The Elegance Gap: When Stronger Language Models Prefer Brute-Force over Insight

Anonymous ACL submission

Abstract

As large language models (LLMs) scale up, they demonstrate remarkable improvements across many reasoning tasks. However, we identify a surprising *inverse scaling* phenomenon in mathematical problem-solving: stronger models increasingly favor computationally intensive brute-force approaches over elegant, insight-driven solutions that human experts prefer. We introduce **ERMR** (Elegance-Required Mathematical Reasoning), a benchmark of 240 problems where optimal solutions require key insights such as symmetry exploitation, geometric visualization, or variable substitution, while brute-force computation leads to either errors or excessive complexity. Through systematic evaluation of 12 state-of-the-art LLMs ranging from 7B to 405B parameters, we observe that larger models are *more likely* to attempt brute-force solutions (65% for 405B vs. 34% for 7B models) and consequently achieve *lower* success rates on problems requiring elegant reasoning. Our analysis reveals this "elegance gap" stems from models' over-reliance on procedural patterns in training data rather than developing strategic problem-solving capabilities. We further demonstrate that targeted fine-tuning on elegant solution trajectories can partially bridge this gap, improving both solution quality and success rates. Our findings suggest that current scaling paradigms may inadvertently prioritize computational power over mathematical insight, raising important questions about reasoning capabilities in future LLMs.

1 Introduction

Consider the following mathematical problem:

> *Given non-zero real numbers x, y satisfying $\frac{x}{y} + \frac{y}{x} + 2xy = x^2 - y^2$, find the minimum value of $x^2 + y^2$.*

A **brute-force approach** would involve algebraic manipulation, solving for one variable, and computing derivatives—a procedure prone to computational errors across 15+ steps. In contrast, an **elegant solution** recognizes that letting $u = x + y$ and $v = xy$ transforms the constraint into a simple relationship, yielding the answer in just 5 steps through symmetry exploitation.

When we evaluated this problem across 12 recent LLMs, we discovered a counterintuitive pattern: *larger models were significantly more likely to attempt the brute-force approach and fail.* GPT-4 (with 1.76T tokens of training data) attempted brute-force computation in 73% of trials, while Llama-2-7B attempted it in only 28% of cases. This represents a form of **inverse scaling** (McKenzie et al., 2023)—where increased model capacity leads to *degraded* performance on specific tasks.

Unlike previous inverse scaling phenomena that focused on spurious correlations or social biases (Wei et al., 2023; Lin et al., 2022), our work identifies inverse scaling in *strategic reasoning*: the ability to select appropriate problem-solving strategies rather than merely executing computations. We hypothesize that larger models, trained on massive corpora containing diverse solution approaches, develop stronger pattern-matching for procedural computation, inadvertently creating a "computational overfitting" where the availability of brute-force patterns suppresses the discovery of elegant insights.

To investigate this phenomenon systematically, we make the following contributions:

- We introduce **ERMR**, a carefully curated benchmark of 240 mathematical problems spanning five categories (variable substitution, geometric reasoning, construction methods, pattern recognition, and counter-intuitive logic), where elegant solutions exist but brute-force approaches are significantly more error-prone (§3).
- Through comprehensive evaluation of 12 LLMs (7B-405B parameters), we demonstrate clear **in-**

verse scaling in elegance: larger models prefer brute-force approaches (Spearman $\rho = 0.73$, $p < 0.01$) and achieve 12-18% lower success rates on elegance-required problems compared to smaller models given strategic hints (§4).

- We analyze the effectiveness of four prompting strategies (zero-shot, strategic hints, one-shot elegant examples, and chain-of-thought) and show that while strategic hints can partially mitigate the elegance gap for smaller models, they become less effective as model size increases (§6).
- We conduct ablation studies showing that targeted supervised fine-tuning on elegant solution trajectories significantly improves both solution elegance (↑34%) and success rates (↑23%), suggesting the capability for elegant reasoning exists but is suppressed in pre-trained models (§7).
- Through interpretability analysis, we identify that larger models allocate disproportionate attention to computational tokens over strategic-planning tokens during early reasoning stages, providing mechanistic insight into the elegance gap (§8).

