TuringQ: Benchmarking AI Comprehension in Theory of Computation

Pardis Sadat Zahraei[♠◇], **Ehsaneddin Asgari**[◇]
[♠] Computer Engineering Department, Sharif University of Technology, Iran
[◇] Qatar Computing Research Institute - QCRI, Qatar
paradisez2001@gmail.com
easgari@hbku.edu.qa

Abstract

We present TuringQ, the first benchmark designed to evaluate the reasoning capabilities of large language models (LLMs) in the theory of computation. TuringQ consists of 4,006 undergraduate and graduate-level question-answer pairs, categorized into four difficulty levels and covering seven core theoretical areas. We evaluate several open-source LLMs, as well as GPT-4, using Chain of Thought prompting and expert human assessment. Additionally, we propose an automated LLM-based evaluation system that demonstrates competitive accuracy when compared to human evaluation. Fine-tuning a Llama3-8B model on TuringQ shows measurable improvements in reasoning ability and out-of-domain tasks such as algebra. TuringQ serves as both a benchmark and a resource for enhancing LLM performance in complex computational reasoning tasks. Our analysis offers insights into LLM capabilities and advances in AI comprehension of theoretical computer science [1].

1 Introduction

The reasoning and comprehension capabilities of large language models (LLMs) are becoming increasingly critical due to their expanding applications across complex domains (Guo et al., 2023). As LLMs continue to evolve, robust benchmarks are essential for assessing their performance, particularly in areas that require deep understanding and logical reasoning (Brown et al., 2020; Ling et al., 2024). While multi-task benchmarks like BIG-Bench (Srivastava et al., 2022) cover a variety of domains, a notable gap remains, a dedicated dataset for evaluating LLM performance on theoretical concepts and problems within the domain of the theory of computation. This gap is significant, as assessing comprehension in formal languages and abstract computational problems is crucial to evaluating the true depth of an LLM's reasoning capabilities. Addressing this need is a key step toward transforming LLMs into sophisticated problem solvers in highly complex domains (Bender and Koller, 2020).

TuringQ fills this critical gap by providing a comprehensive platform for rigorously assessing and comparing the reasoning capabilities of different LLMs on intricate theoretical domains. It drives advancements in enhancing their skills for tackling computationally complex concepts, contributing to the development of more reliable and capable AI systems (Radford et al., 2019; Yang et al., 2023). Mastery of theory of computation principles is particularly vital, as these foundational concepts underpin modern computing systems and algorithms. Improving LLM comprehension in this domain could unlock new potential for reasoning about computational problems, analyzing algorithms, and even contributing to the creation of novel computational models and methodologies. Figure 1 presents a complete visual overview of our work. Our contributions are threefold:

1. **TuringQ Dataset**: We introduce a new resource of 4,006 theory of computation question-answer pairs from universities worldwide. This dataset spans undergraduate and graduate-level concepts across four difficulty levels and seven main areas, including a subset focused on theoretical essentials. It serves as a comprehensive tool for evaluating and fine-tuning LLMs in this domain.

2. **LLM-based Evaluation**: We explore the feasibility of leveraging LLMs themselves as evaluators for TuringQ (Zheng et al., 2024). By defining an 'AutoGrade-TQ' prompt using Llama3-8B, we investigate the potential

[1]The dataset, code, and fine-tuned model are publicly available: the fine-tuned model on HuggingFace, the dataset on HuggingFace Datasets, and the code, along with interactions with the language models, on GitHub.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 12267–12280
November 12-16, 2024 ©2024 Association for Computational Linguistics

