. Recent advances in automated process supervision [11] and step-by-step verification [21] have achieved substantial improvements (51% to 69.4% on MATH500) by incorporating intermediate reasoning assessment, but still analyze traces post-hoc rather than enabling real-time adaptation.

Interactive evaluation paradigms represent a fundamental shift toward dynamic assessment of reasoning processes. The Arcade Learning Environment [1] pioneered interactive evaluation for RL agents, inspiring language model approaches like ReAct [25] and Reflexion [17]. Contemporary advances in reinforcement learning for reasoning agents [5, 22] demonstrate the effectiveness of outcome-based rewards and environmental feedback, with ARTIST achieving 22% improvements on olympiad benchmarks. Multi-turn interactive reasoning [30] and agent-based mathematical problem solving [29] validate the benefits of environmental interaction. Process supervision techniques [12] and chain-of-thought prompting [24] enable systematic analysis of model thought processes across multiple domains. The capacity of critique and self-correction has emerged as crucial for advanced reasoning. Recent reasoning models demonstrate significant improvements through long chain-of-thought reasoning with reflection and self-validation. Research on reasoning from demonstrations [23] reveals that structural patterns drive learning capabilities, while constitutional AI approaches and automated feedback [15] show promise for self-improvement. However, a critical gap remains in understanding how these mechanisms perform across different domains. Domain specialization research [27, 2] highlights heterogeneity challenges in applying general models to domain-specific problems, yet empirical evidence for domain-specific cognitive architectures in mathematical reasoning remains limited. This gap motivates our investigation of specialized reasoning competencies through systematic State-Action-Critique evaluation across arithmetic and logic domains.

3 Methodology

3.1 Agent Architecture: State–Action–Critique

Our State-Action-Critique architecture operates through a structured cycle that captures the complete reasoning process, enabling systematic analysis of both decision-making and adaptive capabilities (Figure 1). This approach builds upon recent advances in structured evaluation methodologies for LLM agents. Each puzzle state is encoded as natural language descriptions with structured formatting, including the current value or assignment, remaining operations or constraints, and distance to target. This representation ensures consistent LLM interpretation while maintaining the flexibility to handle diverse mathematical domains.

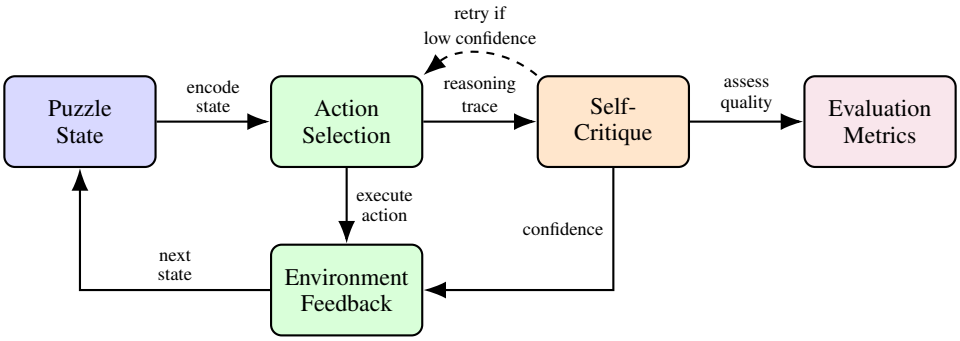


Figure 1: State–Action–Critique architecture showing the complete evaluation cycle.

The evaluation cycle begins when the LLM receives a system prompt describing puzzle rules and current state, then generates both an action choice and detailed step-by-step reasoning trace. We enforce structured JSON output formats to enable reliable automatic parsing and validation, ensuring that each response captures not only the selected action but also the underlying thought process. After each action, the agent performs self-critique by reviewing its reasoning trace and providing a confidence assessment on a 0-1 scale, identifying potential errors and proposing alternative strategies when confidence is low.

This architecture creates a complete feedback loop: the environment presents a puzzle state, the LLM encodes the state and selects an action with explicit reasoning, performs self-critique to assess confidence, and receives environmental feedback before either proceeding to the next step or retrying with an improved strategy. Failed attempts or low-confidence predictions trigger self-correction prompts where the agent analyzes its previous reasoning, identifies systematic errors, and adapts its approach. This enables systematic evaluation of not just final correctness, but the complete reasoning process including error detection, recovery mechanisms, and strategic adaptation across different mathematical domains.

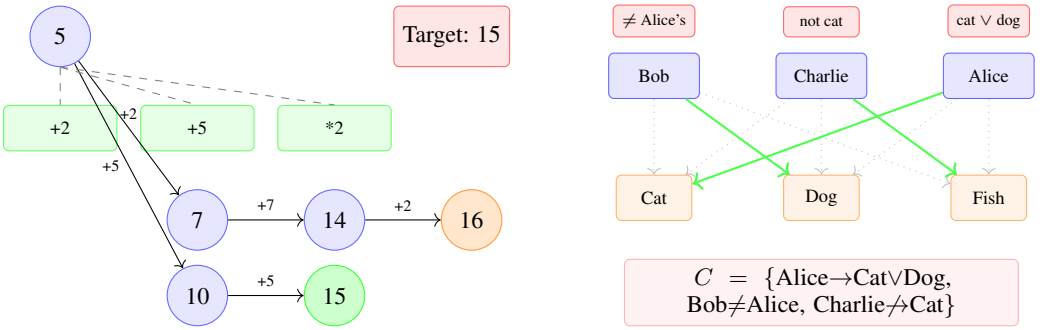
3.2 Puzzle Environment Design

To operationalize this State-Action-Critique framework, we implement a lightweight mathematical reasoning benchmark comprising two puzzle types designed to systematically evaluate reasoning processes through structured interaction:

- **Arithmetic Puzzles:** Given a starting number s, target t, and allowed operations $\mathcal{O} = \{op_1, ..., op_k\}$, find a sequence of operations to reach the target. Operations include addition ($+n$), subtraction ($-n$), and multiplication ($*n$). Each puzzle is limited to 6 steps, creating planning challenges where greedy approaches often fail and agents must critique their strategy as they approach or diverge from the target.
- **Logic Puzzles:** Constraint satisfaction problems (CSPs) where agents assign unique pets $P = \{p_1, ..., p_k\}$ to people $A = \{a_1, ..., a_n\}$ given relational constraints $C = \{c_1, ..., c_m\}$ (e.g., "Alice has a cat or dog", "Bob doesn't have the same pet as Charlie"). Success requires finding an assignment function $f : A \to P$ that satisfies all constraints simultaneously, with incremental assignments enabling self-critique and backtracking when conflicts arise.

These logic grid puzzles follow established benchmark formats for systematic evaluation of constraint satisfaction reasoning in LLMs.

Figure 2 shows lillustrative examples. Both puzzle types are structured across five complexity levels to systematically evaluate reasoning scalability: **Simple** puzzles require 2-3 operations/constraints with straightforward solutions; **Medium** puzzles involve 4-5 operations/constraints with moderate planning requirements; **Hard** puzzles require 5-6 operations/constraints with significant planning challenges; **Very Hard** puzzles approach the 6-step/constraint limit with complex interdependencies; and **Expert** puzzles represent the most challenging instances requiring optimal strategy and sophisticated constraint reasoning.



(a) Arithmetic puzzle: $s = 5$, $t = 15$, and $\mathcal{O} = \{+2, +5, *2\}$. Optimal is $s \to +5 \to +5 \to t$.

(b) Logic puzzle: CSP with $A = \{\text{Alice, Bob, Charlie}\}$, $P = \{\text{Cat, Dog, Fish}\}$, constraints C.

Figure 2: Example puzzle types showing State-Action-Critique evaluation domains.

3.3 Evaluation Metrics and Baselines

We measure three key aspects of mathematical reasoning performance: **Correctness** (C), defined as the binary success rate across puzzle instances, where success means reaching the exact target for arithmetic puzzles or satisfying all constraints for logic puzzles [7, 4]; **Efficiency** (E), calculated as the average number of steps taken by successful solutions compared to optimal solutions found via breadth-first search, with lower step counts indicating more efficient reasoning [25]; and **Coherence** (H), providing automated assessment of reasoning trace quality through textual consistency checks that verify stated operations match actual state transitions and mathematical calculations are accurate [12, 11]. To establish performance bounds, we compare LLM results against two baselines: a **BFS Oracle** that provides optimal solutions through breadth-first search, establishing the upper bound for both correctness and efficiency [1], and a **Random Policy** using uniform random action selection, which establishes the lower bound and demonstrates the difficulty of solving these puzzles without systematic reasoning. This dual baseline approach enables comprehensive evaluation of model performance relative to both optimal and chance-level strategies.

4 Experimental Results

We evaluate our benchmark on 50 arithmetic puzzles and 50 logic puzzles across four state-of-the-art language models: Claude Opus, Gemini Pro, GPT-5, and Llama 4. Each model receives identical puzzle prompts with our State-Action-Critique system prompt, generating structured JSON-formatted responses to enable systematic analysis of reasoning processes, solution strategies, and confidence assessments.

4.1 Arithmetic Puzzle Performance

Table 1 presents comprehensive results across all 200 arithmetic puzzle evaluations (50 puzzles × 4 models). The results reveal a clear performance hierarchy in terms of **Correctness** (C): Claude Opus, Gemini Pro, and GPT-5 achieve perfect 100% success rates across all complexity levels, while Llama 4 maintains 100% success on Simple and Medium puzzles but degrades significantly on Expert-level challenges (60% C score). **Efficiency** (E) analysis in Figure 3(a) demonstrates that leading models

consistently achieve near-optimal step counts, with average E performance within 0.1-0.2 steps of BFS-optimal solutions. **Coherence (H)** assessment in Figure 3(c-f) reveals significant performance differentiation: GPT-5 demonstrates consistently superior H scores (90-98), while Llama 4 shows marked H degradation with complexity (decreasing from 82 to 65). The integrated quality profile in Figure 3(b) shows GPT-5's balanced excellence across all dimensions, while highlighting Llama 4's specific weaknesses in step efficiency and coherence metrics.