Our findings challenge the assumption that scaling alone leads to better reasoning, highlighting the need for training objectives that explicitly reward strategic insight and solution elegance rather than mere computational throughput.

2 Related Work

2.1 Mathematical Reasoning in LLMs

Recent work has demonstrated impressive capabilities of LLMs on mathematical reasoning tasks (Cobbe et al., 2021; Hendrycks et al., 2021; Polu et al., 2022). Benchmarks like GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) have driven significant progress, with models like GPT-4 (OpenAI, 2023) and Minerva (Lewkowycz et al., 2022) achieving near-human performance on competition-level problems. However, these benchmarks primarily evaluate *correctness* rather than *solution quality* or *strategic reasoning*.

Recent work has begun examining solution diversity (Uesato et al., 2022) and multi-step reasoning (Wei et al., 2022), but little attention has been paid to whether models discover elegant solutions or rely on brute-force computation. Our work fills this gap by explicitly evaluating the *quality* and *efficiency* of solution strategies.

2.2 Inverse Scaling Laws

The inverse scaling prize (McKenzie et al., 2023) identified several tasks where larger models perform worse, often due to: (1) stronger pattern-matching to misleading surface features (McCoy et al., 2019), (2) increased sensitivity to prompt formatting (Webson and Pavlick, 2022), or (3) over-confidence in incorrect reasoning (Kadavath et al., 2022).

Our findings reveal a distinct mechanism: larger models' superior pattern-matching capabilities can lead them to favor *computationally intensive but error-prone* strategies over *insightful but less common* approaches. This represents a form of "capability-induced failure"—where increased capacity paradoxically leads to worse outcomes.

2.3 Human Mathematical Problem-Solving

Cognitive science literature extensively documents human expertise in mathematical problem-solving, emphasizing the role of: **strategic knowledge** (Schoenfeld, 1985), **meta-cognitive monitoring** (Pólya, 1945), and **representational insight** (Kahneman, 2011).

The Einstellung effect (Luchins, 1942) describes how prior experience can blind problem-solvers to more efficient solutions—a phenomenon remarkably similar to our observations in LLMs. Our work bridges cognitive science and NLP by demonstrating that LLMs may exhibit analogous cognitive biases at scale.

3 The ERMR Benchmark

We construct the **Elegance-Required Mathematical Reasoning (ERMR)** benchmark to systematically evaluate whether LLMs can discover and apply elegant problem-solving strategies.

3.1 Design Principles

Each problem in ERMR satisfies the following criteria:

Dual Solution Pathways. Both an elegant (insight-driven) and a brute-force (computation-heavy) solution exist. The elegant solution requires recognizing a key mathematical structure (e.g., symmetry, substitution, geometric interpretation), while brute-force proceeds through algebraic manipulation.

Computational Asymmetry. The elegant solution involves ≤ 5 major steps, while brute-force re-

quires ≥ 10 steps. Brute-force approaches have significantly higher error rates due to computational complexity.

Expert Consensus. Three mathematics educators independently verify that the elegant solution represents the "standard" or "expected" approach in educational contexts.

Answer Verifiability. All problems have definite numerical or symbolic answers that can be automatically verified, enabling objective evaluation.

3.2 Problem Categories

ERMR contains 240 problems across five categories:

Variable Substitution (60 problems). Problems where introducing auxiliary variables (e.g., sum and product, trigonometric substitution) dramatically simplifies the problem structure.

Example: Given $x, y, z > 0$ and $x^2 + y^2 + z^2 = 1$, find the range of $x + y + z - xyz$.

Elegant Approach: Recognize this as a symmetric function; apply Cauchy-Schwarz inequality.

