The Relationship Between Reasoning and Performance in Large Language Models—o3 (mini) Thinks Harder, Not Longer

Anonymous ACL submission

Abstract

Large language models have demonstrated remarkable progress in mathematical reasoning, leveraging chain-of-thought and reinforcement learning. However, it is unclear whether improved performance results from longer reasoning chains or more efficient reasoning. We systematically analyze reasoning chain length across o1-mini and o3-mini variants on the Omni-MATH benchmark, finding that o3-mini (m) achieves superior accuracy without requiring longer reasoning chains than o1-mini. Moreover, we show that accuracy generally declines as reasoning chains grow across all models and compute settings. This accuracy drop is significantly smaller in more proficient models, suggesting that new generations of reasoning models use test-time compute more effectively. Finally, we highlight that while o3-mini (h) achieves a marginal accuracy gain over o3-mini (m), it does so by allocating substantially more reasoning tokens across all problems, even the ones that o3-mini (m) can already solve. These findings provide new insights into the relationship between model capability and reasoning length, with implications for efficiency, scaling, and evaluation methodologies.[1]

1 Introduction

Large language models (LLMs) have evolved from handling basic natural language processing tasks to solving complex problems (Brown et al., 2020; Bubeck et al., 2023; Romera-Paredes et al., 2024; Trinh et al., 2024). Scaling model size, data, and compute (Kaplan et al., 2020) has enabled larger models to develop richer internal representations (Gurnee and Tegmark, 2024; Hao et al., 2023) and emergent capabilities (Wei et al., 2022a). Recently, a new class of reasoning models has emerged that couples reinforcement learning with test-time compute scaling (Muennighoff et al., 2025; Snell et al., 2025). These models leverage reasoning tokens to guide the chain-of-thought process and maintain coherence throughout complex problem-solving tasks (Anderson et al., 2025; Chen et al., 2024; Wang et al., 2025). By explicitly optimizing their reasoning process during training (Wei et al., 2022b) and iteratively refining their output at inference-time, they achieve superior performance, even on challenging mathematical benchmarks (DeepSeek-AI et al., 2025; Guan et al., 2025).

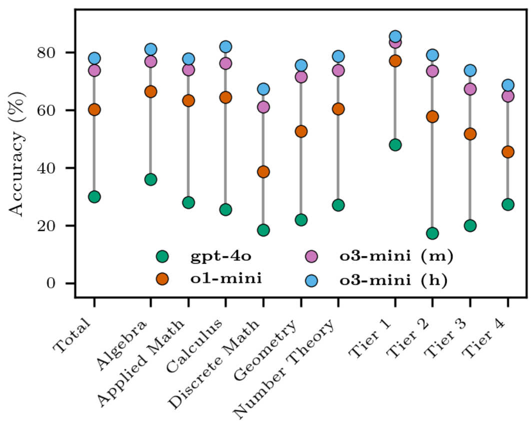

Figure 1: **Performance of OpenAI models on Omni-MATH** across disciplines and difficulty tiers.

In this paper, we examine whether more capable models within a single family (OpenAI's o-series) require a longer reasoning chain to achieve higher performance or if they can reason more effectively. We systematically compare the number of tokens in the reasoning chain generated by o1-mini, o3-mini (m), and o3-mini (h) on the Omni-MATH benchmark (Gao et al., 2024). Omni-MATH spans more than 33 mathematical subdomains and 10 difficulty levels, providing a comprehensive evaluation framework for LLMs' mathematical reasoning abilities. Well-known benchmarks GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b) have become less effective in differentiating the mathematical abilities of LLMs due to the high accuracy rates they

[1] Data and code are enclosed in an anonymized zip file and will be released publicly upon acceptance.

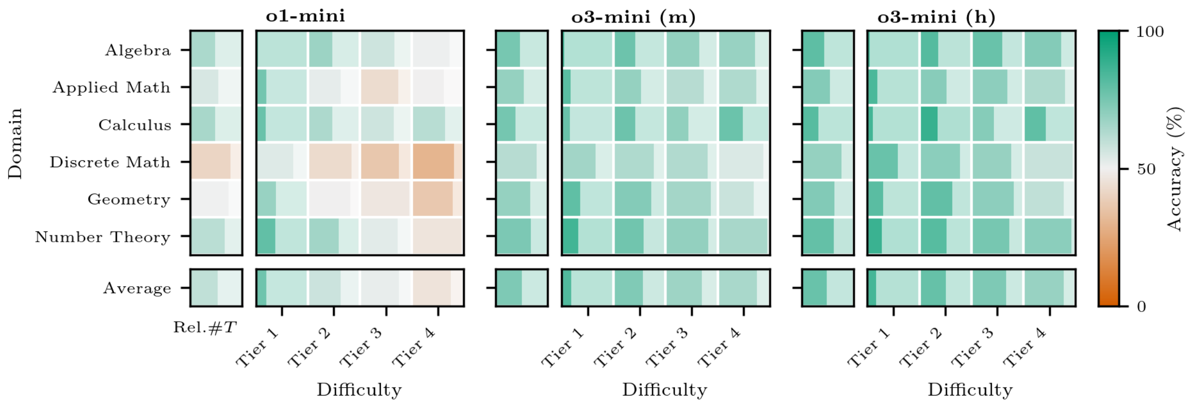

Figure 2: **Granular performance and relative reasoning token use.** This figure shows that models allocate more computational resources to problems that require complex combinatorial reasoning, whereas foundational arithmetic and algebra problems demand relatively fewer resources. On average, token usage scales with difficulty level. The heatmaps visualize cross-sectional performance scores on a 0-100% scale, represented by the color of the progress bar. The length of the progress bar in each cell represents relative token usage for the test-time scaled models. The extra column is computed by averaging over the rows. The extra row and "average" cell are computed independently to give equal weight to multi-domain questions (see Appendix A.1).

achieve on these tests. More challenging benchmarks, such as FrontierMath (Glazer et al., 2024), GSM-Symbolic (Mirzadeh et al., 2024), and sections of Humanity's Last Exam (Phan et al., 2025) only span a few disciplines or are largely being kept private due to data leakage concerns.