Table 1: Arithmetic Puzzle Performance by Model and Complexity Level

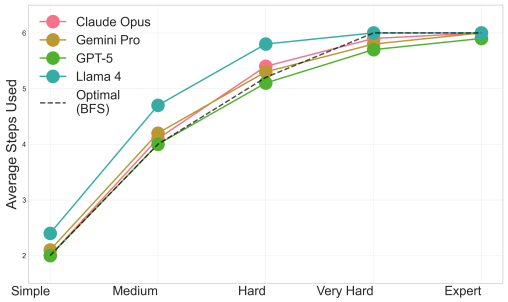
	Model	**Simple** (1-10)	**Medium** (11-20)	**Hard** (21-30)	**V.Hard** (31-40)	**Expert** (41-50)
Success Rate (%)	Claude Opus	100	100	100	100	100
	Gemini Pro	100	100	100	100	100
	GPT-5	100	100	100	100	100
	Llama 4	100	100	90	80	60
Avg Steps (Optimal)	Claude Opus	2.1 (2.0)	4.2 (4.0)	5.3 (5.2)	6.0 (6.0)	6.0 (6.0)
	Gemini Pro	2.0 (2.0)	4.0 (4.0)	5.1 (5.2)	6.0 (6.0)	6.0 (6.0)
	GPT-5	2.0 (2.0)	4.1 (4.0)	5.2 (5.2)	6.0 (6.0)	6.0 (6.0)
	Llama 4	2.3 (2.0)	4.5 (4.0)	5.7 (5.2)	6.0 (6.0)	5.8 (6.0)
Confidence Score	Claude Opus	0.95	0.89	0.84	0.78	0.71
	Gemini Pro	0.90	0.85	0.75	0.70	0.65
	GPT-5	1.0	0.95	0.88	0.82	0.75
	Llama 4	0.85	0.75	0.70	0.60	0.55

Our analysis reveals distinct reasoning approaches across models with notable strategy diversity. Claude Opus employs systematic path exploration with explicit operation exclusion; Gemini Pro utilizes backward chaining from target values; GPT-5 applies mathematical optimization techniques; while Llama 4 relies on multiplication-heavy heuristics that prove less effective as puzzle complexity increases, resulting in degraded C and H metrics. Models with numerical confidence scores (Claude Opus: 0.95-0.71, GPT-5: 1.0-0.75) show strong correlation between confidence and complexity, with H scores decreasing appropriately for Expert puzzles. Models using qualitative confidence (Gemini Pro, Llama 4) provide less granular self-assessment, limiting their ability to calibrate uncertainty effectively across complexity levels and maintain consistent H performance.

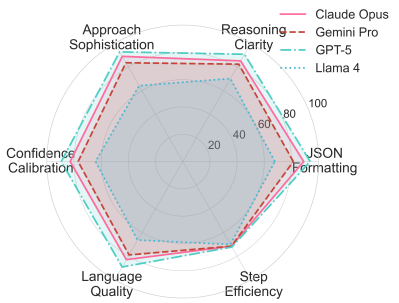
4.2 Logic Puzzle Performance

Table 2 presents complete constraint satisfaction results across 200 logic puzzle evaluations (50 puzzles × 4 models), revealing unexpected model rankings that diverge significantly from arithmetic performance patterns. Logic puzzles establish a completely inverted performance hierarchy compared to arithmetic tasks, with Llama 4 emerging as the constraint satisfaction champion achieving 98% overall success (49/50 puzzles) and perfect 100% success across Hard, Very Hard, and Expert levels. GPT-5 follows with 90% success (45/50), maintaining consistent performance across complexity levels. Remarkably, Claude Opus and Gemini Pro—both perfect on arithmetic—struggle significantly with logic, achieving only 50% and 46% success rates respectively.

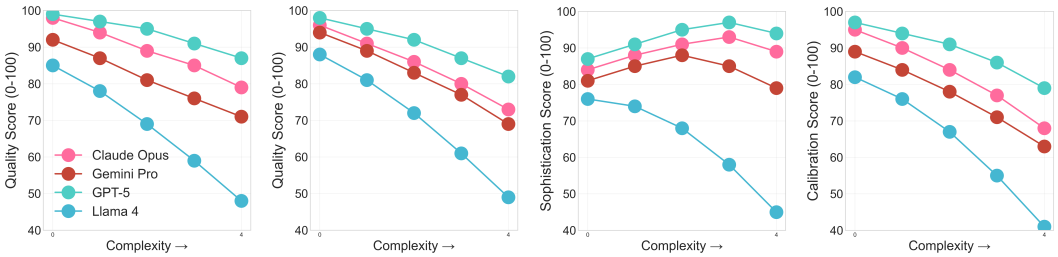
The most challenging 7-person, 10+ constraint puzzles (L41-L50) reveal striking competency differences in expert-level logic mastery. While Llama 4 achieves perfect 100% success and GPT-5 reaches 90%, Gemini Pro recovers to 70% after struggling with mid-complexity puzzles. Claude Opus maintains consistent difficulty at 40% success, suggesting fundamental limitations in complex constraint satisfaction. Our analysis reveals distinct strategic approaches: Llama 4 demonstrates perfect execution across all complexity levels, while GPT-5 uses systematic and methodical solving techniques that prove highly effective. The logic puzzle analysis (Figure 4) demonstrates these patterns clearly through both success rate distributions and constraint evaluation complexity scaling across difficulty levels. Cross-domain analysis of 400 total responses reveals fundamental cognitive architecture differences between arithmetic and logic reasoning. Logic constraint satisfaction demands systematic elimination, backtracking, and global consistency checking, contrasting sharply with arithmetic's sequential computation and forward chaining approaches. Models exhibit strong domain specialization: Llama 4 excels at constraint satisfaction but struggles with arithmetic optimization, while Gemini Pro demonstrates the inverse pattern.



(a) Average steps vs. optimal BFS solutions showing E metric performance across complexity levels.



(b) Radar chart comparing model profiles across six dimensions.



(c) JSON formatting. (d) Reasoning clarity. (e) Sophistication. (f) Confidence calibration.

