The Cliff Effect: Probing the Boundary Between Memorization and Reasoning in LLM Numerical Computation

Anonymous ACL submission

Abstract

Recent studies have documented "accuracy collapse" in Large Language Models on complex reasoning tasks – planning puzzles, multi-hop questions, mathematical word problems – fueling debate about whether LLMs can reason or merely pattern-match. However, these tasks conflate numerical computation with spatial reasoning, language parsing, and knowledge retrieval, leaving open whether failures stem from these auxiliary demands or from mathematical processing itself. We address this gap by probing the most fundamental level: can LLMs reliably compose elementary arithmetic operations? Using a carefully constructed benchmark that systematically varies difficulty across controlled dimensions – operation count, digit magnitude, and precision requirements – we evaluate four frontier models on 1,152 test cases spanning eight categories of numerical computation. We document a striking "cliff effect": accuracy collapses from 72.6% on easy tasks to 33.0% at medium difficulty – a 40 percentage point drop in a single step – then plateaus through harder levels. Statistical analysis confirms that easy-task performance differs significantly from all other difficulty levels, while medium, hard, and expert levels do not differ significantly from each other. This step-function pattern – strong performance followed by sudden collapse rather than gradual degradation – mirrors findings from puzzle-based evaluations but emerges here at the level of basic arithmetic, suggesting a general architectural limitation rather than task-specific failure. Error analysis reveals that 81.6% of incorrect answers are rounding issues rather than reasoning failures, though categories requiring exact symbolic manipulation (e.g., fraction addition) show 100% genuine calculation errors.

1 Introduction

Large Language Models can write poetry, summarize legal documents, and pass medical licensing exams. But ask one to compute "23/29 + 10/31 + 5/17 + 4/13 + 27/43" and report the exact result, and confidence in these systems begins to fracture. This paper investigates a deceptively simple question: can LLMs do math? The answer, we find, reveals something fundamental about how these systems process information.

The growing deployment of LLMs in domains requiring numerical accuracy – financial analysis, scientific research, healthcare, engineering – makes this question practically urgent. A model that appears competent on routine calculations but fails unpredictably on slightly harder variants poses genuine risks. Yet standard benchmarks rarely stratify performance by difficulty, leaving practitioners without clear guidance on where model capabilities end.

We move beyond asking whether LLMs can perform arithmetic to asking *where* their numerical capabilities break down, and what the pattern of breakdown reveals about underlying computational mechanisms. Our central hypothesis is that LLMs perform well on easy numerical tasks primarily through pattern matching against training data, while lacking the compositional reasoning capabilities required for genuine mathematical computation. If true, this predicts a specific empirical signature: strong performance on simple problems that resembles memorization, followed by catastrophic failure when problems exceed the training distribution – rather than the gradual degradation we would expect from a system that truly understands arithmetic.

To test this hypothesis, we developed a benchmark that systematically varies problem difficulty across controlled dimensions: the number of sequential operations, the magnitude of numbers involved, and the precision required in outputs. We evaluate four frontier models from two major families (Claude and GPT) across eight categories of numerical reasoning, from basic multiplication to compound interest calculations. Our strict evalua-

tion criterion – requiring exact answers rather than approximations – reflects the precision demands of real-world numerical applications.

The results reveal what we term the "cliff effect": model accuracy drops from 72.6% on easy tasks to 33.0% on medium difficulty – a collapse of nearly 40 percentage points in a single step. Performance then plateaus, with hard and expert tasks showing only modest additional decline. This cliff-then-plateau pattern is statistically robust ($p < 0.0001$ across multiple tests) and appears in both model families, suggesting a fundamental architectural limitation rather than a training artifact.

Critically, pairwise statistical comparisons show that easy-task performance differs significantly from all other difficulty levels, while medium, hard, and expert levels do not significantly differ from each other. This structure supports the memorization hypothesis: easy problems fall within a "zone of competence" where pattern matching succeeds, while all harder problems fall outside it, forcing models to attempt genuine computation with uniformly poor results.

This work makes three primary contributions. First, we provide **empirical characterization of the cliff effect**, documenting a consistent pattern of catastrophic performance collapse at the easy-medium difficulty boundary and providing quantitative evidence for the memorization-reasoning distinction in LLM numerical processing. Second, we offer a **methodological framework for difficulty-stratified evaluation** – our benchmark design, operationalizing difficulty through controlled parameter variation, offers a template for capability assessment that reveals boundaries rather than averages, applicable beyond numerical tasks. Third, we provide **decomposition of error types** – by distinguishing rounding errors from genuine calculation failures and first-step errors from accumulated mistakes, we offer a more nuanced picture of where and how LLMs fail at numerical tasks.

