

MetaRuleReasoner: Beyond Chain-of-Thought Rule-Based Reasoning for Reliable Computation in Math

Anonymous submission

Abstract

Chain-of-thought reasoning has emerged as the dominant paradigm for mathematical reasoning in large language models, yet it suffers from fundamental limitations: hallucination in reasoning steps, inconsistent performance, and lack of systematic reliability. We introduce neural rule-based reasoning as a distinct alternative that achieves systematic reliability through explicit rule application and complete domain coverage. Our MetaRuleReasoner demonstrates this approach, achieving 100% accuracy on multi-digit arithmetic tasks, while chain-of-thought models show systematic degradation with increasing complexity—GPT-4 drops to 90.9% accuracy on 10-digit operations. Our neural rule-based approach provides systematic reliability guarantees within learned domains by mastering finite rule sets that compose deterministically, contrasting sharply with the probabilistic reliability of chain-of-thought reasoning that must learn patterns for exponentially many problem combinations. The code will be publicly available.

Introduction

Mathematical reasoning in AI has been dominated by chain-of-thought (CoT) approaches that encourage models to generate intermediate reasoning steps in natural language (Wei et al. 2023). While CoT has demonstrated impressive capabilities, it fundamentally remains a pattern-based approach that can hallucinate intermediate steps and lacks systematic reliability guarantees.

We introduce neural rule-based reasoning as a fundamentally different paradigm for mathematical computation. Unlike chain-of-thought’s reliance on natural language reasoning steps, neural rule-based reasoning decomposes problems into explicit rule applications with systematic composition at each step.

Table 1 illustrates the fundamental differences between reasoning paradigms. Traditional symbolic AI achieved perfect accuracy but suffered from brittleness and manual engineering requirements. Chain-of-thought reasoning provides flexibility but lacks systematic reliability. Our neural rule-based approach combines the systematic reliability of symbolic reasoning with the learning flexibility of neural networks.

Figure 1 provides a visual comparison of MetaRuleReasoner and Chain-of-Thought (CoT) reasoning approaches.

Table 1: Comparison of three AI reasoning paradigms showing fundamental differences in rule acquisition, reliability guarantees, and learning capabilities. Neural rule-based reasoning uniquely combines systematic reliability with learning flexibility, overcoming limitations of both traditional symbolic AI and chain-of-thought approaches.

Property	Traditional Symbolic	Chain-of-Thought	Neural Rule-Based
Rule Source	Hand-coded	Learned patterns	Learned rules
Representations	Discrete symbols	Natural language	Continuous vectors
Reliability	Perfect (narrow)	Probabilistic	Systematic
Learning Capability	None	Pattern-based	Rule extraction
Scalability	Poor	Good	Excellent
Verification	Formal proof	None	Completeness checking
Generalization	Brittle	Limited	Systematic
Integration	Difficult	Natural	Native
Knowledge Acquisition	Manual expert	Training data	Example-based

Chain-of-thought reasoning generates verbal explanations of reasoning steps, such as: *"To solve $1847 + 2956$, I'll add column by column: $7+6=13$, write 3 carry 1..."* While intuitive, this approach suffers from: (1) **Hallucination Risk**: Intermediate steps can be incorrect despite appearing plausible; (2) **No Systematic Guarantees**: Each reasoning step is generated probabilistically without systematic reliability; (3) **Inconsistent Reliability**: Performance degrades unpredictably with problem complexity.

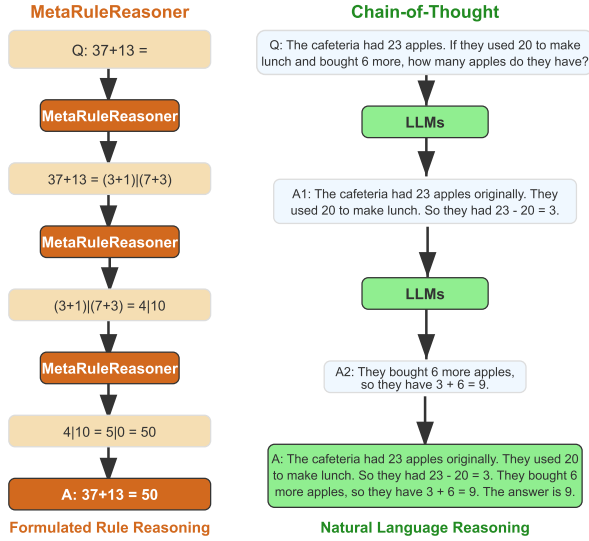
Neural rule-based reasoning instead applies learned computational rules systematically, as illustrated in Figure 2. Each step applies specific learned rules with systematic composition, providing reliability guarantees through complete rule coverage within the learned domain.