We find that more proficient models (o1-mini vs. o3-mini (m)) do not generate longer reasoning chains to achieve higher accuracy. For all models and compute settings, we find that accuracy generally decreases as the reasoning chain grows, even when controlling for question difficulty. This effect is notably smaller for more proficient models, indicating that o3-mini (m) tends to overthink less and uses reasoning tokens more effectively than o1-mini. However, within one model (o3-mini (m) vs. o3-mini (h)), we observe that the slower accuracy decrease per token is partially due to a higher average accuracy, but mainly due to the model allocating (more than) double the reasoning tokens for all questions. Our findings contribute to the ongoing discussion about whether models such as o1 tend to overthink or underthink (Chen et al., 2024; Wang et al., 2025), while complementing studies on reasoning step length (Jin et al., 2024), input length (Levy et al., 2024), reasoning failure modes (Anderson et al., 2025), and the optimization of mathematical reasoning (Zhong et al., 2024).

2 Experiment

Our data consists of $4,428$ Olympiad-level math problems, the Omni-MATH benchmark, together with a reference answer and relevant metadata fields Domain and Difficulty (Figures A1 and A2). We consider six elementary mathematics domains, Algebra, Applied Mathematics, Calculus, Discrete Mathematics, Geometry and Number Theory and divide the data into four difficulty tiers, Tier 1, Tier 2, Tier 3 and Tier 4 (Figures A3 and A4). Subsequently, we feed the problems to four OpenAI models, namely gpt-4o, o1-mini, o3-mini (m) and o3-mini (h) and make automated requests to the Omni-Judge model (Gao et al., 2024)–a math-evaluation model designed to verify and correct model-generated answers against reference answers–to correct their answers. Consult Appendix A for implementation details.

3 Results

Reasoning models consistently outperform gpt-4o. Figures 1 and 2 show the performance of OpenAI models gpt-4o, o1-mini, o3-mini (m) and o3-mini (h) across mathematical disciplines and difficulty tiers. The gpt-4o model performs between 20% and 30% for all disciplines and clearly lags behind the three reasoning models. o1-mini significantly improves accuracy, reaching 40–60% on all categories. The introduction of o3-mini (m) further enhances performance, achieving 50% in all categories. The o3-mini (h) model improves with approximately 4% on average compared to o3-mini (m) and surpasses 80% accuracy for Algebra and Calculus. A notable outlier is Discrete Mathematics, where performance deviates from the overall trend for all models. In general, accuracy declines as tier level increases. An exception is observed

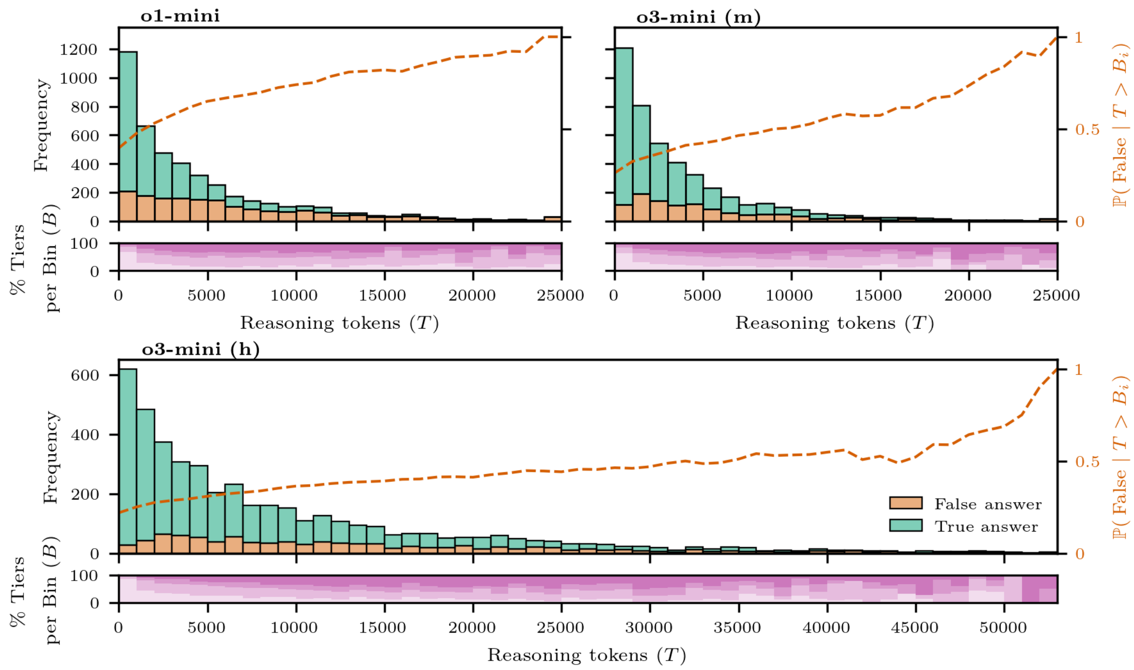

Figure 3: **Accuracy across the reasoning token distribution and difficulty tier consistency.** This figure shows that o1-mini and o3-mini (m) have a similar reasoning token distribution, with o3-mini (m) giving more correct answers for high-token regions. o3-mini (h) has a good ratio of correct vs. incorrect answers, even for very high token counts. Finally, bin composition shifts from mostly low-tier to mostly high-tier questions (high-token regions sometimes have insufficient data points to show this pattern). The main panels of the figure display the proportion of the correct (green bars) versus incorrect (red bars) model responses across the reasoning token distribution. The red dashed line depicts the conditional error rate, i.e. the probability that the model answers incorrectly given that the token count has surpassed the bin threshold (see Appendix A.4). The panels below the histogram contain a filled histogram where the color opacity represents the difficulty level of the math questions (cfr. Figure A4).

in gpt-4o, which performs better on Tier 4 than on Tiers 2 and 3. This anomaly suggests that gpt-4o might leverage unexpected heuristics or struggle disproportionately with mid-tier complexity.

More complex questions demand greater reasoning depth. Besides indicating accuracy (via the colors of the progress bars), Figure 2 also shows relative use of reasoning tokens (via the length of the progress bars). The relative use of tokens increases with the level of difficulty for all models, highlighting the need for computational resources for more difficult tasks. This is confirmed by the lower panels in Figure 3. Discrete Mathematics stands out as a token-intensive domain, indicating a heavier combinatorial or multi-step reasoning load. Foundational mathematics areas such as Calculus and Algebra tend to consume fewer tokens, possibly because they are more procedurally straightforward. Interestingly, we observe that a relatively longer chain of reasoning does not generally lead to better performance, as many Tier 4 math problems from token-intensive domains remain unsolved.