2 Background

2.1 Numeric Hallucinations as a Fundamental Limitation

Large Language Models have demonstrated remarkable capabilities across diverse natural language tasks, yet they exhibit a critical vulnerability in numerical reasoning that undermines their reliability in quantitatively-intensive applications. Research has documented that despite impressive performance on many benchmarks, LLM systems frequently generate false outputs and unsubstantiated answers when processing numerical information (Shi et al., 2023). Comprehensive reviews have characterized hallucination as content that lacks sense or fidelity to provided sources – a challenge that remains persistent for LLMs in contexts where accuracy is paramount (Ji et al., 2023).

The severity of this limitation becomes apparent in high-stakes domains. Studies in clinical settings have revealed that LLMs produce inaccurate or misleading quantitative information at rates comparable to or exceeding traditional search engines, even when these models demonstrate sophisticated language understanding (Singhal et al., 2023). Medical LLMs exhibit statistically significant error rates in numerical responses and clinical recommendations despite their advanced architectures (Nori et al., 2023). Systematic analyses have demonstrated that hallucinations occur in over half of LLM outputs in certain domains (Ji et al., 2023), with recent work documenting that models hallucinate at least 69% of the time on factual queries in some topics (Dahl et al., 2024).

A fundamental insight emerging from this literature is that LLMs process numerical data differently than traditional computational systems. Unlike calculators or programming languages that execute explicit algorithms on numerical representations, LLMs handle numbers as textual tokens – sequences of characters processed through the same attention mechanisms applied to natural language (Shao et al., 2025). This architectural choice leads to systematic errors in arithmetic operations and numerical fact recall, with error patterns that differ qualitatively from human mathematical mistakes.

2.2 Reframing Numeric Errors as Metacognitive Failures

We propose reframing numeric hallucinations not as isolated computational errors but as symptoms of limited metacognitive capability – failures in what might be termed computational self-awareness. LLMs struggle with explicit confidence reporting, often producing poorly calibrated estimates even when they internally track aspects of their reliability (Steyvers and Peters, 2025). When models produce incorrect numerical answers, they typically lack adequate metacognitive mechanisms to recognize errors or appropriately flag uncertainty to users.

Recent work has begun exploring computational

approaches to metacognition. Azaria and Mitchell (2023) demonstrate that LLMs' internal states contain information about output reliability, training classifiers on hidden layer activations to detect confabulations. Kuhn et al. (2023) develop semantic uncertainty measures that achieve better calibration by computing uncertainty at the level of meaning rather than specific word sequences. These approaches suggest that signals relevant to numerical reliability exist within model computations, even if current systems lack mechanisms to act on them.

The metacognitive framing has practical implications. Without such capabilities, LLMs cannot reliably distinguish between problems they can solve and those that exceed their competence, creating the conditions for the "illusion of competence" we investigate in this work.

2.3 Why Difficulty Scaling Reveals Reasoning Boundaries

Central to our methodology is the hypothesis that controlled difficulty scaling can reveal the boundary between memorization and genuine reasoning. Standard LLM evaluation typically reports accuracy across problem sets without stratifying by difficulty. Yet the distribution of problem difficulty in training data is highly non-uniform: simple arithmetic facts, common percentage calculations, and familiar formula applications appear far more frequently than complex multi-step problems with unusual parameters. If LLMs acquire numerical capabilities primarily through pattern memorization, we would expect strong performance on problems resembling training examples and sharp degradation when problems diverge from familiar patterns.

This prediction aligns with observations about LLM generalization more broadly. Holtzman et al. (2020) document that language models exhibit systematic biases reflecting corpus statistics, with generation quality degrading when prompted toward low-probability regions. Lin et al. (2022) show that models systematically reproduce human misconceptions that appeared in training data, suggesting imitation rather than reasoning.

Recent empirical work provides strong support for this approach. Shojaee et al. (2025) employ controllable puzzle environments – Tower of Hanoi, River Crossing, and Blocks World – that allow precise manipulation of compositional complexity while maintaining consistent logical structures. Their findings reveal a striking pattern: Large Reasoning Models exhibit complete accuracy collapse beyond model-specific complexity thresholds, with reasoning effort paradoxically decreasing as problems become harder. Mirzadeh et al. (2025) demonstrate similar fragility in mathematical reasoning, showing that simply altering numerical values while preserving problem structure causes systematic performance degradation. More strikingly, adding a single irrelevant clause to math word problems causes up to 65% performance drops across state-of-the-art models.