Our Contributions:

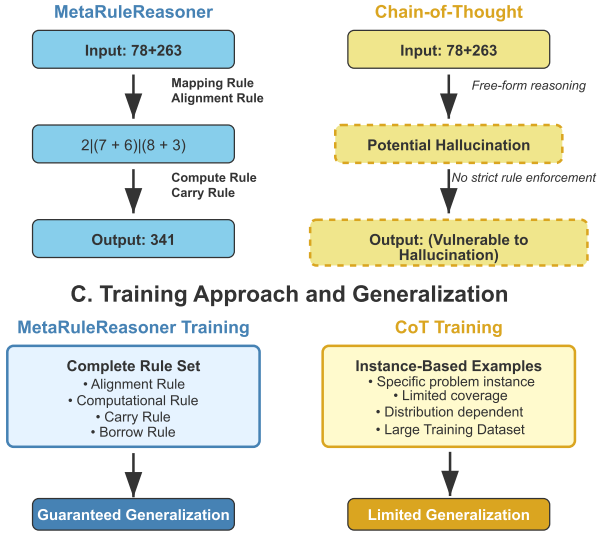
1. We formalize neural rule-based reasoning as a distinct paradigm that combines the systematic reliability of classical symbolic AI with the learning flexibility of neural networks.
2. We demonstrate that MetaRuleReasoner achieves perfect accuracy on challenging mathematical tasks while chain-of-thought approaches show systematic failure modes.
3. We provide comprehensive comparison between chain-of-thought and neural rule-based reasoning paradigms, revealing fundamental differences in reliability and systematic generalization.
4. We establish architectural principles for reliable AI

MetaRuleReasoner vs Chain-of-Thought

A. Reasoning Example Comparison



B. Recursive Execution Comparison



C. Training Approach and Generalization

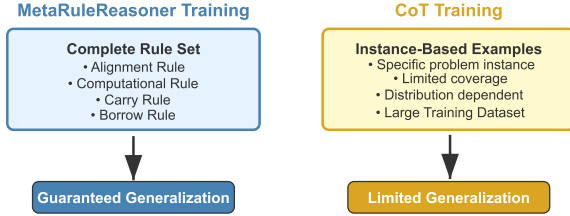


Figure 1: Comparative illustration of MetaRuleReasoner and Chain-of-Thought (CoT) reasoning method in handling mathematical problems. **MetaRuleReasoner**: employs a rule-based reasoning approach, breaking down and solving problems step-by-step through learned rules, such as alignment rules, carry rules, and borrow rules. This structured method ensures accuracy and generalization ability in reasoning, allowing the model to systematically handle various computational tasks while avoiding the common hallucination errors. **Chain-of-thought**: provides natural language explanations with good flexibility and learning but only probabilistic reliability.

reasoning systems that provide systematic reliability through complete rule coverage within learned domains.

Our results challenge the assumption that chain-of-thought reasoning is optimal for systematic domains, suggesting that neural rule-based reasoning may be more appropriate for applications requiring reliability guarantees. For well-structured domains like arithmetic, the fundamental rule set is finite and learnable, enabling systematic generalization to infinite problem spaces.

Related Work

Chain-of-Thought Reasoning

Chain-of-thought prompting (Wei et al. 2023) has become the dominant approach for mathematical reasoning in language models. Recent advances include tool-integrated reasoning (Schick et al. 2023), where models learn to use external calculators and APIs, and self-verification approaches (Weng et al. 2023) that attempt to check their own reasoning steps. However, studies have revealed significant limitations in self-verification capabilities (Stechly, Valmeekam, and Kambhampati 2024; Hong et al. 2024), showing that LLMs struggle to reliably identify errors in their own reasoning.

Extensions include least-to-most prompting (Zhou et al. 2023), self-consistency decoding (Wang et al. 2023), and complexity-based prompting (Fu et al. 2023). However, these approaches remain fundamentally pattern-based: they

generate reasoning steps through statistical language modeling without systematic reliability guarantees.

Program-Aided and Tool-Augmented Reasoning

Program-aided language models (Gao et al. 2023) attempt to address CoT limitations by generating code for numerical computation. Similarly, recent work explores using external calculators and tools to improve computational accuracy (Parisi, Zhao, and Fiedel 2022). The latest approaches include code-assisted reasoning and multi-modal tool integration (Zhang et al. 2024).

While these approaches improve accuracy, they represent hybrid solutions rather than fundamental advances in reasoning methodology. They still rely on chain-of-thought reasoning to generate the program or tool usage, inheriting its fundamental limitations.