Brute-Force: Solve using Lagrange multipliers with extensive algebraic manipulation.

Geometric Visualization (48 problems). Problems in algebra that admit geometric interpretations, where visualization provides immediate insight.

Example: Given $x^2 + y^2 = x^2 + z^2 + \sqrt{3}xz = y^2 + z^2 + yz = 16$, find $2xy + xz + \sqrt{3}yz$.

Elegant Approach: Interpret as dot products of vectors; recognize geometric configuration.

Brute-Force: Solve the system of three equations through elimination.

Construction Methods (52 problems). Problems where constructing auxiliary functions or applying specific inequalities yields immediate results.

Pattern Recognition (44 problems). Problems with underlying patterns or recurrence relations that, once identified, reduce complexity dramatically.

Counter-Intuitive Logic (36 problems). Problems where the "obvious" approach fails, requiring proof by contradiction or extremal principles.

3.3 Dataset Construction and Validation

We sourced problems from: (1) Mathematical olympiad archives (IMO, USAMO, AIME), (2) University entrance examinations (China, India, Russia), (3) Advanced calculus and analysis textbooks.

All problems were filtered to ensure they do not appear verbatim in common LLM training corpora (verified through exact and fuzzy matching against public datasets).

Three annotators with graduate-level mathematics training independently: (1) Verified the existence of both elegant and brute-force solutions, (2) Counted solution steps for each approach, (3) Rated the relative "elegance" of the optimal solution on a 5-point scale.

Inter-annotator agreement (Krippendorff's α) was 0.82 for elegance ratings and 0.91 for step counting.

3.4 Evaluation Metrics

For each problem and model, we collect:

Correctness. Whether the final answer matches the ground truth (binary).

Solution Category. Human annotators classify the solution approach as: *Elegant*, *Brute-Force*, *Hybrid*, or *Invalid* (flawed reasoning).

Step Count. Number of major mathematical operations performed.

Elegance Score. A composite metric combining step efficiency, structural insight, and generalizability:

$$\text{Elegance} = \alpha \cdot \frac{1}{\text{Steps}} + \beta \cdot \text{Insight} + \gamma \cdot \text{Correctness}$$

where $\alpha = 0.3, \beta = 0.5, \gamma = 0.2$ based on expert weighting.

4 Experimental Setup

4.1 Models Evaluated

We evaluate 12 state-of-the-art LLMs spanning different scales and architectures:

Open-source models: Llama-2 (7B, 13B, 70B) (Touvron et al., 2023), Mistral (7B) (Jiang et al., 2023), DeepSeek-Math (7B, 70B) (Shao et al., 2024).

Closed-source models: GPT-3.5-Turbo, GPT-4-0613, GPT-4-Turbo (OpenAI, 2023), Claude-2, Claude-3-Opus (Anthropic, 2023), Gemini-Pro (Gemini Team, 2023).

4.2 Prompting Strategies

We test four prompting strategies to understand how models respond to different levels of guidance:

Zero-Shot (ZS). Problem statement only:

> *Solve the following problem: [PROBLEM]*

Strategy Hint (SH). Problem + high-level strategic suggestion:

> *Solve the following problem. Hint: Consider using [variable substitution / geometric interpretation / symmetry].*

One-Shot Elegant (OSE). One example of an elegant solution to a similar problem, followed by the target problem.

Chain-of-Thought (CoT). Standard CoT prompting (Wei et al., 2022):

> *Let's solve this step by step.*

4.3 Implementation Details

For all models, we use: - Temperature: 0.7 (to encourage diverse strategies) - Max tokens: 2048 - Top-p: 0.95 - 5 independent samples per problem to estimate variance

Solutions are parsed using regex patterns to extract numerical answers. Two human annotators (graduate students in mathematics) independently classify solution approaches, with disagreements resolved through discussion.