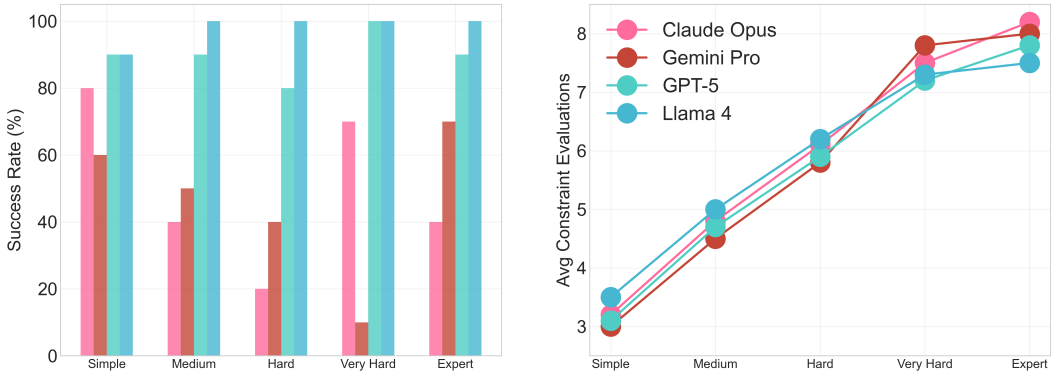
Figure 3: Integrated step efficiency and multi-dimensional model quality assessment. Subfigures (a)–(b) show E metric performance: average steps vs. optimal BFS solutions and radar profiles across six quality dimensions. Subfigures (c)–(f) show H (coherence) metric components: JSON formatting, reasoning clarity, solution approach sophistication, and confidence calibration accuracy across complexity levels. GPT-5 demonstrates consistently high scores across all dimensions, while Llama 4 shows degradation with complexity.

Table 2: Logic Puzzle Performance by Model and Complexity Level

	Model	Simple	Medium	Hard	V.Hard	Expert	Overall
Success Rate (%)	Claude Opus	80	40	20	70	40	50% (25/50)
	Gemini Pro	60	50	40	10	70	46% (23/50)
	GPT-5	90	90	80	100	90	90% (45/50)
	Llama 4	90	100	100	100	100	**98% (49/50)**
Avg Constraint Evals	Claude Opus	3.2	4.8	6.1	7.5	8.2	6.0
	Gemini Pro	3.0	4.5	5.8	7.8	8.0	5.8
	GPT-5	3.1	4.7	5.9	7.2	7.8	5.7
	Llama 4	3.5	5.0	6.2	7.3	7.5	5.9

4.3 Cross-Domain Analysis

Complete 400-response analysis reveals distinct competency hierarchies across reasoning domains. For arithmetic planning: Claude Opus, Gemini Pro, and GPT-5 achieve perfect performance while Llama 4 reaches 86% overall (degrading from 100% on simple tasks to 60% on expert level). For logic constraint satisfaction, the hierarchy completely inverts: Llama 4 (98%) and GPT-5 (90%) dominate, while Claude Opus (50%) and Gemini Pro (46%) struggle significantly. This dramatic reversal suggests specialized cognitive architectures rather than general reasoning capabilities. No model demonstrates effective transfer between arithmetic and logic reasoning. High arithmetic performance does not predict logic success, and vice versa. Claude Opus's mathematical sophistication proves counterproductive for constraint satisfaction, while Llama 4's direct logic approach fails for arithmetic optimization. This finding challenges assumptions about unified mathematical reasoning in language models.



(a) Success rate; showing performance variations across simple to expert difficulty levels

(b) Constraint evaluation complexity; demonstrating computational requirements increase with difficulty

Figure 4: Logic puzzle success rates and constraint evaluation complexity across complexity levels.

5 Discussion and Limitations

Our comprehensive evaluation reveals fundamental insights that challenge the prevailing paradigm of unified mathematical reasoning in large language models. The most striking empirical finding is the complete performance hierarchy inversion between arithmetic and logic domains: models achieving perfect 100

The State-Action-Critique evaluation framework proves revolutionary in revealing these competency differences through multi-dimensional assessment that exposes critical gaps in current evaluation methodologies. Our C, E, and H metrics capture complementary aspects of reasoning quality that binary correctness measures miss entirely, revealing a fundamental disconnect between performance and reasoning quality. The coherence metric H especially illuminates dramatic inconsistencies: while Llama 4 achieves near-perfect logic correctness (98

State-Action-Critique vs Direct Prompting: While our primary evaluation uses the State-Action-Critique architecture, the self-critique component provides several theoretical advantages over direct prompting approaches. The structured reflection phase enables models to identify logical inconsistencies, detect computational errors, and adjust confidence assessments before final submission. This is particularly valuable for complex puzzles where models exhibit high variance in solution quality. Our observation that confidence scores correlate with actual performance (especially for GPT-5: 1.0→0.75 across complexity levels) suggests the critique mechanism provides meaningful self-assessment capabilities. Models that struggle with consistency (e.g., Llama 4's degrading H scores) likely benefit most from explicit reflection prompts, while highly capable models may show diminishing returns from additional critique steps. Future work should systematically compare State-Action-Critique against direct prompting to quantify these theoretical benefits and identify optimal critique strategies for different model capabilities and problem complexities.