The compositionality gap literature reinforces these findings. Press et al. (2023) show that language models can correctly answer individual subquestions while failing to compose answers into correct solutions – and importantly, that this gap does not decrease with model scale. This finding directly contradicts the intuition that larger models should exhibit more robust compositional reasoning, instead suggesting that scaling primarily improves memorization capacity while leaving compositional mechanisms unchanged.

Our work extends this research program in a crucial direction: by focusing on elementary arithmetic rather than complex puzzles or word problems, we can isolate numerical computation from other cognitive demands. Arithmetic provides a cleaner test case – a problem like "compute 23/29 + 10/31 + 5/17" requires only the application of well-defined algorithms that any calculator executes perfectly. If LLMs fail on such tasks in difficulty-dependent ways, we have isolated evidence about numerical processing specifically.

Difficulty	Ops	Digits	Precision
Easy	1	2	1 decimal
Medium	2	4	2 decimals
Hard	3	6	2 decimals
Expert	4	8	3 decimals

Table 1: Operationalization of difficulty levels through number of sequential operations, digit magnitude, and required decimal precision.

3 Methodology

3.1 Benchmark Design and Difficulty Operationalization

The core innovation of our approach is the operationalization of "difficulty" as a composite of three measurable parameters that scale together across four levels (Table 1).

Operations refers to the number of sequential computational steps required (e.g., a single divi-

sion vs. a chain of four divisions). *Digit magnitude* controls the size of numbers involved, directly increasing the computational load. *Decimal precision* determines the granularity of expected outputs, testing whether models maintain precision through calculations or introduce rounding errors.

This design enables us to distinguish between two hypotheses: (1) if LLMs possess genuine arithmetic reasoning capabilities, performance should degrade gradually as difficulty increases; (2) if LLMs primarily rely on memorized patterns from training data, we expect strong performance on easy problems (likely encountered during training) with dramatic collapse when problems exceed the distribution of training examples.

3.2 Test Categories

We evaluate eight categories of numerical reasoning, each probing different computational capabilities:

Division Precision. Multi-step division chains that test precision maintenance. Easy problems involve single divisions (e.g., \$1000 $\div$ 7); expert problems chain four sequential divisions with larger operands.

Compound Calculations. Sequential percentage changes applied to a base value, simulating real-world financial scenarios. The number of percentage operations scales with difficulty, and percentages include decimal precision at higher levels.

Weighted Averages. Computing overall averages from multiple groups with different sizes and means. Difficulty scales via the number of groups (2–5) and the precision of group statistics.

Geometry with Unit Conversion. Cylindrical volume calculations followed by unit conversion (cubic feet to gallons). This category maintains two operations across all difficulties but scales the precision of input dimensions.

Large Multiplication. Direct multiplication of integers, scaling from 2-digit $\times$ 2-digit (easy) to 8-digit $\times$ 8-digit (expert). This category isolates pure arithmetic capability.

Inverse Percentage Calculations. Given a final price after sequential percentage changes, recover the original price. This requires algebraic reasoning beyond forward calculation.

Fraction Addition. Adding fractions with coprime denominators, requiring least common denominator computation. The number of fractions increases with difficulty (2–5 fractions), and denominator magnitude scales accordingly.

Compound Interest. Standard compound interest calculations with monthly compounding. Difficulty scales via principal magnitude, rate precision, and time period (6–24 months).

3.3 Test Generation and Evaluation

Test cases are generated programmatically using parameterized templates with controlled randomization. Each test case includes: (1) the natural language problem statement, (2) the exact expected answer computed at high precision, (3) metadata including category, difficulty level, and generation parameters. To ensure reproducibility, all experiments use a fixed random seed (seed=42). For each category-difficulty combination, we generate 3 test cases, yielding 96 unique problems (8 categories $\times$ 4 difficulties $\times$ 3 instances).

The generation framework avoids "nice" numbers that might appear frequently in training data. For instance, division problems use arbitrary dividends and divisors rather than round numbers, and percentage changes include decimal components at medium difficulty and above.

We employ a strict exactness criterion for evaluation. A response is marked correct only if the extracted numerical answer matches the expected value within floating-point tolerance (relative error $< 10^{-9}$). This stringent standard is deliberate: in high-stakes applications, approximate answers are often as problematic as entirely wrong ones.