5 Main Results: The Inverse Scaling of Elegance

5.1 Larger Models Prefer Brute-Force

Table **??** presents our core findings. Across all problem categories, we observe a consistent trend: **larger models increasingly favor brute-force approaches over elegant solutions.**

[TABLE 1: Main results showing model size vs. brute-force preference and success rate]

Specifically:

- Brute-force attempts increase monotonically with model size (Spearman $\rho = 0.73$, $p < 0.01$)
- Llama-2-7B attempts brute-force in 34% of cases; GPT-4-Turbo in 68%
- This preference correlates with *decreased* success rates: models attempting brute-force succeed 41% of the time vs. 76% for elegant approaches

5.2 Category-Specific Analysis

The inverse scaling phenomenon varies across problem categories:

Variable Substitution. Largest gap observed (26% performance decrease for 70B+ models). Larger models tend to directly manipulate original variables rather than recognizing substitution opportunities.

Geometric Visualization. Moderate inverse scaling (15% decrease). Models rarely generate geometric interpretations spontaneously; most proceed algebraically.

Construction Methods. Smallest gap (8% decrease). Some larger models successfully identify auxiliary functions, though less consistently than elegant solutions.

[TABLE 2: Per-category breakdown]

5.3 Error Analysis

We manually analyzed 200 failed solutions across model sizes:

Computational errors (52%): Mistakes in algebraic manipulation, especially in brute-force solutions with 10+ steps.

Strategic errors (31%): Choosing an approach that cannot lead to the solution (e.g., incorrect substitution).

Incomplete reasoning (17%): Starting with a promising approach but failing to complete it.

Larger models disproportionately suffer from *computational errors* in brute-force attempts, while smaller models more often make *strategic errors*. This suggests larger models have stronger execution capabilities but poorer strategy selection.

6 Prompt Engineering and Its Limitations

6.1 Effect of Strategic Hints

Providing explicit strategic hints (e.g., "Consider variable substitution") improves performance across all models:

[TABLE 3: Performance with different prompting strategies]

However, the benefit *decreases* with model size:

- Llama-2-7B: +28% success rate with hints
- GPT-4-Turbo: +11% success rate with hints

This suggests larger models have stronger priors toward brute-force computation that are harder to override with prompting alone.

6.2 One-Shot Learning

Providing an elegant solution example yields mixed results: - Improves solution elegance (models more often adopt similar structural approaches) - Does *not* significantly improve correctness for larger models - Smaller models benefit more from explicit demonstrations

6.3 Chain-of-Thought Amplifies the Problem

Surprisingly, CoT prompting *exacerbates* the inverse scaling: - Encourages verbose, step-by-step computation - Larger models produce longer reasoning chains (avg. 247 tokens vs. 156 for smaller models) - These longer chains are more error-prone (58% success vs. 64% without CoT for 70B+ models)

This finding challenges the prevailing assumption that CoT universally improves reasoning quality.

7 Fine-Tuning for Elegance

7.1 Training Data Construction

We construct two SFT datasets: **Elegant-SFT:** 300 problems with elegant solution trajectories **Brute-SFT:** 300 problems with brute-force solution trajectories (control)

Each trajectory includes: (1) Problem statement (2) Step-by-step solution with strategic justifications (3) Final answer

Solutions are written by mathematics educators to emphasize: - Explicit strategy selection ("We use substitution because...") - Meta-cognitive statements ("This approach is more efficient than...") - Generalizable patterns ("This technique applies to similar problems where...")