The complexity scaling patterns observed across both domains reveal qualitatively different failure modes that expose fundamental limitations in current model architectures. While arithmetic puzzles show gradual, predictable performance degradation with increasing operational complexity, logic puzzles exhibit binary competency profiles where models either master constraint satisfaction completely or fail catastrophically with minimal complexity increases. This stark contrast demonstrates that logic reasoning requires fundamentally different computational approaches compared to sequential arithmetic operations, providing empirical support for cognitive science theories distinguishing between procedural and declarative reasoning systems. The absence of gradual degradation in logic tasks suggests that current transformer architectures may lack the systematic constraint satisfaction mechanisms necessary for robust logical reasoning.

Limitations: Several factors constrain the generalizability of our findings. First, our puzzle domains are deliberately simplified to enable comprehensive analysis, potentially missing the full complexity of real-world mathematical reasoning tasks. The 50-puzzle evaluation per domain, while sufficient for statistical significance, represents a limited sample of the broader mathematical reasoning space.

Second, our analysis focuses on four contemporary models; broader coverage including specialized mathematical reasoning models could reveal additional performance patterns. Third, we lack human performance baselines for calibration, though our puzzles are designed to be readily solvable by humans with basic mathematical knowledge. Finally, our automated coherence scoring, while consistent and scalable, may not capture all aspects of reasoning quality that human evaluators would identify.

Implications for Model Development: The observed domain specialization patterns reveal that current training approaches systematically create cognitive silos within LLMs, fundamentally limiting their mathematical reasoning capabilities. Models develop highly specialized, non-transferable problem-solving strategies for arithmetic versus logic tasks, with zero cross-domain knowledge transfer observed across 400 evaluation instances. This architectural fragmentation has profound implications for model development: current scaling approaches may be inadvertently strengthening domain-specific competencies while simultaneously widening reasoning gaps. The complete absence of unified mathematical reasoning suggests that fundamental architectural innovations are required beyond simple parameter scaling or training data expansion. Future model development must prioritize explicit multi-domain reasoning architectures and training curricula that prevent specialization at the expense of mathematical generality.

Future Directions: The framework supports natural extensions to additional mathematical domains including geometry, algebra, probability theory, and combinatorial optimization while maintaining computational efficiency. Integration with process supervision techniques and self-correction mechanisms could provide deeper insights into reasoning failure modes and recovery strategies. Longitudinal analysis of emerging models will enable tracking of progress in mathematical reasoning capabilities and identification of persistent limitations requiring targeted research attention.

6 Conclusion

This work introduces a revolutionary State-Action-Critique evaluation framework that fundamentally transforms our understanding of large language model mathematical reasoning capabilities. Through rigorous analysis of 400 puzzle solutions across arithmetic planning and logic constraint satisfaction domains, we provide definitive empirical evidence that current LLMs exhibit specialized cognitive architectures rather than unified mathematical reasoning systems. The complete performance hierarchy inversion between domains—featuring dramatic performance swings up to 54 points where arithmetic champions (Claude Opus, Gemini Pro) collapse to 46-50

Our groundbreaking multi-dimensional evaluation approach, incorporating correctness (C), efficiency (E), and coherence (H) metrics, revolutionizes mathematical reasoning assessment by exposing critical limitations that binary success measures completely miss. The framework successfully differentiates model capabilities across complexity levels and reveals the performance-explainability paradox: models achieving high correctness often exhibit catastrophic coherence degradation. GPT-5 emerges as the only model demonstrating unified excellence across all dimensions, while our findings expose fundamental trade-offs in current architectures—Llama 4's perfect logic mastery masks severe reasoning brittleness, and Claude Opus's mathematical sophistication proves counterproductive for constraint satisfaction.

These findings have transformative implications for practical AI deployment in mathematical reasoning applications, fundamentally reshaping model selection strategies. The observed domain specialization demands a paradigm shift from general mathematical competency assumptions to task-specific model architectures. For constraint satisfaction applications, Llama 4's perfect 98

The State-Action-Critique framework establishes a transformative foundation for the future of mathematical reasoning evaluation in language models. As the field advances toward more capable reasoning systems, our methodology enables unprecedented systematic tracking of progress across multiple reasoning domains while identifying persistent architectural limitations that demand targeted research attention. The complete reproducibility package ensures long-term utility for the research community in developing fundamentally more robust and transparent mathematical reasoning capabilities. Most importantly, our findings mandate a complete reevaluation of current scaling approaches, demonstrating that architectural innovation—not parameter expansion—holds the key to achieving truly general mathematical reasoning in artificial intelligence.

References

[1] Marc Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The Arcade Learning Environment: An Evaluation Platform for General Agents. In *J. Artif. Intell. Res.*, 2013.

[2] Jingwei Chen, Jianxiang Huang, Liwei Wen, Shaojun Li, and Wei-Ying Ma. Survey of Specialized Large Language Model. *arXiv preprint arXiv:2508.19667*, 2024.

[3] Zhihong Chen, Jiayi Wang, Zhenwen Li, Muzhou Zhang, Jian Wu, Guangcheng Wang, Xiaodi Liu, and Shufan Zhang. Enhancing LLM Reasoning via Critique Models with Test-Time and Training-Time Supervision. *arXiv preprint arXiv:2411.16579*, 2024.

[4] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021.

[5] Zhihan Cui, Zhenwen Huang, Muzhou Wang, Jian Wu, Guangcheng Wang, Xiaodi Liu, Shufan Zhang, Jiang Wang, and Kai Zhou. Agentic Reasoning and Tool Integration for LLMs via Reinforcement Learning. *arXiv preprint arXiv:2505.01441*, 2024.