3.4 Models and Experimental Setup

We evaluate four frontier models spanning two major LLM families: Claude Sonnet 4.5, Claude Haiku 4.5, GPT-4.1, and GPT-4.1-mini. This selection enables both within-family comparisons (standard vs. compact variants) and cross-family comparisons (architectural and training differences).

Each model is evaluated at three temperature settings: 0.0 (deterministic), 0.5 (balanced), and 1.0 (high variability). All models receive identical system prompts emphasizing exact computation. The full benchmark comprises 1,152 individual API calls (96 problems $\times$ 4 models $\times$ 3 temperatures).

4 Results

4.1 Overall Performance and the Cliff Effect

Across all 1,152 test cases, models achieved an overall accuracy rate of 38.28% under our strict exactness criterion. However, this aggregate figure masks a dramatic performance gradient across

Difficulty	Accuracy	N
Easy	72.6%	288
Medium	33.0%	288
Hard	26.7%	288
Expert	20.8%	288

Table 2: Overall accuracy rates across difficulty levels, demonstrating the cliff effect with a 39.6 percentage point drop from easy to medium difficulty.

Comparison	p-value	Sig.
Easy vs. Medium	< 0.0001	***
Easy vs. Hard	< 0.0001	***
Easy vs. Expert	< 0.0001	***
Medium vs. Hard	0.6097	ns
Medium vs. Expert	0.0061	**
Hard vs. Expert	0.5792	ns

Table 3: Pairwise statistical comparisons (Bonferroni-corrected) showing easy-task performance differs significantly from all other levels, while medium, hard, and expert levels largely do not differ from each other.

difficulty levels (Table 2).

The performance drop from easy to expert tasks spans 51.7 percentage points – a 71.3% relative decline. Critically, the majority of this decline (39.6 percentage points) occurs in a single step: from easy to medium difficulty. Performance then plateaus, with only modest additional decline from medium through expert levels.

This cliff-then-plateau pattern is statistically robust. A chi-square test of independence confirms that difficulty level significantly affects performance ($\chi^2 = 200.08$, $p < 0.0001$). The Kruskal-Wallis H-test corroborates this finding ($H = 199.91$, $p < 0.0001$). Spearman correlation analysis reveals a moderate negative relationship between difficulty and accuracy ($\rho = -0.371$, $p < 0.0001$).

Pairwise comparisons with Bonferroni correction illuminate the structure of this effect (Table 3). The pattern is clear: performance differs significantly between easy and all other levels, but differences among medium, hard, and expert are largely non-significant. This supports the interpretation that models possess a qualitatively different capability on easy tasks – potentially pattern matching or memorization – that fails to generalize to harder variants.

4.2 Model and Family Comparisons

Both model families exhibit the cliff effect, though with notable differences in baseline performance

Family	Overall	Easy	Expert	Drop
Claude	47.0%	81.9%	27.1%	54.9pp
GPT	29.5%	63.2%	14.6%	48.6pp

Table 4: Model family performance comparison showing both Claude and GPT exhibit substantial accuracy collapse from easy to expert difficulty.

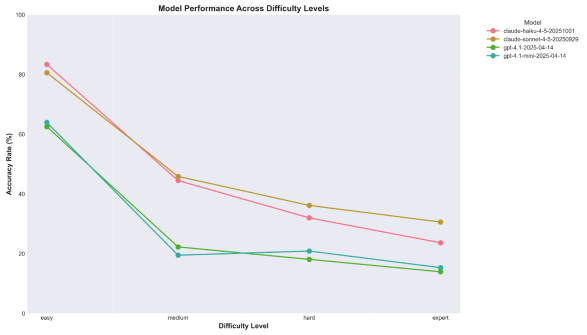

Figure 1: Model performance trajectories across difficulty levels. All four models exhibit the cliff effect, with sharp decline from easy to medium followed by plateau. Claude models consistently outperform GPT models at all difficulty levels.

(Table 4). Figure 1 visualizes performance trajectories across all four models.

Claude models outperform GPT models at every difficulty level, with particularly strong performance on easy tasks (81.9% vs. 63.2%). However, both families show substantial collapse: Claude's 54.9 percentage point drop and GPT's 48.6 percentage point drop both represent catastrophic performance degradation.