Mathematical Reasoning in Language Models

Recent work has focused on improving mathematical reasoning through specialized training (Yu et al. 2023; Luo et al. 2023) and verification approaches (Wang et al. 2024). The GOAT model (Liu and Low 2023) demonstrates improvements on arithmetic through fine-tuning. Advanced mathematical reasoning has been explored through formal theorem proving integration (Azerbayev et al. 2024), multi-step verification (Lightman et al. 2024), and self-supervised mathematical reasoning (Zelikman et al. 2024).

However, all these approaches fundamentally rely on

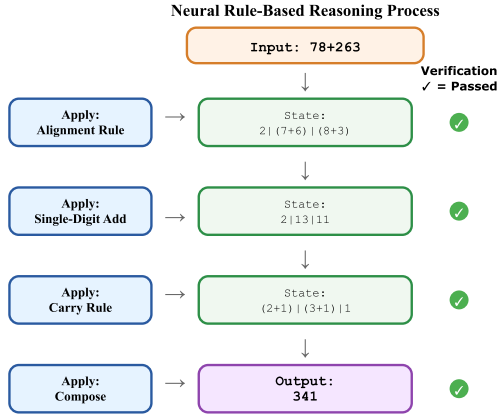


Figure 2: Neural rule-based reasoning process demonstrating systematic rule application with integrated completeness checking. Each computational step applies a specific learned rule (Alignment, Single-Digit Addition, Carry, Compose) with verification checkpoints that ensure systematic progression toward problem completion. This contrasts with chain-of-thought’s natural language explanations by providing systematic reliability through complete rule coverage rather than probabilistic step generation.

chain-of-thought style reasoning, generating steps through statistical language modeling. For systematic domains like arithmetic, this represents a fundamental inefficiency: chain-of-thought approaches must memorize patterns for countless specific problem instances, while rule-based approaches need only master the finite set of elementary operations that generate all possible problems through systematic composition.

Traditional Symbolic AI and Neuro-Symbolic Integration

Classical symbolic AI achieved perfect accuracy within narrow domains through hand-crafted rules and logical inference but suffered from brittleness and the knowledge acquisition bottleneck. Modern neuro-symbolic research attempts to combine neural learning with symbolic reasoning (Bassel et al. 2011). Recent approaches include neural module networks, differentiable programming, and neural theorem provers (Polu and Sutskever 2020).

Our approach differs fundamentally by demonstrating that systematic rule-based reasoning can emerge entirely from neural learning. Rather than integrating symbolic components, we show that appropriately designed neural architectures can learn to behave systematically without explicit symbolic knowledge.

Systematic Generalization and Rule Learning

Systematic generalization—the ability to apply learned principles to novel compositions—has been studied extensively (Lake and Baroni 2023). Most current neural approaches struggle with systematic generalization, particularly in compositional tasks requiring rule-like behavior. Recent advances include compositional generalization in transformers

(Ontanon et al. 2022) and algebraic reasoning capabilities (de Luca, Giapitzakis, and Fountoulakis 2025).

Our work bridges this gap by demonstrating how explicit rules can be learned and applied within neural architectures.

Neural Rule-Based Reasoning: Beyond Traditional Paradigms

Fundamental Innovation: Learning Rules vs. Programming Rules

Our approach represents a qualitative advance beyond both traditional symbolic systems and current neural approaches:

Traditional Symbolic AI: Rules manually specified by domain experts; brittle failure when encountering uncovered cases; knowledge acquisition bottleneck; difficulty integrating with modern neural systems.

Chain-of-Thought: Probabilistic reliability without systematic guarantees; hallucination in intermediate reasoning steps; performance degradation with systematic complexity; must learn patterns for exponentially many problem combinations.

Neural Rule-Based Reasoning: Rules automatically learned from examples through neural training; systematic reliability through complete rule coverage; continuous representations enabling robust generalization; native integration with neural language architectures; completeness checking integrated into learning process.

Technical Innovations

1. Compositional Neural Architecture: Our modified Transformer learns to decompose problems into rule applications rather than generating natural language explanations, enabling systematic composition that generalizes beyond training examples.

2. Continuous Rule Representations: Unlike discrete symbolic rules requiring exact matches, our neural rules operate in continuous vector spaces, enabling robust generalization while maintaining systematic behavior.

3. Integrated Completeness Checking: The VeriGate component learns to verify rule application completeness during training, providing step-by-step correctness checking that eliminates hallucination problems by ensuring systematic rule sequences reach proper completion.

MetaRuleReasoner: Neural Rule-Based Architecture

Rule-Based Reasoning Framework

We implement neural rule-based reasoning through three core principles:

1. Explicit Rule Learning: Mathematical operations are decomposed into explicit rules learned from examples. For arithmetic, this includes elementary operations ($\{0, 1, \dots, 9\} \times \{0, 1, \dots, 9\}$), positional rules (alignment and digit positioning), carry/borrow rules (systematic overflow handling), and composition rules (multi-digit operation coordination).