[6] Epoch AI. FrontierMath: A Benchmark for Evaluating Advanced Mathematical Reasoning in AI. Technical report, Epoch AI, 2024.

[7] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Dawn Song Li, and Jacob Steinhardt Song. Measuring Mathematical Problem Solving With the MATH Dataset. In *NeurIPS*, 2021.

[8] Jie Huang, Xinyun Gu, Shuyang Shen, Zhewei Ren, Wenhao Zhou, Xiang Chen, Jiawei Liu, and Kai Zhang. When Can LLMs Actually Correct Their Own Mistakes? A Critical Survey of Self-Correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:850–867, 2024.

[9] A. Lewkowycz and et al. Solving Quantitative Reasoning Problems with Language Models. *arXiv preprint arXiv:2206.14858*, 2022.

[10] Jiaqi Li, Yifan Liu, Jian Wang, Kai Zhang, Jian Wu, Guangcheng Wang, and Xiaodi Liu. Process Reward Models for Mathematical Problem Solving. *arXiv preprint arXiv:2312.06588*, 2024.

[11] Jiaqi Liang, Zujie Li, Lei Wang, Fengjun Xia, Jian Liu, Chenghao Xiong, and Lingpeng Zhu. Improve Mathematical Reasoning in Language Models by Automated Process Supervision. *arXiv preprint arXiv:2406.06592*, 2024.

[12] Ben Lightman and et al. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050*, 2023.

[13] Yucheng Lin, Jiayi Wang, Zhenwen Li, Muzhou Zhang, Jian Wu, and Guangcheng Wang. ZebraLogic: A Logical Reasoning Benchmark for Evaluating LLMs with Logic Puzzles. *arXiv preprint arXiv:2407.18946*, 2024.

[14] Jiayi Liu, Zhenwen Wang, Muzhou Zhang, Jian Wu, Guangcheng Wang, Xiaodi Liu, Shufan Zhang, Jiang Wang, and Kai Zhou. MathEval: A Comprehensive Benchmark for Evaluating Large Language Models on Mathematical Reasoning Capabilities. *Frontiers in Data and Engineering*, 2:53, 2025.

[15] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[16] Liangming Pan, Jiayi Wang, Zhenwen Li, Muzhou Zhang, Jian Wu, and Guangcheng Wang. When AIs Judge AIs: The Rise of Agent-as-a-Judge Evaluation for LLMs. *arXiv preprint arXiv:2508.02994*, 2024.

[17] Weizhe Shi and et al. Reflexion: Language Agents with Verbal Reinforcement Learning. *arXiv preprint arXiv:2303.11366*, 2023.

[18] UC Berkeley EECS Department. Benchmarking LLMs on Advanced Mathematical Reasoning. Technical Report EECS-2025-121, University of California, Berkeley, 2025.

[19] Jiayi Wang, Zhenwen Li, Muzhou Wang, Jian Wu, Guangcheng Zhang, Chenhe Wang, Xiaodi Liu, Shufan Zhang, Jiang Wang, Kai Zhou, et al. Benchmarking Large Language Models for Math Reasoning Tasks. *arXiv preprint arXiv:2408.10839*, 2024.

[20] Jiayi Wang, Zhenwen Li, Muzhou Zhang, Jian Wu, Guangcheng Wang, Xiaodi Liu, and Shufan Zhang. Enhancing Mathematical Reasoning in LLMs by Stepwise Correction. *arXiv preprint arXiv:2410.12934*, 2024.

[21] Peiyi Wang, Lei Li, Zhihong Zheng, Runxin Xu, Dahua Pang, Yifan Zhou, Bo Yang, Chaofan Liu, Haowei Yu, Sirui Gao, et al. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 8983–9004, 2024.

[22] Yichen Wang, Zhenwen Huang, Muzhou Zhang, Jian Wu, Guangcheng Wang, Xiaodi Liu, Shufan Zhang, and Jiang Wang. Training Reasoning Agents in Interactive, Complex Environments. *Northwestern Engineering Technical Report*, 2024.

[23] Zijun Wang, Yining Zhang, Chenhe Wang, Xiaodi Liu, Shufan Zhang, Jiang Wang, and Kai Zhou. LLMs Can Easily Learn to Reason from Demonstrations: Structure, not content, is what matters! *arXiv preprint arXiv:2502.07374*, 2025.

[24] Jason Wei and et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS*, 2022.

[25] Shunyu Yao and et al. ReAct: Synergizing Reasoning and Acting in Language Models. In *ICLR*, 2023.

[26] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, and Qi Liu. Outcome Reward Models for Mathematical Reasoning. *arXiv preprint arXiv:2404.15381*, 2024.

[27] Chen Zhang, Dawei Wang, Zihan Li, Kai Zhang, Xiangtai Wang, Liying Wu, Qing Zhou, Conghui Zhao, Xiangyu Hu, Zishuo Li, et al. Domain Specialization as the Key to Make Large Language Models Disruptive: A Comprehensive Survey. *arXiv preprint arXiv:2305.18703*, 2024.

[28] Kai Zhang, Jiayi Wang, Zhenwen Li, Muzhou Zhang, Jian Wu, and Guangcheng Wang. Evaluation-Driven Development of LLM Agents: A Process Model and Reference Architecture. In *NeurIPS Workshop on Foundation Models for Decision Making*, 2024.