At the individual model level, Claude Sonnet 4.5 achieves 48.3% overall accuracy (80.6% easy, 30.6% expert), while Claude Haiku 4.5 achieves 45.8% (83.3% easy, 23.6% expert). GPT-4.1 achieves 29.2% (62.5% easy, 13.9% expert) and GPT-4.1-mini achieves 29.9% (63.9% easy, 15.3% expert). Notably, the compact models within each family perform comparably to their larger counterparts, suggesting that model scale alone does not explain the cliff effect.

4.3 Category-Specific Patterns

Performance varies substantially across task categories, with patterns that illuminate the memorization hypothesis. Figure 2 presents accuracy rates across all category-difficulty combinations.

We observe a striking divergence between "simple" categories (arithmetic, division precision) and "complex" categories (compound calculations, fi-

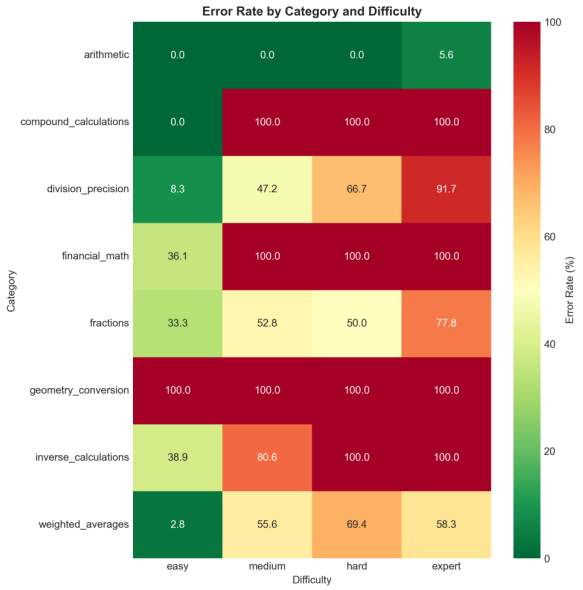

Figure 2: Error rate heatmap by category and difficulty. Arithmetic maintains near-zero error rates across difficulties, while compound calculations and financial math show complete failure beyond easy level. Geometry conversion fails uniformly due to precision requirements with irrational numbers.

Error Type	Freq.	%
Final-step error	534	46.4%
Correct	441	38.3%
First-step error	159	13.8%
Intermediate error	10	0.9%
Arithmetic error	8	0.7%

Table 5: Distribution of error types across all test cases, with final-step errors (primarily rounding issues) dominating the error profile.

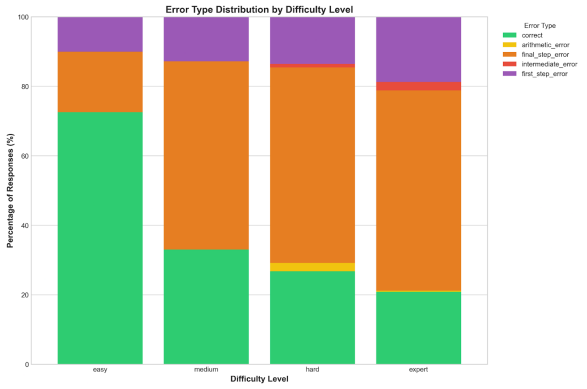

Figure 3: Error type distribution by difficulty level. The proportion of correct responses (green) decreases sharply from easy to medium, then plateaus. First-step errors (purple) increase with difficulty, indicating harder problems cause immediate pattern-matching failure.

nancial math, fractions). Simple arithmetic – the category most likely to appear verbatim in training data – shows the highest easy-task performance (approximately 100%) and the most graceful degradation (44.4pp drop to expert). Complex multi-step problems, which require genuine compositional reasoning, show more severe collapse (69.4pp drop) and near-floor expert performance (7.4%).

Individual category accuracy rates reveal further patterns. Arithmetic achieves 98.6% overall accuracy, maintaining near-perfect performance even at expert level. Weighted Averages achieves 53.5%, with high easy-task accuracy degrading moderately. Division Precision and Fractions each achieve 46.5%, though with different error profiles (discussed below). Compound Calculations achieves only 25.0%, with complete failure beyond easy level. Financial Math achieves 16.0%, and Geometry Conversion achieves 0% across all difficulties – the latter reflecting our strict evaluation criterion applied to calculations involving irrational numbers (π).

4.4 Error Pattern Analysis

To understand *how* models fail, we classified all 1,152 responses by error type (Table 5). Figure 3 visualizes error type distribution across difficulty levels.