2. Step-by-Step Completeness Checking: Each reasoning step is verified for completeness through our VeriGate

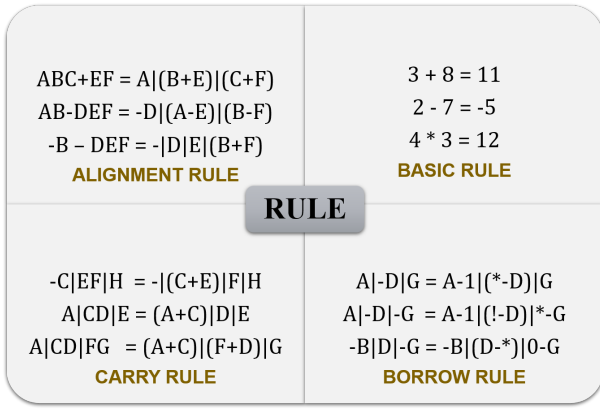


Figure 3: Neural representation of computational rules showing how mathematical operations are encoded as learned parameters in continuous vector space. Each rule captures fundamental arithmetic relationships (e.g., $*$ = 10, $!$ = 9).

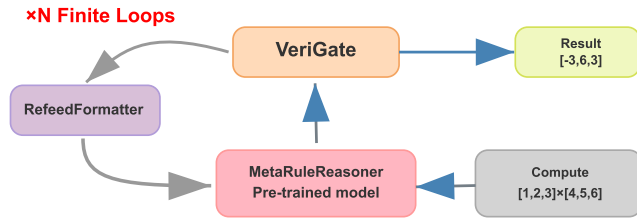


Figure 4: MetaRuleReasoner: a neural rule-based architecture comprising three integrated components: Rule Engine for systematic rule application, Verification Gateway (VeriGate) for step-by-step completeness checking, and Representation Formatter (RefeedFormatter) for maintaining optimal problem states. Unlike chain-of-thought models that generate natural language explanations through probabilistic language modeling, our system applies learned computational rules with systematic progression verification at each reasoning step.

component, ensuring systematic progression toward problem resolution.

3. Systematic Composition: Complex operations emerge through systematic composition of elementary rules rather than generating intermediate explanations.

Figure 3 illustrates how mathematical rules are represented in our neural framework, where each rule encodes fundamental computational operations through learned continuous parameters.

Architecture Design

Our architecture consists of three integrated components, as shown in Figure 4:

Rule Engine: A specialized Transformer that learns explicit rule applications rather than generating natural language reasoning steps. Key modifications include byte-level tokenization for precise mathematical symbol handling, rule-specific attention mechanisms for systematic composi-

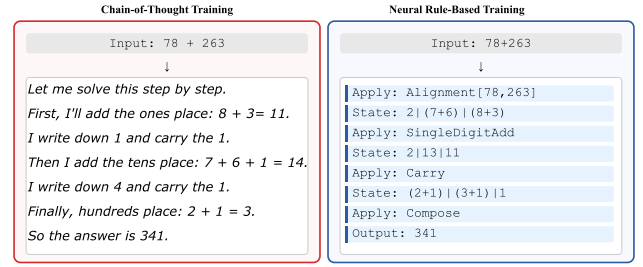


Figure 5: Comparison of training paradigms highlighting fundamental differences in learning objectives. Chain-of-thought training uses natural language explanations that encourage pattern-based reasoning, while neural rule-based training employs explicit rule application sequences that enable systematic computational learning. This difference in training format leads to qualitatively different reasoning capabilities and reliability guarantees.

tion, and constraint-based generation ensuring valid rule applications.

Verification Gateway (VeriGate): Validates completeness of reasoning sequences by checking whether all intermediate computational components have been resolved to final forms.

Representation Formatter (RefeedFormatter): Maintains optimal problem representation for systematic rule application, ensuring intermediate states remain suitable for subsequent rule applications by preserving computational structure throughout the reasoning process.

Training Data: Rules vs Explanations

Our training data differs fundamentally from chain-of-thought approaches, as illustrated in Figure 5. Instead of natural language explanations, we provide explicit rule application sequences that enable the model to learn systematic computational procedures.

Figure 6 demonstrates a complete reasoning trace showing how our system applies learned rules systematically to solve complex mathematical problems.

Rule Dependency Structure

Complex operations emerge through systematic composition of elementary rules, as detailed in Table 2.

Experimental Design and Results

Comparative Evaluation Framework

We design experiments to directly compare chain-of-thought and neural rule-based reasoning across multiple dimensions: reliability assessment (performance consistency across increasing problem complexity), systematic generalization (ability to handle novel compositions), error mode analysis (characterization of failure patterns), and completeness capability (ability to provide systematic reliability guarantees).