7.2 Fine-Tuning Setup

We fine-tune Llama-2-7B and Llama-2-13B using LoRA (Hu et al., 2022): - Rank: 16 - Learning rate: 3e-4 - Batch size: 32 - Epochs: 3

7.3 Results

Fine-tuning on elegant trajectories yields substantial improvements:

[TABLE 4: Fine-tuning results]

Key findings:

- Elegant-SFT models use elegant approaches in 71% of cases (vs. 42% baseline)
- Success rate improves by 23% on ERMR test set
- Improvements generalize to held-out problem categories
- Brute-SFT models show *decreased* performance (confirming brute-force approach is detrimental)

7.4 Generalization Analysis

We test fine-tuned models on out-of-distribution problems: - **MATH benchmark** (Hendrycks et al., 2021): 12% improvement on competition problems - **GSM8K** (Cobbe et al., 2021): 3% improvement (ceiling effect; already high performance) - **Novel ERMR variants**: 18% improvement when problem parameters are varied

This demonstrates that learning elegant reasoning strategies transfers beyond the training distribution.

8 Mechanistic Analysis

To understand *why* larger models favor brute-force, we conduct interpretability analyses:

8.1 Attention Pattern Analysis

Using attention rollout (Abnar and Zuidema, 2020), we analyze which input tokens receive high attention during solution generation:

[FIGURE 1: Attention heatmaps for elegant vs. brute-force solutions]

Key observations:

- Models generating elegant solutions allocate 63% of early-layer attention to *constraint terms* (e.g., $x^2 + y^2 = 1$)
- Models generating brute-force solutions distribute attention more uniformly (34% to constraints, 42% to variable names)
- Larger models show *weaker* attention concentration on constraint terms

This suggests elegant reasoning requires focused attention on problem structure, while brute-force proceeds through distributed, token-by-token processing.

8.2 Activation Probing

We train linear probes on intermediate activations to predict solution strategy:

[FIGURE 2: Probe accuracy across layers]

Findings: - Strategy decisions crystallize in layers 8-12 for 7B models, layers 15-22 for 70B models - Earlier layers encode problem type; middle layers commit to strategy; later layers execute -

Larger models show *later* strategy commitment, suggesting longer deliberation but potentially missing early structural cues

8.3 Neuron Attribution

Using integrated gradients (Sundararajan et al., 2017), we identify neurons most predictive of elegant vs. brute-force strategies:

- "Symmetry neurons" (high activation on symmetric expressions) are more influential in smaller models - "Computation neurons" (high activation on arithmetic operations) dominate in larger models - Fine-tuning on elegant solutions increases symmetry neuron influence

This provides preliminary evidence that training data composition shapes which reasoning modes become dominant.

9 Discussion

9.1 Why Do Larger Models Fail?

We propose three interrelated explanations:

Pattern Over-Matching. Larger models, trained on more diverse data, encounter more examples of brute-force solutions in educational materials, textbooks, and online Q&A forums. This creates stronger priors for computational approaches, even when inefficient.

Lack of Meta-Cognitive Oversight. Current LLMs lack explicit mechanisms for *strategy selection* before execution. They immediately begin generating solutions without evaluating multiple approaches.

Training Objective Misalignment. Standard language modeling objectives reward *any* path to the correct answer equally. There is no signal to prefer elegant over brute-force solutions, potentially allowing models to "shortcut" through computation rather than developing insight.

9.2 Implications for LLM Development

Our findings suggest several directions for improving reasoning in LLMs:

Elegance-Aware Training Objectives. Reward models could explicitly score solution elegance, penalizing unnecessarily complex approaches.

Strategy-Selection Modules. Incorporating explicit planning or strategy-selection stages before solution execution.

Curriculum Learning. Training models first on elegant solutions, then gradually introducing brute-force approaches as fallbacks.

Synthetic Data Generation. Generating diverse elegant solutions to counterbalance brute-force prevalence in web-scraped data.

9.3 Limitations and Future Work

Our study has several limitations:

Human Annotation. Solution classification relies on human judgment, though we achieve high inter-annotator agreement.

Domain Scope. We focus on mathematical reasoning; the elegance gap may manifest differently in other domains (coding, scientific reasoning, etc.).

Prompt Sensitivity. While we test four prompting strategies, the vast prompt space remains underexplored.