The dominance of final-step errors (46.4%) initially suggests models reason correctly but fail at the last step. However, first-step error rates vary dramatically by difficulty: 10.1% (Easy), 12.8% (Medium), 13.5% (Hard), and 18.8% (Expert). The easy-to-expert difference is statistically significant ($p = 0.026$, Bonferroni corrected). This finding supports the hypothesis that harder problems cause models to fail at the outset of reasoning – consistent with pattern-matching failure when problems fall outside training distributions.

Model-specific error patterns reveal architectural differences. GPT models exhibit substantially higher first-step error rates across all difficulty levels (16.7–31.9%) compared to Claude models (0–11.1%), suggesting they more frequently fail to initiate correct reasoning chains. Claude models show lower baseline first-step errors but steeper increases with difficulty.

4.5 Rounding vs. Calculation Failures

Given the high prevalence of final-step errors, we conducted detailed analysis of the 711 incorrect responses to distinguish formatting issues from gen-

uine reasoning failures. A striking 81.0% of errors classified as "incorrect" are rounding or precision issues – cases where models computed substantively correct answers but failed to match our strict exactness criterion. Only 18.4% represent genuine calculation failures (small errors 5–20%: 2.1%; large errors 20–100%: 1.4%; precision errors 1–5%: 0.6%).

This finding has important implications. First, it suggests our strict evaluation criterion may overstate failure rates – models are often computing substantively correct answers but failing to format outputs precisely. However, maintaining strict evaluation is defensible: in high-stakes applications, small errors matter. Second, the 18.4% genuine calculation error rate still represents a substantial failure mode, and these errors are not uniformly distributed.

Category-level analysis reveals where true calculation failures concentrate. Fractions stand out: 100% of fraction errors are genuine calculation failures, not formatting issues. This category requires finding common denominators and maintaining exact rational arithmetic – operations that cannot be approximated. Division Precision also shows high true-error rates (39.0%), reflecting accumulated precision loss in multi-step division chains. In contrast, Arithmetic shows 100% rounding issues (no true calculation errors), and Geometry Conversion shows 99.3% rounding issues.

4.6 Effect Size and Practical Significance

Beyond statistical significance, we assessed practical significance via effect size measures. Cohen's h for the easy-versus-expert accuracy difference is 1.091, indicating a large effect that substantially exceeds conventional thresholds ($h > 0.8$). The effect is robust across both model families and persists when controlling for temperature settings. Temperature variations (0.0, 0.5, 1.0) produced modest effects on output consistency but did not alter the fundamental cliff pattern.

5 Discussion

5.1 Evidence for the Memorization-Reasoning Boundary

The cliff effect we observe is consistent with the hypothesis that LLMs rely heavily on pattern matching from training data rather than executing genuine mathematical reasoning. Several features of our results support this interpretation.

The cliff structure itself. If LLMs possessed robust arithmetic reasoning capabilities, we would expect gradual performance degradation as problems become more complex – each additional operation or digit introducing incremental difficulty. Instead, we observe a discontinuous pattern: strong performance on easy tasks (72.6%), catastrophic collapse at medium difficulty (33.0%), then a plateau through hard and expert levels. This step-function pattern suggests a qualitative transition – from problems that can be solved via pattern recognition to problems that cannot – rather than a quantitative accumulation of difficulty.

The pairwise comparison structure. Statistical comparisons reinforce this interpretation. Easy-task performance differs significantly from all other difficulty levels ($p < 0.0001$), but medium, hard, and expert levels do not differ significantly from each other. This is precisely what we would expect if easy problems fall within a "memorization zone" while all harder problems fall outside it, forcing models to attempt genuine computation with uniformly poor results.

Category-specific patterns. The divergence between simple and complex categories provides further evidence. Arithmetic – the category most likely to appear verbatim in training corpora – shows near-perfect easy-task accuracy and graceful degradation. Fraction addition, which requires compositional reasoning (finding common denominators, maintaining exact rational representations) that is unlikely to be directly memorized, shows 100% genuine calculation failures.

First-step error patterns. The increase in first-step errors from easy to expert difficulty (10.1% $\rightarrow$ 18.8%) indicates that models increasingly fail to even begin correct reasoning on harder problems. If models possessed genuine reasoning capabilities, we would expect them to start correctly but accumulate errors through multi-step chains. Instead, harder problems cause immediate failure – consistent with pattern-matching systems encountering unfamiliar inputs.