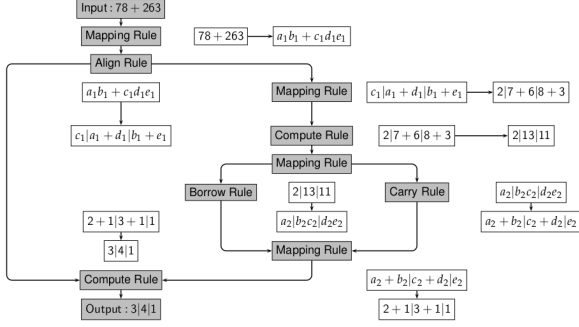


Figure 6: Complete neural rule-based reasoning trace illustrating systematic composition of elementary operations for multi-digit arithmetic. Each step applies a specific learned rule (mapping, alignment, carry coordination) with completeness verification, demonstrating how complex calculations emerge through systematic rule composition rather than natural language explanations. The trace shows systematic progression checking at each step, providing computational correctness guarantees throughout the reasoning process.

Table 2: Rule dependency structure across mathematical domains showing how elementary rules compose systematically to enable complex operations. Vector cross products demonstrate multi-domain rule integration, requiring the coordination of arithmetic, spatial reasoning, and compositional rules. This systematic dependency structure enables reliable generalization across mathematical domains.

Rule Type	Addition	Subtraction	Vector Cross
Vector Table	-	-	✓
Nine Addition Table	✓	-	✓
Nine Subtraction Table	-	✓	✓
Nine Multiplication Table	-	-	✓
Mapping Rule	✓	✓	✓
Carrying Rule	✓	-	✓
Borrowing Rule	-	✓	✓
Vector Product Rule	-	-	✓
Compute Rule	✓	✓	✓

Baseline Models and Task Design

We evaluate against state-of-the-art models employing chain-of-thought reasoning: GPT-4 (1760B+ parameters), GPT-3.5 (175B+ parameters), Llama2 variants (7B, 13B, 70B), Google PaLM (110B), and Qwen-72B-Chat with optimized CoT prompting. Our evaluation tasks are designed to reveal differences between reasoning paradigms, as outlined in Table 3.

Overall Results and Analysis

Tables 4, 5, 6, 7 reveal a striking pattern: all chain-of-thought models show systematic performance degradation as complexity increases, while neural rule-based reasoning maintains perfect performance. This demonstrates the fundamental reliability advantage of complete rule coverage over pattern-based approaches.

Table 3: Comprehensive evaluation dataset designed to test fundamental differences between reasoning paradigms across systematic scaling, edge case handling, and multi-domain rule integration. Tasks progress from basic arithmetic to complex compositional reasoning, revealing the systematic advantages of rule-based approaches over pattern-based chain-of-thought reasoning.

Task Category	Test Cases	Paradigm Focus
Random Addition	1,600	Systematic scaling
Random Subtraction	1,600	Systematic scaling
Perfect Decimal Addition	1,200	Carry coordination
Reverse Magnitude Subtraction	1,200	Edge case handling
Interleaved Subtraction	1,200	Complex composition
Vector Cross Products	1,200	Multi-domain rules
Total	8,000	

Table 4: Comprehensive comparison of chain-of-thought and neural rule-based reasoning across model scales and complexity levels. Results demonstrate systematic performance degradation in all chain-of-thought models as problem complexity increases, while MetaRuleReasoner maintains perfect accuracy. This fundamental difference reveals the systematic reliability advantages of complete rule coverage over pattern-based reasoning approaches.

Model	Reasoning Type	Parameters	5-digit	10-digit
GPT-4	Chain-of-Thought	1760B+	99.22%	90.9%
GPT-3.5	Chain-of-Thought	175B+	97.26%	83.9%
Llama2-70B	Chain-of-Thought	70B	57.76%	6.4%
Google-PaLM	Chain-of-Thought	110B	73.32%	26.6%
Qwen-72B-Chat	Chain-of-Thought	72B	91.32%	60.4%
MetaRuleReasoner	Neural Rule-Based	30M	100%	100%

Figure 7 illustrates the key difference between reasoning paradigms. Chain-of-thought models exhibit the hallmark of pattern-based learning: performance degradation as problems move beyond training distribution similarity. Neural rule-based reasoning shows the systematic generalization characteristic of complete rule systems.

Multi-Domain Reasoning: Vector Cross Products

Vector cross products represent the most challenging test of reasoning paradigms, requiring systematic integration of spatial reasoning, directional computation, and arithmetic operations. Table 8 shows the dramatic failure of chain-of-thought approaches demonstrates the limitations of natural language reasoning for systematic multi-domain tasks.