More proficient models give more correct answers for high-token regions. Figures 3 and 4 display the relationship between the number of reasoning tokens and the performance of o1-mini, o3-mini (m), and o3-mini (h) on the Omni-MATH dataset (consult Figure A5 for gpt-4o analysis with completion tokens, which encompass both the tokens leading up to the answer and the answer itself). One first thing to note is that higher performing models have a better ratio of correct to incorrect answers, even for high token counts. This pattern is also reflected in the conditional error rate: the conditional error rate is almost instantly at 50% for o1-mini whereas it takes about $12,000$ tokens for o3-mini (m) and $30,000$ for o3-mini (h) to reach a 50% error rate.

o3 (mini) Thinks Harder, Not Longer. A second thing to note is that the token distributions of o1-mini and o3-mini (m) are very similar. Figure 4b together with the left QQ-plot in Figure A6 further investigate this behavior by comparing the distribution of the reasoning tokens only for the questions that the models answered *correctly*. Indeed, the almost identical token distributions show that o3-mini (m) does not use more reasoning tokens to achieve its superior performance to o1-mini

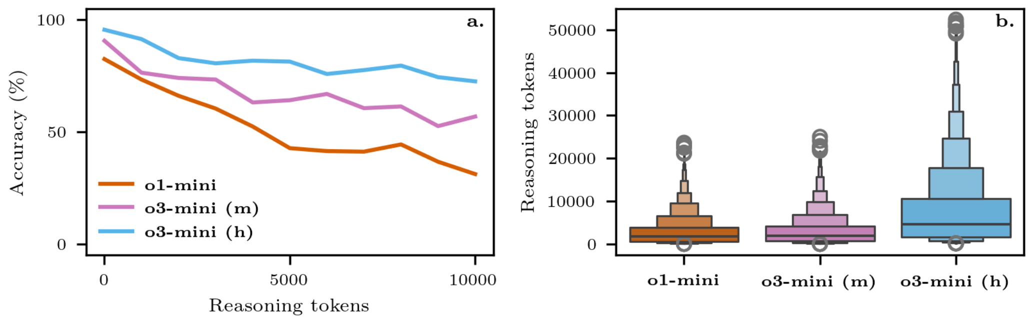

Figure 4: **Accuracy vs. token use.** This figure shows that o3-mini (m) does not require longer reasoning chains than o1-mini to achieve better accuracy and that, in general, more proficient models exhibit less accuracy decay as reasoning tokens increase. Figure A6 confirms that o1-mini and o3-mini (m) have a very similar token distribution and that the token distribution of o3-mini (h) is stretched linearly with respect to the one of o3-mini (m). **a,** Accuracy per reasoning token, dividing the number of correctly answered questions by the total number of questions in each bin of the histograms in Figure 3. **b,** Distribution of the reasoning tokens for *correctly* answered questions.

on Omni-MATH. This suggests that o3-mini (m) reasons more effectively.

Accuracy decreases with token use. Figure 4a shows that the average accuracy decreases with increasing use of reasoning tokens for all three models, but that this trend is the most pronounced for o1-mini and smaller for o3-mini (m) and o3-mini (h). While this could be attributed to higher-tier questions requiring more tokens, Figure A7 shows that the trend remains even when stratifying by tier level. In Figure A8, we show this also holds when stratifying across domains. This suggests that increased token usage, rather than question complexity alone, is related to accuracy. We use a logistic regression to quantify the effect size of using additional reasoning tokens on the probability of answering a question correctly, controlling for different levels of difficulty and domains (see Appendix A.5 and Table A1). The average marginal effects indicate that the accuracy decrease per 1000 reasoning tokens is 3.16% for o1-mini, 1.96% for o3-mini (m), and 0.81% for o3-mini (h). These results indicate that while deeper reasoning is necessary for solving complex problems, there is a diminishing return, where excessive token usage correlates with reduced accuracy.

o3-mini (h) allocates more tokens for all questions. Figure 3 (bottom) shows that the token distribution of o3-mini (h) spans a significantly wider range of values, with the model allocating over $50,000$ reasoning tokens for some math problems. In addition, Figure 4b shows that o3-mini (h) uses more reasoning tokens to solve all correctly answered questions, indicating that the small accuracy gain of 4% compared to o3-mini (m) is accompanied by a large extra computational cost (confirmed by the right QQ-plot in Figure A6). The slower accuracy decrease per token of o3-mini (h) compared to o3-mini (m) is thus attributed to a stretched out token distribution along the x-axis.

4 Conclusion

By systematically comparing the number of tokens in the reasoning chain generated by o1-mini, o3-mini (m), and o3-mini (h) on the Omni-MATH dataset (Gao et al., 2024), we find two important results. First, more proficient models (o1-mini vs. o3-mini (m)) do not require longer reasoning to achieve higher accuracy. Second, while accuracy generally declines with a longer chain-of-thought, this effect is notably smaller in more proficient models, underscoring that "thinking harder" is not the same as "thinking longer". A possible hypothesis for this accuracy drop is that models tend to reason more on problems they cannot solve. Another possibility is that longer reasoning chains inherently have a higher probability of leading to a wrong final solution, highlighting the need for mathematical benchmarks with reference reasoning templates. A practical takeaway from our study is that constraining the chain-of-thought (by setting `max_completion_tokens`) is more useful for weaker reasoning models than for stronger ones, as the latter still give a significant amount of correct answers for high-token regions.

The token count for o3-mini (h) contained the following subtlety: although o3-mini (h) solves additional problems compared to o3-mini (m), the model uses more tokens for all math problems. The slower decrease in accuracy per token is thus due to a stretched token distribution rather than a more effective usage of reasoning tokens.

Limitations

This work evaluates the reasoning abilities of four large language models within the OpenAI family. Our methodological choice to focus on one family, stems from the fact that this allows us to directly compare post-training effects and test-time compute settings without introducing confounding factors such as closed- vs. open-source models, different training strategies or architectures. Furthermore, at the time of submission, the OpenAI o-series is the largest family that allows for this type of comparison.