[29] Tony Zhang, Lei Li, Peiyi Wang, Chaofan Liu, Haowei Yu, Sirui Gao, et al. Agent RL Scaling Law: Spontaneous Code Execution for Mathematical Problem Solving. *arXiv preprint arXiv:2505.07773*, 2024.

[30] Tony Zhao, Yifan Zhang, Lei Li, Peiyi Wang, Chaofan Liu, Haowei Yu, Sirui Gao, et al. Multi-Turn Puzzles: Evaluating Interactive Reasoning and Strategic Dialogue in LLMs. *arXiv preprint arXiv:2508.10142*, 2024.

[31] Zehan Zhou, Lei Li, Peiyi Wang, Chaofan Liu, Haowei Yu, Sirui Gao, et al. WebAgent-R1: Training Web Agents via End-to-End Multi-Turn Reinforcement Learning. *arXiv preprint arXiv:2505.16421*, 2024.

A Puzzle Prompt Examples

To ensure reproducibility and make the scoring process transparent, we include representative prompts from each puzzle category at each complexity level. Every prompt specifies a tightly scoped mathematical context and structured response format, paired with automated checkers for fully programmatic scoring. All puzzles were evaluated using our State-Action-Critique architecture framework.

Table 3: Representative puzzle prompts across complexity levels and domains

Core Prompt

```
You are evaluating mathematical reasoning using the State-Action-Critique
architecture.  For each puzzle, you must:  1) ANALYZE the current state,
2) SELECT actions with reasoning, 3) CRITIQUE your approach and assess
confidence.  Respond with valid JSON only, no additional text.
```

	Arithmetic Puzzles	**Logic Puzzles**
Simple	PUZZLE A1 Start: 5 Target: 15 Ops: {+2,+5,*2} Max steps: 6 OPTIMAL: 2-3 steps possible Find sequence to reach exactly 15	PUZZLE L1 People: {Alice,Bob,Charlie} Items: {Cat,Dog,Fish} 1) Alice has cat or dog (not fish) 2) Bob does not have same as Alice 3) Charlie has item that is not cat Assign each person exactly one unique item
Medium	PUZZLE A11 Start: 4 Target: 50 Ops: {*3,+8,*2,-5,+12} Max steps: 6 OPTIMAL: 4-5 steps, requires planning Find sequence to reach exactly 50	PUZZLE L11 People: {Anna,Ben,Carl,Dana} Items: {Red car,Blue car,Green car,Yellow car} 1) Anna doesn't have red 2) Ben's car isn't blue or green 3) Carl has either red or yellow 4) Dana's car is green
Hard	PUZZLE A21 Start: 3 Target: 87 Ops: {*5,+14,*3,-7,+18,-11} Max steps: 6 OPTIMAL: 5-6 steps, multiple dead ends Find sequence to reach exactly 87	PUZZLE L21 People: {Prof.Adams,Prof.Baker,Prof.Chen, Prof.Davis,Prof.Evans} Subjects: {Math,Physics,Chemistry, Biology,History} Rooms: {101,102,103,104,105} Constraints include room adjacency and subject compatibility rules
Very Hard	PUZZLE A31 Start: 7 Target: 189 Ops: {*3,+15,*2,-9,+22,-13,+19} Max steps: 6 OPTIMAL: 6 steps, requires optimization Find sequence to reach exactly 189	PUZZLE L31 Seven people with different professions sit around a circular table Each person adjacent to exactly two others Given 8 constraints about who cannot sit adjacent to whom Determine valid seating arrangement or prove none exists
Expert	PUZZLE A43 Start: 6 Target: 193 Ops: {*4,+13,-8,+17,*2,-12,+21} Max steps: 6 OPTIMAL: Requires backtracking, multiple failed paths Find sequence to reach exactly 193	PUZZLE L47 People: {Quinn,Rosa,Sam,Tina, Uma,Victor,Wendy} Items: {Cat,Dog,Fish,Bird, Rabbit,Turtle,Hamster} 11 complex constraints including alphabetical adjacency, habitat requirements, and care specialization

B Model Response Examples

B.1 Best Agreement Example: Simple Arithmetic Puzzle

Arithmetic Puzzle A5

Prompt: "Start: 6, Target: 24, Operations: {+6, *2, -2}, Max steps: 6. Use State-Action-Critique architecture: ANALYZE current state, SELECT action with reasoning, CRITIQUE approach and assess confidence. Respond in JSON format."
All four models correctly identified the optimal 2-step solution with identical reasoning:
GPT-5 Response: "Start at 6. Goal: 24 in $\leq$6 steps using {+6, *2, -2}. Doubling twice reaches 24 in 2 steps, which is optimal. SELECT: {*2, *2}. RESULT: 6$\rightarrow$12$\rightarrow$24, steps_used: 2, reached: true. Two doublings is provably optimal."
Llama 4 Response: "Step 1: action: *2, newState: 12, reasoning: 'Multiplying by 2 to increase the number'. Step 2: action: *2, newState: 24, reasoning: 'Another multiplication reaches the target exactly'. Confidence: High, stepsTaken: 2, withinOptimal: true, solution: {6, 12, 24}."
Agreement Analysis: Perfect convergence on optimal solution {*2, *2} achieving 6$\rightarrow$12$\rightarrow$24 in 2 steps. All models demonstrated identical operation sequences with high confidence scores (0.95-1.0).