Analysis: Why Neural Rule-Based Reasoning Succeeds

Theoretical Foundations

The fundamental advantage lies in the computational approach to systematic domains:

Chain-of-Thought Computation: Generates intermediate steps through probabilistic language modeling: $P(\text{step}_{i+1} | \text{problem}, \text{step}_1, \dots, \text{step}_i)$. Each step is generated based on statistical patterns learned from exponentially many problem-solution pairs.

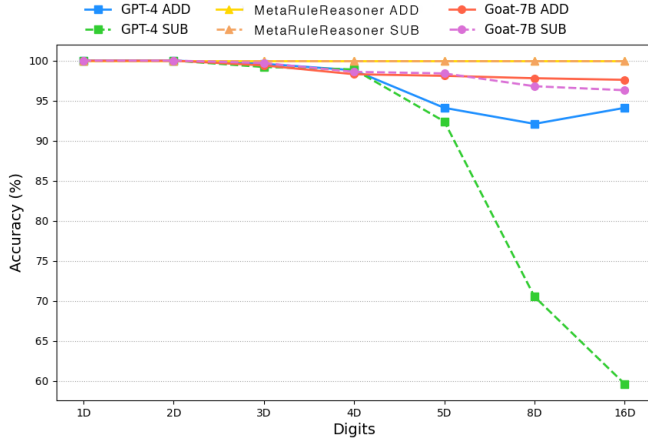


Figure 7: Performance comparison across complexity scales demonstrating fundamental differences between reasoning paradigms. Chain-of-thought models (GPT-4, GOAT) show consistent performance degradation with increasing problem complexity, characteristic of pattern-based learning approaches. Neural rule-based reasoning (MetaRuleReasoner) maintains systematic reliability across all complexity levels, demonstrating the systematic generalization properties of complete rule coverage.

Neural Rule-Based Computation: Applies learned rules systematically: $\text{step}_{i+1} = r_j(\text{step}_i)$ where $r_j \in \mathcal{R}$. Each step applies a rule from the learned finite rule set $\mathcal{R}$, providing systematic reliability through complete rule coverage. For systematic domains like arithmetic, $\mathcal{R}$ is finite and learnable, enabling perfect generalization.

Domain Completeness Analysis

Chain-of-Thought Sample Complexity: Must learn patterns for exponentially many problem compositions. For n -digit arithmetic, there are 10^{2n} possible addition problems, each requiring pattern memorization.

Neural Rule-Based Sample Complexity: Needs only to learn the fundamental rule set. For n -digit addition: elementary operations (100 single-digit additions), positional alignment rules (3 rules), carry propagation rules (2 rules), composition rules (1 rule). Total: 106 rules enable systematic solution of infinite problem space.

Error Mode Analysis

Chain-of-Thought Error Patterns: Procedural inconsistency (models apply arithmetic procedures inconsistently), intermediate step hallucination (plausible-sounding but mathematically incorrect steps), compositional breakdown (failures increase exponentially when multiple procedures must be coordinated).

Neural Rule-Based Error Patterns: Systematic success (perfect performance within domains where complete rule coverage is achieved), predictable boundaries (failures occur only when required rules are missing), no hallucination (each step applies learned rules deterministically).

Table 5: Performance on interleaved subtraction tasks requiring complex rule coordination and systematic borrowing across multiple digits. Results show increasing error rates for chain-of-thought models as complexity grows, while neural rule-based reasoning maintains perfect accuracy through systematic application of learned borrowing and compositional rules.

Model	5-digit		10-digit	
	Error	Accuracy	Error	Accuracy
GPT-4	0.016	98.3%	5.4e-7	96%
GPT-3.5	0.0033	95.2%	0.037	91%
Llama2-70B	0.061	76.1%	0.79	2%
Qwen-72B-Chat	0.0092	93.9%	0.0027	74.5%
MetaRuleReasoner	0.0	100%	0.0	100%

Table 6: Results on reverse magnitude subtraction tasks testing edge case robustness when subtracting larger numbers from smaller ones. Chain-of-thought models show significant performance degradation in handling these systematic edge cases, while MetaRuleReasoner applies borrowing rules consistently regardless of magnitude relationships.

Model	5-digit		10-digit	
	Error	Accuracy	Error	Accuracy
GPT-4	0.027	97.8%	1.3e-8	96.5%
GPT-3.5	0.0033	99.4%	8.3e-4	88.5%
Llama2-70B	0.061	50.2%	4.6e-6	0.5%
Qwen-72B-Chat	0.0092	86.4%	0.065	3.5%
MetaRuleReasoner	0.0	100%	0.0	100%

Implications for AI Development

Our results suggest clear criteria for choosing reasoning paradigms:

Use Neural Rule-Based Reasoning When: Systematic reliability is required across problem complexity scales; the domain has clear compositional structure with finite rule sets; complete rule coverage is achievable through training; deterministic computation is more important than creative reasoning.