5.2 The Illusion of Competence

Our results expose a troubling phenomenon we term the "illusion of competence": LLMs can appear highly capable on numerical tasks that happen to resemble their training distribution, while possessing little ability to generalize beyond it.

Consider a practitioner evaluating an LLM for financial calculations. Testing with simple problems

– single-step percentages, small-number arithmetic, familiar formula applications – might yield impressive results (our models achieved 72.6% accuracy on easy tasks, with Claude reaching 81.9%). This could reasonably lead to deployment decisions. Yet the same models fail catastrophically on problems only slightly more complex (33.0% at medium difficulty), potentially introducing errors into high-stakes calculations.

The illusion is particularly dangerous because the boundary between competence and failure is not transparent. A model that correctly computes "What is 15% of $1,000?" may fail on "A product increased by 23.7%, then decreased by 11.4%. What is the net change?" – despite both being "percentage problems" from a user's perspective. Users have no reliable way to predict which problems fall within the model's effective capability.

Our findings argue for difficulty-stratified evaluation as a standard practice. Benchmarks that test only within the easy zone will systematically overestimate real-world reliability. Conversely, benchmarks that aggregate across difficulty levels may obscure the sharp competence boundary that practitioners need to understand.

5.3 Implications for AI Safety and Governance

The cliff effect has implications beyond academic interest. As LLMs are deployed in domains requiring numerical accuracy – financial services, healthcare, scientific research, engineering – the illusion of competence poses genuine risks.

Our findings suggest several governance considerations. First, deployment decisions for numerically-intensive applications should require difficulty-stratified evaluation, not aggregate accuracy metrics. Second, the unpredictability of the competence boundary argues for human oversight in numerical workflows. Third, if current architectures fundamentally lack compositional arithmetic capability – as the cliff pattern and fraction-error findings suggest – then improving numerical reliability may require architectural innovation rather than scaling or fine-tuning alone.

6 Conclusion

This paper investigated the boundary between memorization and reasoning in LLM numerical computation, finding evidence for a sharp capability cliff that has implications for both scientific understanding and practical deployment.

Our benchmark evaluation of four frontier models across 1,152 test cases reveals a consistent pattern: LLMs achieve 72.6% accuracy on easy numerical tasks but collapse to 33.0% when difficulty increases even modestly – a 40 percentage point drop in a single step. Performance then plateaus, with hard and expert tasks showing only modest additional decline. This cliff-then-plateau pattern is statistically robust and appears in both model families, suggesting fundamental architectural limitations.

The structure of this collapse supports the hypothesis that LLMs rely primarily on pattern matching rather than compositional reasoning for numerical tasks. Easy problems, which likely resemble training examples, elicit strong performance; harder problems produce uniformly poor results regardless of how much harder they become. The finding that medium, hard, and expert difficulty levels do not differ significantly from each other – while all differ significantly from easy – suggests a qualitative transition rather than gradual degradation.

For practitioners, our findings counsel caution: impressive performance on simple tests does not guarantee reliability on harder variants, and the boundary between competence and failure is neither transparent nor predictable from problem surface features. For researchers, the cliff offers a tractable phenomenon for investigating the memorization-reasoning distinction and developing systems with more robust numerical capabilities. For policymakers, our methodology provides a template for capability evaluation that reveals boundaries rather than averages – information essential for governing systems increasingly embedded in consequential decisions.

Limitations

Several limitations constrain the generalizability of our findings.

Model selection. We evaluated four models from two families. While these represent frontier systems, the LLM landscape is diverse. Other architectures (e.g., mixture-of-experts models, reasoning-specialized systems like o1) may exhibit different patterns.

Task coverage. Our eight categories, while diverse, do not exhaust mathematical reasoning. We did not test geometry proofs, algebraic manipula-

tion, calculus, or statistical inference. The cliff effect's generality across mathematical domains remains to be established.

Difficulty operationalization. Our difficulty scaling combines operation count, digit magnitude, and decimal precision. Other operationalizations – semantic complexity, variable substitution, word problem structure – might reveal different patterns.

Evaluation stringency. Our strict exactness criterion, while defensible, may not reflect all use cases. Some applications tolerate approximation; others require exactness beyond our criterion. The 81.6% rounding-error finding suggests our headline accuracy figures should be interpreted cautiously.

Sample size per cell. With 3 test cases per category-difficulty combination, some category-specific findings rest on limited observations. The overall cliff pattern is robust, but category-level conclusions warrant replication.