The reasoning tokens usage of o1-mini, o3-mini (m) and o3-mini (h) is analysed exclusively when solving the Omni-MATH benchmark (Gao et al., 2024). As discussed in the Introduction, Omni-MATH is currently the only publicly available, annotated, mathematical dataset that is not deprecated. More general benchmarks like those in Srivastava et al. (2023) and MMLU (Hendrycks et al., 2021a), along with specialized tests such as AI2 Reasoning (Clark et al., 2018) and GPQA (Rein et al., 2024) would broaden the evaluation landscape to diverse reasoning domains. However, current focus remains on mathematics due to the relative ease of implementing objective reward models and automated verification procedures.

Finally, our study relies on automated correction by Omni-Judge (Gao et al., 2024), a model based evaluator that, while effective, may produce judgements that diverge from human corrections (Verga et al., 2024; Li et al., 2025). Omni-Judge has only been validated for data leakage checks on o1-mini (Gao et al., 2024), and extending these checks to o3-mini remains future work, though we assume minimal overlap. Additionally, our prompting strategy employed here (Kojima et al., 2022; Wang et al., 2023; Yao et al., 2024) may not generalize to alternative approaches or more constrained prompt settings (Tam et al., 2024). This interaction between prompt design and test-time compute is an important direction for further investigation, particularly as much of the existing research has focused on models without test-time compute. As a result, the broader implications for the latest generation of reasoning models (DeepSeek-AI et al., 2025) remain to be fully explored.

Ethical Considerations

We do not foresee any immediate ethical or societal implications arising from our work. However, as large language models become increasingly capable of solving complex (mathematical) problems, it is important to consider the implications of integrating such technologies into educational, scientific and professional contexts. Our work contributes to the responsible use of AI in these contexts by providing a rigorous analysis of the mathematical reasoning abilities of LLMs, with wider implications for efficient resource allocation, the refinement of LLMs' universal reasoning skills, and AI-driven scientific discovery.

References

Carolyn Jane Anderson, Joydeep Biswas, Aleksander Boruch-Gruszecki, Federico Cassano, Molly Q Feldman, Arjun Guha, Francesca Lucchetti, and Zixuan Wu. 2025. PhD Knowledge Not Required: A Reasoning Challenge for Large Language Models. *arXiv:2502.01584*. Version 2.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language Models are Few-Shot Learners. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, pages 1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, John A. Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuan-Fang Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *arXiv:2303.12712*. Version 5.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do NOT Think That Much for 2+ 3=? On the Overthinking of o1-like LLMs. *arXiv:2412.21187*. Version 2.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv:1803.05457*. Version 1.

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *arXiv:2110.14168*. Version 2.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv:2501.12948*. Version 1.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2024. Omni-MATH: A Universal Olympiad Level Mathematic Benchmark For Large Language Models. *arXiv:2410.07985*. Version 3.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, and 5 others. 2024. FrontierMath: A Benchmark for Evaluating Advanced Mathematical Reasoning in AI. *arXiv:2411.04872*. Version 5.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rStar-Math: Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking. *arXiv:2501.04519*. Version 1.

Wes Gurnee and Max Tegmark. 2024. Language Models Represent Space and Time. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with Language Model is Planning with World Model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring Massive Multitask Language Understanding. In *The Ninth International Conference on Learning Representations (ICLR 2011)*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring Mathematical Problem Solving With the MATH Dataset. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks*.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The Impact of Reasoning Step Length on Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1830–1842. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. *arXiv:2001.08361*. Version 1.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. volume 35, pages 22199–22213.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same Task, More Tokens: the Impact of Input Length on the Reasoning Performance of Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 15339–15353. Association for Computational Linguistics.

Dawei Li, Renliang Sun, Yue Huang, Ming Zhong, Bohan Jiang, Jiawei Han, Xiangliang Zhang, Wei Wang, and Huan Liu. 2025. Preference Leakage: A Contamination Problem in LLM-as-a-judge. *arXiv:2502.01534*. Version 1.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv:2501.19393*. Version 3.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Richard Ren, Ryan Kim, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Daron Anderson, Tung Nguyen, Imad Ali Shah, and 646 others. 2025. Humanity's Last Exam. *arXiv:2501.14249*. Version 4.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. In *First Conference on Language Modeling*.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, Alhussein Fawzi, Josh Grochow, Andrea Lodi, Jean-Baptiste Mouret, Talia Ringer, and Tao Yu. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Model Parameters. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2023. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*. Featured Certification.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let Me Speak Freely? A Study On The Impact Of Format Restrictions On Performance Of Large Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models. *arXiv:2404.18796*. Version 2.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Thoughts Are All Over the Place: On the Underthinking of o1-like LLMs. *arXiv:2501.18585*. Version 1.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, pages 24824–24837.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *37th International Conference on Neural Information Processing Systems (NeurIPS 2023)*, pages 11809 – 11822.

Qihuang Zhong, Kang Wang, Ziyang Xu, Juhua Liu, Liang Ding, Bo Du, and Dacheng Tao. 2024. Achieving> 97% on GSM8K: Deeply Understanding the Problems Makes LLMs Perfect Reasoners. *arXiv:2404.14963*. Version 4.

A Appendix

We describe our experimental setup and provide the data processing details necessary to replicate our analysis. At the end of this section, we elaborate on the regression analysis conducted to analyse the effect size of increased reasoning token usage on accuracy.

A.1 Datasets

The Omni-MATH benchmark (Gao et al., 2024) contains Olympiad-level math problems specifically designed to test the reasoning abilities of large language models. Each entry in the dataset consists of a problem, an exact answer, and a written out solution together with the following metadata fields: Domain, Difficulty, and Source (see Figure A1). Each problem has between one and three domains of the form Mathematics $\rightarrow$ Primary domain $\rightarrow \ldots$, with a maximum length of five. In this paper, we only take the primary domains into account, as a more granular classification gives rise to very imbalanced or underpopulated classes. Figure A3 shows the number of math problems per (primary) domain where we follow Gao et al. (2024) in double- or triple-counting the multi-domain questions. We made sure to delete the duplicate entries, e.g. some data entries had multiple domain trees but the same primary domain. Every domain-specific analysis in the paper follows this convention. Finally, we joined the Calculus and Pre Calculus class and deleted the Other class to obtain a more balanced domain distribution. Math problems are also classified according to difficulty level as presented in Figure A4. We divide the data into difficulty tiers based on the quartiles of the difficulty distribution (without separating difficulty levels).