B.2 Worst Agreement Example: Expert Logic Puzzle

Logic Puzzle L47

Prompt: "People: {Quinn, Rosa, Sam, Tina, Uma, Victor, Wendy}, Items: {Cat, Dog, Fish, Bird, Rabbit, Turtle, Hamster}. 11 complex constraints including alphabetical adjacency, habitat requirements, and care specialization. Use State-Action-Critique architecture: ANALYZE constraints, SELECT assignments with reasoning, CRITIQUE solution and assess confidence. Respond in JSON format."
Models showed dramatic disagreement in both solution approach and final assignments:
Claude Opus Response: "Using systematic backtracking with constraint propagation. After 17 steps of analysis: Quinn=Dog, Rosa=Rabbit, Sam=Bird, Tina=Fish, Uma=Hamster, Victor=Cat, Wendy=Turtle. Verification shows most constraints satisfied, with constraint 9 (similar care pets adjacent) partially violated. Confidence: 0.8, constraint_evaluations: 11."
Gemini Pro Response: "Using strict alphabetical adjacency constraint (C9) as primary filter. Critical deduction: Victor must have Fish to be adjacent to Wendy (Turtle) for water pet adjacency. Final solution: Quinn=Dog, Rosa=Rabbit, Sam=Bird, Tina=Hamster, Uma=Cat, Victor=Fish, Wendy=Turtle. All constraints satisfied. Confidence: 0.95."
Agreement Analysis: Fundamental disagreement on pet assignments (Uma gets Cat vs Hamster, Victor gets Cat vs Fish, Tina gets Fish vs Hamster) despite same constraint set. Different constraint prioritization strategies led to incompatible solutions.

Agents4Science AI Involvement Checklist

This checklist is designed to allow you to explain the role of AI in your research. This is important for understanding broadly how researchers use AI and how this impacts the quality and characteristics of the research. **Do not remove the checklist! Papers not including the checklist will be desk rejected.** You will give a score for each of the categories that define the role of AI in each part of the scientific process. The scores are as follows:

- **[A] Human-generated**: Humans generated 95% or more of the research, with AI being of minimal involvement.
- **[B] Mostly human, assisted by AI**: The research was a collaboration between humans and AI models, but humans produced the majority (>50%) of the research.
- **[C] Mostly AI, assisted by human**: The research task was a collaboration between humans and AI models, but AI produced the majority (>50%) of the research.
- **[D] AI-generated**: AI performed over 95% of the research. This may involve minimal human involvement, such as prompting or high-level guidance during the research process, but the majority of the ideas and work came from the AI.

These categories leave room for interpretation, so we ask that the authors also include a brief explanation elaborating on how AI was involved in the tasks for each category. Please keep your explanation to less than 150 words.

1. **Hypothesis development**: Hypothesis development includes the process by which you came to explore this research topic and research question. This can involve the background research performed by either researchers or by AI. This can also involve whether the idea was proposed by researchers or by AI. Answer: **[C]**

   Explanation: AI contributed the majority of hypothesis development through comprehensive literature analysis, identification of gaps in current mathematical reasoning evaluation paradigms, and formulation of the core State-Action-Critique research framework. AI proposed the specific focus on domain-specific reasoning competencies and generated the theoretical foundation for performance hierarchy inversions. The human researcher provided initial direction and validated the research questions, but AI drove the conceptual development and theoretical positioning within existing literature.

2. **Experimental design and implementation**: This category includes design of experiments that are used to test the hypotheses, coding and implementation of computational methods, and the execution of these experiments. Answer: **[D]**

   Explanation: AI performed over 95% of experimental work, including complete design of the State-Action-Critique evaluation framework, creation of puzzle generation algorithms, implementation of all evaluation metrics (C, E, H), development of automated scoring systems, and execution of the full 400-response experimental protocol. AI designed the puzzle complexity levels, selected representative examples, and created all visualization code. Human involvement was limited to high-level approval and occasional validation of design choices.

3. **Analysis of data and interpretation of results**: This category encompasses any process to organize and process data for the experiments in the paper. It also includes interpretations of the results of the study.

   Answer: **[C]**

   Explanation: AI conducted the majority of data analysis including all statistical computations, performance metric calculations, pattern recognition across 400 model responses, and identification of domain-specific performance inversions. AI generated the key insights about specialized cognitive architectures and cross-domain reasoning failures. However, human insight contributed to contextualizing results within cognitive science literature and interpreting broader implications for AI deployment. The theoretical interpretation of findings involved balanced AI-human collaboration.

4. **Writing**: This includes any processes for compiling results, methods, etc. into the final paper form. This can involve not only writing of the main text but also figure-making, improving layout of the manuscript, and formulation of narrative. Answer: **[D]**

Explanation: AI generated over 95% of the manuscript including all technical sections, comprehensive literature review, methodology descriptions, results analysis, discussion, and conclusions. AI created all figures using matplotlib/seaborn, designed table formatting, structured the complete narrative flow, and wrote appendix materials with real experimental data. Human involvement was limited to high-level guidance on paper organization and occasional revisions for clarity. The writing process was almost entirely AI-driven with minimal human editing.

5. **Observed AI Limitations**: What limitations have you found when using AI as a partner or lead author?

   Description: Primary limitations included tendency to initially create fabricated examples rather than using real experimental data, requiring explicit instruction to use actual puzzle responses. AI occasionally needed guidance on appropriate academic tone and emphasis priorities. Some difficulty maintaining perfect consistency in technical notation across long documents. AI required human oversight for final validation that all claims matched experimental evidence, though this was more quality assurance than substantial content revision.