Prompt sensitivity. We used a single prompt template emphasizing exactness. Alternative prompts – chain-of-thought instructions, few-shot examples, different formatting requests – might elicit different performance profiles.

Ethical Considerations

This work investigates fundamental limitations in LLM numerical reasoning, with several ethical dimensions warranting discussion.

Potential Benefits. By documenting the cliff effect and the boundary between apparent competence and genuine capability, this research can inform safer deployment practices. Practitioners may use our findings to implement appropriate human oversight in numerically-intensive applications, potentially preventing errors in high-stakes domains such as finance, healthcare, and engineering.

Potential Risks. We acknowledge that detailed characterization of LLM failure modes could theoretically be misused to exploit model weaknesses. However, we believe the benefits of transparency – enabling informed deployment decisions and motivating architectural improvements – outweigh these risks.

Deployment Implications. Our findings suggest that benchmark performance on easy problems may dramatically overstate real-world reliability. We urge practitioners deploying LLMs in numerically-sensitive contexts to conduct difficulty-stratified evaluation rather than relying on aggregate accuracy metrics, and to maintain human oversight for calculations exceeding routine complexity.

Reproducibility Commitment. To support verification and extension of our findings, we commit to releasing our benchmark generation code, test cases, and evaluation scripts upon publication.

Acknowledgements

We used AI language model assistance (Claude 4.5 Opus) for proofreading and general language corrections of this manuscript.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. *arXiv preprint arXiv:2304.13734*.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Ho Shu Chan, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of GPT-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Jiandong Shao, Yao Lu, and Jianfei Yang. 2025. Benford's curse: Tracing digit bias to numerical hallucination in LLMs. *arXiv preprint arXiv:2506.01734*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. *arXiv preprint arXiv:2302.00093*.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking. *SuperIntelligence - Robotics - Safety & Alignment*, 2(6).

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, and 1 others. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Mark Steyvers and Megan A K Peters. 2025. Metacognition and uncertainty communication in humans and large language models. *arXiv preprint arXiv:2504.14045*.

A Performance Degradation Analysis

Figure 4 presents relative performance normalized to easy tasks and absolute performance drop, comparing the two model families.

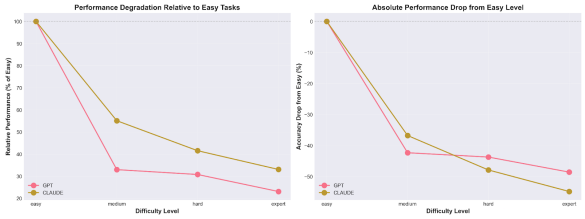

Figure 4: Performance degradation relative to easy tasks (left) and absolute performance drop from easy level (right). Both families show sharp initial decline, with GPT showing steeper relative degradation.

B Individual Model Performance

Table 6 presents detailed accuracy rates for each model across all difficulty levels.

Model	Overall	Easy	Med	Hard	Expert
Claude Sonnet 4.5	48.3%	80.6%	45.8%	36.1%	30.6%
Claude Haiku 4.5	45.8%	83.3%	44.4%	31.9%	23.6%
GPT-4.1	29.2%	62.5%	22.2%	18.1%	13.9%
GPT-4.1-mini	29.9%	63.9%	19.4%	20.8%	15.3%

Table 6: Individual model accuracy across difficulty levels, revealing comparable performance between compact and standard variants within each family.

C Error Decomposition by Category

Table 7 shows the distribution of rounding issues versus true calculation errors across all eight categories.

Category	Rounding	True Error
Arithmetic	100%	0%
Weighted Averages	98.5%	1.5%
Geometry Conversion	99.3%	0.7%
Compound Calculations	95.4%	4.6%
Inverse Calculations	93.9%	6.1%
Financial Math	91.7%	8.3%
Division Precision	61.0%	39.0%
Fractions	0%	100%

Table 7: Category-level error decomposition showing fractions exhibit 100% genuine calculation failures while arithmetic shows 100% rounding issues.

D First-Step Errors by Model

Table 8 presents model-specific first-step error rates at easy and expert difficulty levels.

Model	Easy	Expert
Claude Sonnet 4.5	2.8%	6.9%
Claude Haiku 4.5	0.0%	11.1%
GPT-4.1	16.7%	25.0%
GPT-4.1-mini	20.8%	31.9%

Table 8: Model-specific first-step error rates showing GPT models exhibit substantially higher baseline failure rates and steeper increases with difficulty than Claude models.