A.2 Large Language Models

We evaluate the performance of the several OpenAI models that are affordable for most users: gpt-4o-06-08-2024, o1-mini-12-09-2024, o3-mini-31-01-2025 medium (default) and o3-mini-31-01-2025 high. The o3-mini high model, instead of medium, is obtained by setting `reasoning_effort` to high. We feed each model the math problems using the Batch API with the following vanilla prompt as user message:

> Solve the following problem. Enclose the final answer in a \\boxed{{}} environment.
> Problem: {problem}

Furthermore, we set `max_completion_tokens` limits of $25,000$ for o1-mini and o3-mini medium, and a $100,000$ token limit for o3-mini high. Each reasoning model refused to answer a few questions (flagged as invalid prompts), which were subsequently omitted from the analysis.

To correct the responses of the four OpenAI models on the Omni-MATH dataset, we employ another large language model called Omni-Judge (`KbsdJames/Omni-Judge`). Omni-Judge is an efficient and low cost open-source math-evaluation model developed by the authors of Gao et al. (2024). The model is trained to assess the correctness of an answer generated by an LLM, given the problem and a reference answer (see Figure A2). Table 9 in Gao et al. (2024) shows that Omni-Judge is 91.78% consistent with gpt-4o as a judge (who is 98% consistent with human evaluators) and has almost a 100% success rate of correctly parsing model generated answers. To judge the models' generated answers, we make requests to the chat completions endpoint of the `kbsdjames.omni-judge` API by running the model in LM Studio. We use the same few-shot prompt as in Gao et al. (2024) and set the `max_new_tokens` parameter to 300. In the very few cases where Omni-Judge fails to parse the model output ($< 1\%$), we omit that question from the performance evaluation.

A.3 Data and Code Availability

The original Omni-MATH dataset is available at https://huggingface.co/datasets/KbsdJames/Omni-MATH. The original Omni-Judge model is available at https://huggingface.co/KbsdJames/Omni-Judge. Data and code for this publication are attached in an anonymized zip file. The code is based on the Omni-MATH benchmark analysis code, publicly available at https://github.com/KbsdJames/Omni-MATH. We used Python 3.12.6 (*pandas 2.2.3*, *numpy 2.1.1*, *matplotlib 3.9.2*, *seaborn 0.13.2*, *statsmodels 0.14.4*, and *scikit-learn 1.5.2*) to analyse and visualize data and to conduct statistical analyses.

A.4 Conditional Probability

The conditional probability appearing in Figure 3 and Figure A5 is computed using a full Bayesian model with uninformative priors (we assume that $\mathbb{P}(\text{False}) = \mathbb{P}(\text{True}) = 0.5$). In particular, we have that

$$\mathbb{P}(\text{False} \mid T > B_i) = \frac{\mathbb{P}(T > B_i \mid \text{False})}{\mathbb{P}(T > B_i \mid \text{False}) + \mathbb{P}(T > B_i \mid \text{True})}, \tag{1}$$

where $\{T > B_i\}$ is the event that the number of tokens exceeds the right bin threshold and "False" indicates that the model answered incorrectly. Because B_i can only take a finite number of values, we have that

$$\mathbb{P}(\text{False} \mid T > B_i) \tag{2}$$

$$= \sum_{k=i+1}^{n} \frac{\mathbb{P}(T \in B_k \mid \text{False})}{\mathbb{P}(T \in B_k \mid \text{False}) + \mathbb{P}(T \in B_k \mid \text{True})} \tag{3}$$

$$= \sum_{k=i+1}^{n} \frac{|\text{False} \in B_k|}{|\text{False} \in B_k| + |\text{True} \in B_k|}, \tag{4}$$

which can be easily computed using the stacked histogram data.

A.5 Regression Analysis

We use a logistic regression to estimate the effect of additional reasoning tokens on the probability of an accurate response on a question Y_i, while controlling for different levels of difficulty and domains. The regression takes the following form:

$$\log\left(\frac{\Pr(Y_i = 1)}{\Pr(Y_i = 0)}\right) = \beta_0 + \beta_1 \text{tokens}_i + \underbrace{\sum_{k=1}^{K-1} \delta_k \text{ difficulty tier}_{k(i)}}_{\text{difficulty fixed effects}} + \underbrace{\sum_{m=1}^{M-1} \gamma_m \text{ domain}_{m(i)}}_{\text{domain fixed effects}}, \tag{5}$$

where i, k, and m denote the question-response pair, the difficulty tier, and the domain, respectively. Moreover, $k(i)$ and $m(i)$ indicate that the difficulty tier k and domain m depend on the question-response pair i. The difficulty tier and domain fixed effects can be estimated by including dummy variables, which are equal to one if the difficulty tier or domain is equal to the difficulty tier or domain of the current question-response pair and equal to zero otherwise, with the exclusion of a reference category (i.e., $K - 1$ and $M - 1$). The reference category for difficulty tiers is the lowest difficulty Tier 1 and for the domains it is Algebra. We obtain similar results when using the more fine-grained difficulty levels $(0 - 10)$ instead of difficulty tiers.

To facilitate interpretation, we compute the Average Marginal Effect (AME) of additional reasoning tokens on the probability of an accurate response. Unlike the raw logistic regression coefficients, which are expressed in log-odds, the AME directly quantifies the effect of an additional token in probability terms. Specifically, it represents the average change in the probability of accuracy for a one-token increase, while holding difficulty tier and domain constant. By computing AMEs, we ensure that our estimates account for the full distribution of difficulty levels and domains, rather than relying on effects evaluated at a single reference point.

A.6 Figures and Tables

Sample problem Omni-MATH

Domain: Mathematics → Discrete Mathematics → Graph Theory
Difficulty: 9.0
Source: China National Olympiad

Problem: A table tennis club hosts a series of doubles matches following several rules:

1. each player belongs to two pairs at most;
2. every two distinct pairs play one game against each other at most;
3. players in the same pair do not play against each other when they pair with others respectively.