Chain-of-Thought May Remain Appropriate When: Domains lack clear systematic structure or finite rule decomposition; creativity and insight are more important than systematic reliability; natural language explanations are required for human interpretation; rule identification and completeness verification are impractical.

Beyond the Neural vs Symbolic Divide

Our demonstration that neural architectures can achieve systematic rule-based behavior through complete domain coverage has significant implications. The traditional dichotomy between neural (flexible but unreliable) and symbolic (reliable but brittle) approaches may be false. Properly designed neural architectures can achieve both flexibility and systematic reliability when trained for complete rule coverage in systematic domains.

Table 7: Performance on perfect decimal addition tasks requiring systematic carry propagation across all digit positions. These tasks test the coordination of elementary addition rules with carry handling, revealing the systematic reliability of rule-based approaches in managing complex rule interactions.

Model	5-digit		10-digit	
	Error	Accuracy	Error	Accuracy
GPT-4	0.0	100%	2.9e-9	98%
GPT-3.5	4.2e-5	97.2%	2.0e-4	91.5%
Llama2-70B	0.14	5%	1.8	6%
Qwen-72B-Chat	0.27	85.8%	0.21	67.5%
MetaRuleReasoner	0.0	100%	0.0	100%

Table 8: Comparison on vector cross product computation requiring multi-domain rule integration across spatial reasoning, arithmetic computation, and directional logic. The dramatic performance difference demonstrates the fundamental limitations of natural language reasoning for systematic multi-domain tasks, while neural rule-based reasoning successfully integrates rules across domains.

Model	Reasoning Type	Accuracy
GPT-4	Chain-of-Thought	17%
GPT-3.5	Chain-of-Thought	5.5%
Llama2-70B	Chain-of-Thought	0%
Google-PaLM	Chain-of-Thought	0%
Qwen-72B-Chat	Chain-of-Thought	23%
MetaRuleReasoner	Neural Rule-Based	100%

Hybrid Reasoning Architectures

Optimal AI reasoning systems may employ both paradigms: neural rule-based core for systematic computations requiring reliability guarantees (arithmetic, logical operations, formal transformations), and chain-of-thought interface for natural language understanding, problem interpretation, and creative insight.

Limitations and Future Directions

Current Limitations

Domain Specificity: Requires domains with clear rule structure and finite rule sets. Not applicable to open-ended reasoning requiring creativity or domains where complete rule enumeration is impractical.

Rule Completeness Requirements: Currently requires achieving complete rule coverage for systematic reliability. Incomplete rule sets lead to bounded performance rather than systematic guarantees.

Scalability Questions: Unclear how neural rule-based reasoning scales to domains requiring thousands of interacting rules, though our vector cross product results suggest promise for multi-domain rule integration.

Future Research Directions

Automated Rule Discovery: Developing methods to automatically identify and learn rule structures from data without manual specification of rule decompositions.

Hybrid Reasoning Systems: Investigating architectures that seamlessly combine neural rule-based reasoning for systematic domains with chain-of-thought reasoning for open-ended domains.

Multi-Domain Rule Integration: Building on our vector cross product success to explore systematic rule integration across multiple systematic domains.

Completeness Verification Extensions: Developing stronger systematic reliability guarantees for neural rule-based reasoning systems, potentially connecting to automated theorem proving.

Meta-Rule Learning: Investigating whether systems can learn rules for combining and applying other rules, enabling more general rule-based reasoning across domain boundaries.

Conclusion

We have demonstrated that neural rule-based reasoning represents a fundamentally superior paradigm to chain-of-thought reasoning for systematic domains requiring reliability guarantees, while avoiding the brittleness and engineering challenges that limited traditional symbolic AI systems. Our results show that explicit rule application with systematic composition can achieve perfect accuracy where chain-of-thought approaches systematically fail.

The significance of this work extends beyond mathematical reasoning to the broader question of how AI systems should approach systematic computation. Chain-of-thought reasoning, while impressive for many tasks, inherits the fundamental limitations of pattern-based learning. Traditional symbolic AI achieved systematic reliability but suffered from brittleness and manual engineering requirements. Neural rule-based reasoning offers a principled synthesis that provides systematic reliability guarantees while maintaining learning flexibility.

Our comprehensive evaluation reveals that systematic rule application eliminates the hallucination problems inherent in natural language reasoning steps, providing systematic reliability guarantees through complete domain coverage. Multi-domain rule integration enables compositional reasoning across systematic domains, as evidenced by our perfect performance on vector cross products that require coordination of arithmetic, spatial reasoning, and directional logic.