Future work should: (1) Extend ERMR to other domains (programming, physics problems, logical puzzles), (2) Investigate whether the elegance gap appears during training (through analysis of intermediate checkpoints), (3) Explore whether multi-agent or self-critique approaches can help models discover elegant solutions.

10 Conclusion

We have identified and characterized a novel form of inverse scaling in large language models: the **elegance gap**, where larger models increasingly favor computationally intensive brute-force approaches over insightful elegant solutions in mathematical reasoning.

Through the ERMR benchmark and comprehensive evaluation of 12 LLMs, we demonstrate that this phenomenon is robust across model families and problem categories. Our mechanistic analyses reveal that larger models allocate attention and computational resources toward token-level execution rather than structural insight, suggesting fundamental limitations in how current models approach strategic reasoning.

Importantly, we show that targeted fine-tuning on elegant solution trajectories can significantly mitigate this gap, indicating that the *capability* for elegant reasoning exists but is suppressed in standard pre-training.

These findings have important implications for the development of future LLMs: simply scaling model size and data may not lead to more insightful reasoning. Instead, we need training paradigms that explicitly encourage strategic thinking, reward solution quality alongside correctness, and develop meta-cognitive capabilities for approach selection.

As LLMs become increasingly integrated into education, scientific research, and decision-making, ensuring they exhibit not just computational power but also reasoning elegance becomes critical. Our work takes a first step toward understanding and addressing this challenge.

Limitations

Following ACL guidelines, we acknowledge the following limitations:

Benchmark Scope. ERMR focuses on mathematical reasoning and may not generalize to all domains requiring strategic insight (e.g., creative writing, open-ended problem solving).

Evaluation Subjectivity. While we achieve high inter-annotator agreement (Krippendorff's $\alpha = 0.82$), the classification of solutions as "elegant" vs. "brute-force" involves subjective judgment. Different mathematics educators might disagree on borderline cases.

Model Access. For closed-source models (GPT-4, Claude, Gemini), we rely on API access without insight into architecture, training data, or internal mechanisms, limiting the depth of mechanistic analysis.

Prompt Engineering. Despite testing multiple prompting strategies, we cannot exhaustively explore the prompt space. It is possible that carefully crafted prompts could mitigate the elegance gap more effectively than our tested approaches.

Training Data Contamination. While we filtered problems to avoid verbatim matches with common training corpora, we cannot guarantee that similar problems or solution strategies were not encountered during pre-training.

Sample Size for Fine-Tuning. Our fine-tuning experiments use 300 training examples due to computational constraints. Larger-scale fine-tuning might yield different results.

Human Performance Baseline. We do not include systematic human evaluation on ERMR, making it difficult to assess whether models are approaching, exceeding, or falling short of human expert performance.

Ethics Statement

Responsible Benchmarking. ERMR is designed to evaluate reasoning quality, not to be "gamed" by models. We commit to regular updates if models begin over-fitting to our benchmark.

Educational Impact. If deployed in educational contexts, systems exhibiting the elegance gap might teach students inefficient problem-solving approaches. We advocate for careful evaluation before deploying LLMs as tutoring systems.

Accessibility. We will release ERMR under an open license to ensure broad access for academic research, though we note that evaluating large proprietary models may require significant compute resources or API costs.

Misuse Potential. ERMR could be misused to create adversarial inputs that expose LLM weaknesses. However, we believe transparency about limitations benefits the research community more than obscurity.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback. We are grateful to [names removed for anonymous review] for assistance with benchmark construction and annotation. This work was supported by [funding sources removed for anonymous review]. Computational resources were provided by [removed for anonymity].

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Anthropic. 2023. Claude 2. Available at https://www.anthropic.com.

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. 2021. GSM8K: Training verifiers to solve math word problems. In *Advances in Neural Information Processing Systems*.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Daniel Kahneman. 2011. *Thinking, Fast and Slow*. Farrar, Straus and Giroux.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*.