Every player plays a certain number of games in this series. All these distinct numbers make up a set called the *set of games*. Consider a set $A = \{a_1, a_2, \ldots, a_k\}$ of positive integers such that every element in A is divisible by 6. Determine the minimum number of players needed to participate in this series so that a schedule for which the corresponding *set of games* is equal to set A exists.

Answer: $\frac{1}{2} \max A + 3$.

Solution: To determine the minimum number of players needed to participate in the series such that the set of games is equal to the set A, we start by analyzing the problem through graph theory.
Consider a graph $\mathcal{G}$ where each vertex represents a player and an edge between two vertices represents a pair of players. According to the problem's conditions:

(i) Each player belongs to at most two pairs.

(ii) Every two distinct pairs play one game against each other at most.

(iii) Players in the same pair do not play against each other when they pair with others respectively.

Given these conditions, each vertex in $\mathcal{G}$ can have a degree of at most 2. This implies that $\mathcal{G}$ can be decomposed into disjoint cycles, paths, and isolated vertices. Let $\max A = a$. We need to ensure that there is a player who plays a games, and each element in A is divisible by 6. To achieve this, we need to construct a graph where the number of games each player plays corresponds to the elements in A. ### Proof of optimality: Let x be a vertex with the maximum number of games a. This vertex x must be connected to at least one other vertex. If x has only one edge, it must have a games on it. If x has two edges, each edge must have at least $\frac{a}{2}$ games. Let xy be an edge with $\geq \frac{a}{2}$ games. The remaining vertices must account for at least $\frac{a}{2}$ edges, ensuring that there are at least $\frac{a}{2} + 2$ vertices. However, due to the constraints, there must be at least $\frac{a}{2} + 3$ vertices to satisfy all conditions. ### Construction: To construct such a graph, we can use clusters of three players, each forming pairs within the cluster. This ensures that each player in a cluster plays 6 games per edge. By assigning each cluster to a vertex in a graph $\mathcal{G}$ with b vertices, where $b = \frac{1}{6} \max A + 1$, we can ensure that the set of games corresponds to A. Thus, the minimum number of players needed is: $\frac{1}{2} \max A + 3$. The answer is:

$$\boxed{\frac{1}{2} \max A + 3}$$

Figure A1: **Sample problem from the Omni-MATH dataset.** The Omni-MATH dataset consists of 4428 Olympiad-level math problems together with an exact answer, a written out solution and metadata Domain, Difficulty and Source.

Sample output Omni-judge

Problem: A table tennis club hosts a series of doubles matches following several rules:

(i) each player belongs to two pairs at most;

(ii) every two distinct pairs play one game against each other at most;

(iii) players in the same pair do not play against each other when they pair with others respectively.

Every player plays a certain number of games in this series. All these distinct numbers make up a set called the *set of games*. Consider a set $A = \{a_1, a_2, \ldots, a_k\}$ of positive integers such that every element in A is divisible by 6. Determine the minimum number of players needed to participate in this series so that a schedule for which the corresponding *set of games* is equal to set A exists.

Answer: $\frac{1}{2} \max A + 3$.

Model generation: We will show that one may 'hit' a prescribed set $A = \{a_1, a_2, \ldots, a_k\}$ (where each a_i is a positive multiple of 6) by the following construction, and that no schedule may use fewer than

$$\frac{a_1 + a_2 + \cdots + a_k}{3} + 2k \quad (1)$$

players. [...] In any such schedule every pair plays some number of games. Rule (ii) means that once we have fixed a pair P the 'opponent-teams' of P form a set; we denote by $d(P)$ the number of games in which P plays. Then a given player's 'score' (the number of games in which he participates) is determined by whether he belongs to one pair (in which case his score is the degree of that pair) or to two pairs (in which case his score is the sum of the two degrees). Note that by rule (iii) if a player belongs to two pairs then those two pairs never meet so that the numbers add! Thus if we 'read off' the numbers of games played by the players (omitting any repetitions) we obtain a set - called the 'set of games'. In our problem we wish this set to be exactly the given A.
An important idea is that players who belong to two pairs are 'economical' in that one player now serves to 'cover' two teams. (In a 'non-economical' schedule each pair would contribute two 'single–members'.) One may show that in a schedule in which as many players as possible are 'doubles' (that is, belong to two teams) one may 'control' the scores so that in fact every player's number is of the form $d(P)$ or $d(P) + d(Q)$ (with $d(P)$ and $d(Q)$ coming from a suitable graph on the pairs). One may then prove the following two claims:

Claim 1. If the schedule uses only double–players then the total number n of players is exactly the number T of pairs, and a routine "pairing" (that is, a perfect matching) of the teams shows that one must have $n \geq 2k$ (since one obtains exactly $n/2$ sums, one for each game–of–a–double, and these must contain the k given numbers.)

Claim 2. Since a team's degree is at most $(n-1)$ the maximum score obtainable by a double is at most $2(n-1)$. In order to be able to "reach" a number a_i we must have $2(n-1) \geq a_i$ for every i, so that in particular $n \geq \frac{1}{2} \max\{a_i\}+1$. A careful double–counting (namely, counting the total number of games by summing the scores of players in two different ways) shows that one must have $n \geq \frac{a_1+a_2+\cdots+a_k}{3} + 2k$. Moreover one may show by an inductive construction that equality is attainable. One acceptable answer to the problem is therefore to prove that the minimum number of players needed is $\frac{a_1+a_2+\cdots+a_k}{3} + 2k$. An answer equivalent to (1) is completely acceptable. Thus, our final answer is

$$\boxed{\frac{a_1 + a_2 + \cdots + a_k}{3} + 2k}$$

Omni-Judge: ## Student Final Answer: $\frac{a_1+a_2+\cdots+a_k}{3} + 2k$. ## Equivalence Judgement: FALSE ## Justification: The student's answer is $\frac{a_1+a_2+\cdots+a_k}{3} + 2k$, which does not match the reference answer of $\frac{1}{2} \max A + 3$. The discrepancy lies in the treatment of the maximum value of A and the constant term. In the student's solution, they consider the average of all elements in set A, whereas the reference answer focuses on half of the maximum element in A plus an additional constant term. This difference indicates that the student's answer does not capture the same meaning as the reference answer. === report over ===

Figure A2: **Sample output of the Omni-Judge model.** The Omni-Judge model is prompted as in Gao et al. (2024), which is to repeat the 'student's' final answer, asses the equivalence of this answer with the reference answer and provide complementary justification. We omit the part where o3-mini (h) repeats the problem specifications, indicated by [...].