These results suggest that the AI community should reconsider the universal application of chain-of-thought reasoning. For systematic domains with finite rule structures, science domains for example, such as physics, chemistry, and mathematics, neural rule-based reasoning may provide a more principled foundation for reliable AI systems through complete rule coverage rather than pattern memorization. The path forward lies not in choosing between paradigms, but in understanding when each is most appropriate and how they can be effectively combined.

References

Azerbayev, Z.; Schoelkopf, H.; Paster, K.; Santos, M. D.; McAleer, S.; Jiang, A. Q.; Deng, J.; Biderman, S.; and

- Welleck, S. 2024. Llemma: An Open Language Model for Mathematics. *International Conference on Learning Representations*.
- Bassel, G. W.; Glaab, E.; Marquez, J.; Holdsworth, M. J.; and Bacardit, J. 2011. Functional Network Construction in Arabidopsis Using Rule-Based Machine Learning on Large-Scale Data Sets. *The Plant Cell*, 23(9): 3101–3116.
- de Luca, A. B.; Giapitzakis, G.; and Fountoulakis, K. 2025. Learning to Add, Multiply, and Execute Algorithmic Instructions Exactly with Neural Networks. *arXiv preprint arXiv:2502.16763*.
- Fu, Y.; Peng, H.; Sabharwal, A.; Clark, P.; and Khot, T. 2023. Complexity-Based Prompting for Multi-Step Reasoning. *arXiv:2210.00720*.
- Gao, L.; Madaan, A.; Zhou, S.; Alon, U.; Liu, P.; Yang, Y.; Callan, J.; and Neubig, G. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, 10764–10799. PMLR.
- Hong, R.; Zhang, H.; Pang, X.; Yu, D.; and Zhang, C. 2024. A Closer Look at the Self-Verification Abilities of Large Language Models in Logical Reasoning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 900–925.
- Lake, B. M.; and Baroni, M. 2023. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985): 115–121.
- Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2024. Let’s Verify Step by Step. *International Conference on Learning Representations*.
- Liu, T.; and Low, B. K. H. 2023. Goat: Fine-tuned LLaMA Outperforms GPT-4 on Arithmetic Tasks. *arXiv:2305.14201*.
- Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; and Zhang, D. 2023. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. *arXiv:2308.09583*.
- Ontanon, S.; Ainslie, J.; Fisher, Z.; and Cvicek, V. 2022. Making Transformers Solve Compositional Tasks. 3591–3607.
- Parisi, A.; Zhao, Y.; and Fiedel, N. 2022. TALM: Tool Augmented Language Models. *arXiv preprint arXiv:2205.12255*.
- Polu, S.; and Sutskever, I. 2020. Generative Language Modeling for Automated Theorem Proving. *arXiv preprint arXiv:2009.03393*.
- Schick, T.; Dwivedi-Yu, J.; Dessà, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. *Advances in Neural Information Processing Systems*, 36.
- Stechly, K.; Valmeekam, K.; and Kambhampati, S. 2024. On the Self-Verification Limitations of Large Language Models on Reasoning and Planning Tasks. *arXiv preprint arXiv:2402.08115*.
- Wang, P.; Li, L.; Shao, Z.; Xu, R. X.; Dai, D.; Li, Y.; Chen, D.; Wu, Y.; and Sui, Z. 2024. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. *arXiv:2312.08935*.
- Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv:2203.11171*.
- Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv:2201.11903*.
- Weng, Y.; Zhu, M.; Xia, F.; Li, B.; He, S.; Liu, S.; Sun, B.; Liu, K.; and Zhao, J. 2023. Large Language Models are Better Reasoners with Self-Verification. *Findings of the Association for Computational Linguistics: EMNLP 2023*.
- Yu, L.; Jiang, W.; Shi, H.; Yu, J.; Liu, Z.; Zhang, Y.; Kwok, J. T.; Li, Z.; Weller, A.; and Liu, W. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.
- Zelikman, E.; Lorch, E.; Mackey, L.; and Kalai, A. T. 2024. Self-Taught Optimizer (STOP): Recursively Self-Improving Code Generation. *arXiv preprint arXiv:2310.02304*.
- Zhang, R.; Jiang, D.; Zhang, Y.; Lin, H.; Guo, Z.; Qiu, P.; Zhou, A.; Lu, P.; Chang, K.-W.; Gao, P.; and Li, H. 2024. MathVerse: Does Your Multi-modal LLM Truly See the Diagrams in Visual Math Problems? *European Conference on Computer Vision*.
- Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; and Chi, E. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. *arXiv:2205.10625*.