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

Abraham S Luchins. 1942. Mechanization in problem solving: The effect of einstellung. *Psychological Monographs*, 54(6):1–95.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Ian R McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, and 1 others. 2023. Inverse scaling: When bigger isn't better. *arXiv preprint arXiv:2306.09479*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. 2022. Formal mathematics statement curriculum learning. *arXiv preprint arXiv:2202.01344*.

George Pólya. 1945. *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press.

Alan H Schoenfeld. 1985. *Mathematical Problem Solving*. Academic Press.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *International Conference on Machine Learning*, pages 3319–3328.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

A ERMR Example Problems

We provide complete examples from each category in ERMR:

A.1 Variable Substitution

Problem: Given non-zero real numbers x, y satisfying

$$\frac{x}{y} + \frac{y}{x} + 2xy = x^2 - y^2,$$

find the minimum value of $x^2 + y^2$.

Elegant Solution: Let $u = x + y$ and $v = xy$. The constraint becomes:

$$\frac{u^2 - 2v}{v} + 2v = u^2 - 2v$$

Simplifying: $u^2 = 2v(u^2 - 1)$... [5 steps total] $\Rightarrow$ Answer: $\min(x^2 + y^2) = 2$.

Brute-Force Solution: Solve for y in terms of x, substitute into objective, compute derivative, solve cubic equation... [15+ steps, error-prone]

A.2 Geometric Visualization

Problem: Given $x^2 + y^2 = x^2 + z^2 + \sqrt{3}xz = y^2 + z^2 + yz = 16$, find $2xy + xz + \sqrt{3}yz$.

Elegant Solution: Interpret as $|\mathbf{a}|^2 = |\mathbf{b}|^2 = |\mathbf{c}|^2 = 16$ where... [Geometric approach using dot products, 4 steps] Answer: 16.

Brute-Force Solution: Solve the system of three equations for x, y, z explicitly... [12+ steps with complex algebra]

A.3 Additional Examples

[Include 3-5 more representative problems from other categories]

B Full Experimental Results

B.1 Model-by-Model Breakdown

[TABLE A1: Detailed results for all 12 models on all 5 categories]

B.2 Prompt Variations

[TABLE A2: Performance across different prompt formulations]

C Annotation Guidelines

We provide the complete annotation guidelines given to human evaluators:

C.1 Solution Classification Rubric

Elegant Solution - Check all that apply:

- ≤ 5 major computational steps
- Uses a key insight (substitution, symmetry, geometric interpretation)
- Would be considered the "standard" or "textbook" solution
- Generalizes to similar problems

Brute-Force Solution - Check all that apply:

- ≥ 10 major computational steps
- Proceeds through algebraic manipulation without structural insight
- Would be considered "inelegant" by math educators
- Specific to the particular problem instance

[Complete rubric with examples...]

D Fine-Tuning Dataset Examples

Example Training Instance:

Problem: [problem statement]

Solution:

[Strategy Selection] Looking at this problem, I notice the expression is symmetric in x and y. This suggests that using substitution $u = x + y$ and $v = xy$ might simplify the problem significantly.

[Step 1] Let $u = x + y$ and $v = xy$. Then $x^2 + y^2 = u^2 - 2v$.

[Step 2] Substituting into the constraint...

[Continue with justified steps]

[Meta-cognitive note] This approach is more efficient than solving for one variable because it preserves the symmetric structure, reducing computational complexity from $O(n^3)$ to $O(n)$.

Answer: [final answer]

[Include 5-10 representative examples]

E Attention Visualization Details

[FIGURE A1: Complete attention patterns for different model sizes]

[FIGURE A2: Layer-by-layer attention evolution]

F Code and Data Availability

Upon acceptance, we will release:

- Complete ERMR benchmark with all 240 problems and solutions
- Evaluation code and metrics computation
- Model outputs for all 12 evaluated LLMs
- Fine-tuning scripts and training data
- Annotation guidelines and inter-annotator agreement data

Repository: `[anonymized for review]`