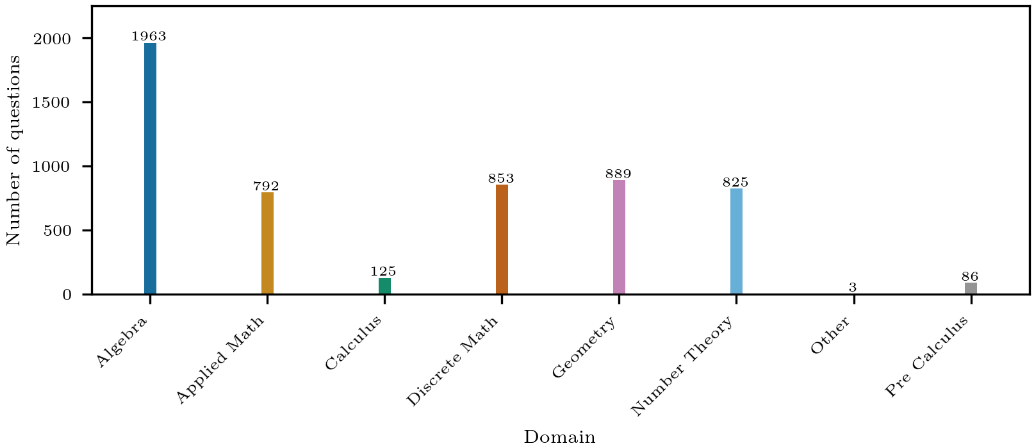

Figure A3: **Domain distribution of the Omni-MATH dataset.** This figure displays the distribution of the primary domains of the Omni-MATH dataset. Math problems that belong to multiple domains are counted for each domain, so the total number of question is higher than in Figure A4.

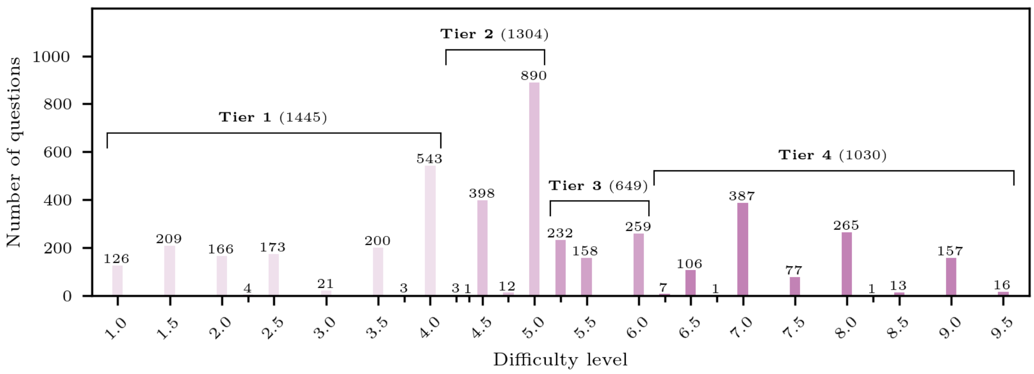

Figure A4: **Classification of difficulty levels in balanced difficulty tiers.** This figure shows the difficulty distribution of the Omni-MATH dataset. The difficulty levels are classified in difficulty tiers based on the quartiles of the distribution (without separating difficulty levels).

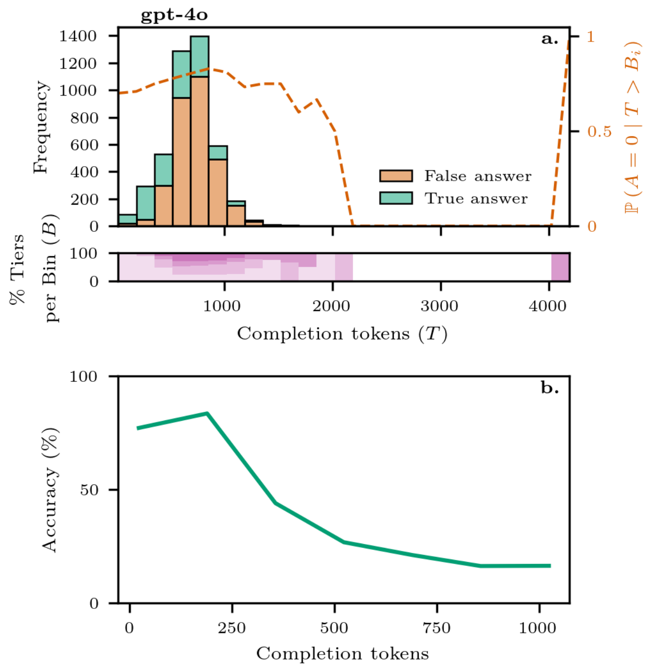

Figure A5: **Accuracy across the completion token distribution and accuracy vs. token use for gpt-4o.** This figure shows that gpt-4o uses predominantly between 200 and 1000 completion tokens for answering the Omni-MATH problems. We also observe that shorter answers are more likely to lead to a correct final answer. Finally, the relative proportion of tier levels in each bin reveals a clear transition from a region where the majority of the questions come from the lowest tiers to a region where the majority of the questions come from the highest tiers (for bins with a sufficient amount of data points). **a,** The main panel of this plot displays a stacked histogram of the reasoning tokens used for correctly and incorrectly answered questions in the Omni-MATH dataset. The secondary y-axis depicts the probability that the model answers incorrectly given that the token count has surpassed the bin threshold (see Appendix A.4). The subplot contains a filled histogram where the color opacity represents the difficulty level of the math questions (cfr. Figure A4). **b,** Accuracy per reasoning token, computed by dividing the number of correctly answered questions by the total number of questions in each bin of the histogram depicted in **a**.

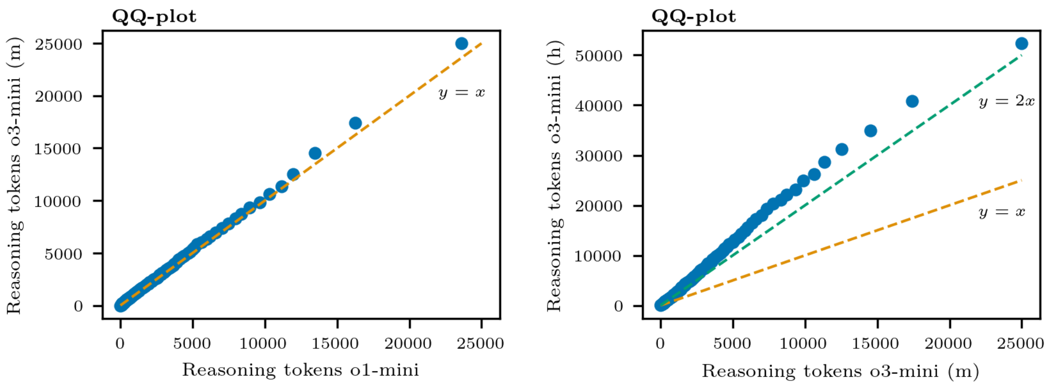

Figure A6: **Reasoning token distribution of o1-mini vs. o3-mini (m), and o3-mini (m) vs. o3-mini (h).** This figure compares the token distribution of three OpenAI reasoning models for *correctly* answered problems in the Omni-MATH dataset by means of a QQ-plot. We observe that o1-mini and o3-mini (m) have an almost identical reasoning token distribution. The token distribution of o3-mini (h) is a linearly scaled version of the distribution of o3-mini (m) with a factor slightly larger than 2.

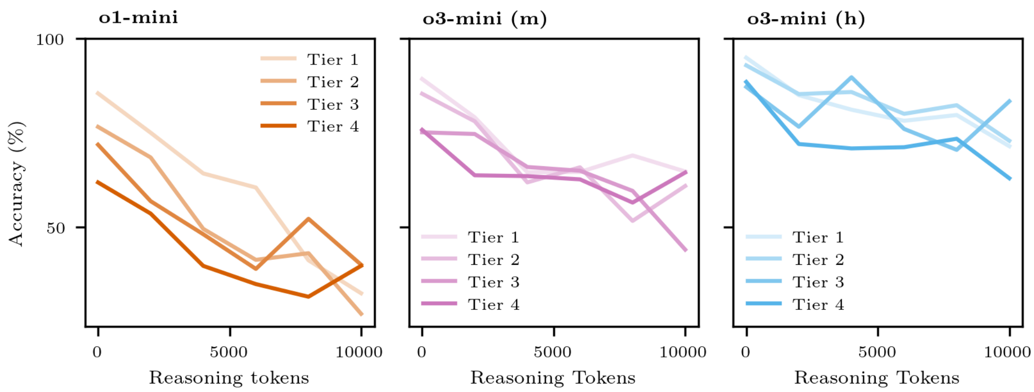

Figure A7: **Stratification of Figure 4a by difficulty tier of the Omni-MATH dataset.** This figure shows that accuracy also decreases within the difficulty tiers as the use of reasoning tokens increases.

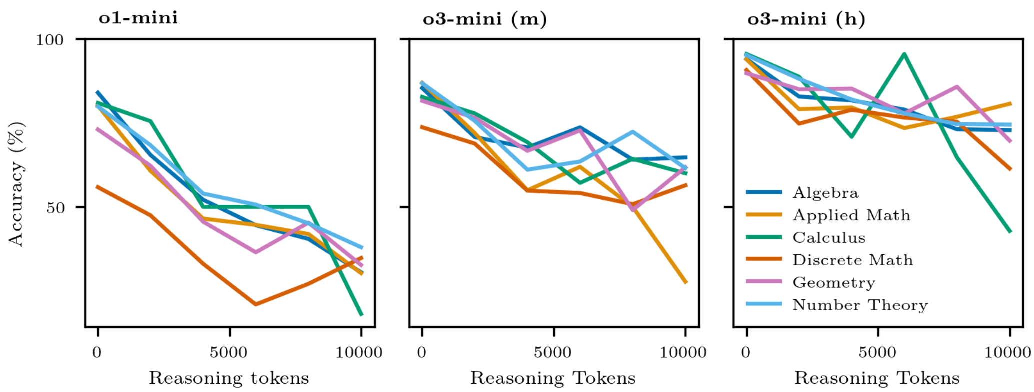

Figure A8: **Stratification of Figure 4a by mathematical domain of the Omni-MATH dataset.** This figure shows that, on average, accuracy also decreases within the domains as the use of reasoning tokens increases.

	o1-mini	o3-mini (m)	o3-mini (h)
Without controls			
Tokens	$-1.85\text{e}{-4}^{***}$	$-1.25\text{e}{-4}^{***}$	$-5.77\text{e}{-5}^{***}$
Constant	1.19^{***}	1.53^{***}	1.87^{***}
With controls			
Tokens	$-1.61\text{e}{-4}^{***}$	$-1.08\text{e}{-4}^{***}$	$-5.10\text{e}{-5}^{***}$
Difficulty Tier 2	-0.53^{***}	-0.36^{***}	-0.20^{**}
Difficulty Tier 3	-0.74^{***}	-0.56^{***}	-0.37^{***}
Difficulty Tier 4	-1.08^{***}	-0.70^{***}	-0.63^{***}
Applied Math	-0.41^{***}	-0.34^{***}	-0.37^{***}
Calculus	0.13	0.03	0.09
Discrete Math	-0.86^{***}	-0.50^{***}	-0.41^{***}
Geometry	-0.46^{***}	-0.21^{**}	-0.26^{**}
Number Theory	0.02	0.04	0.08
Other	0.47	0.02	-0.03
Constant	1.93^{***}	2.02^{***}	2.25^{***}
N	$5,535$	$5,531$	$5,526$
McFadden's pseudo-R^2 (without controls)	0.11	0.06	0.06
McFadden's pseudo-R^2	0.15	0.07	0.08

Table A1: **Logistic regression models to estimate the effect size of reasoning tokens on accuracy.** We use a logistic regression to estimate the effect of additional reasoning tokens on the probability of an accurate response on a question, while controlling for different levels of difficulty and domains. Estimates are from a logistic regression (Eq. 5) fit by maximum likelihood, with robust (Huber–White) standard errors to account for potential heteroskedasticity. The significance levels are for a two-sided Wald test with a null hypothesis of the regression coefficient being equal to zero ($^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$).