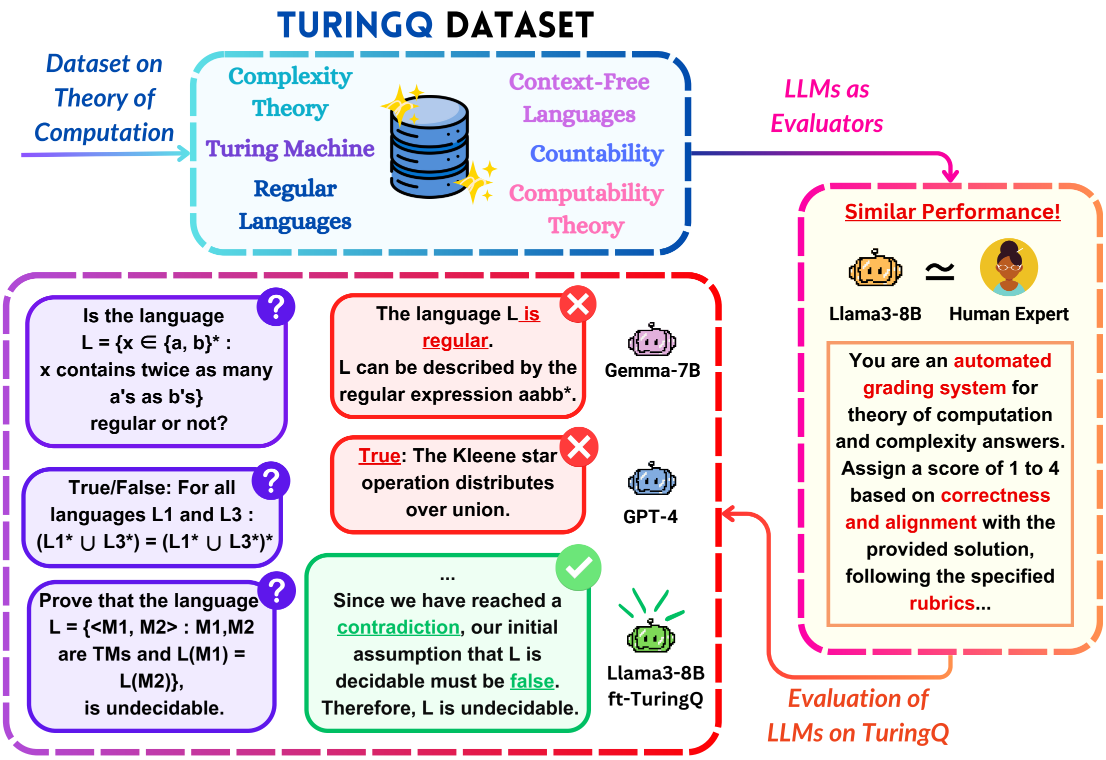


Figure 1: **TuringQ Dataset and its Evaluation Framework.** This diagram presents the TuringQ dataset, a comprehensive resource for theory of computation, and illustrates the automated assessment of LLMs using Llama3-8B. It showcases sample questions, LLM responses, and their evaluations by the AI evaluator. The fine-tuned Llama3-8B-ft-TuringQ model demonstrates improved performance but still encounters certain challenges in addressing TuringQ questions.

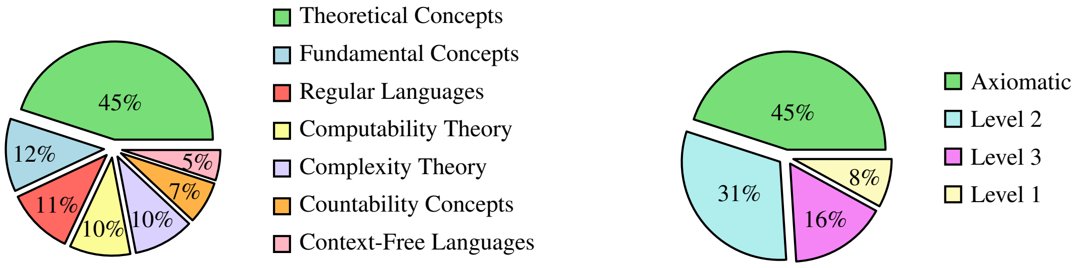
for automating the evaluation process, thereby reducing the time and cost associated with manual grading.

3. **Llama3-8B-ft-TuringQ Model**: We present a fine-tuned large language model, Llama3-8B-ft-TuringQ, specifically tailored for reasoning in the theory of computation. Through extensive evaluation, we provide a comparative analysis of the performance of large language models across various TuringQ categories, showcasing how our fine-tuned model competes with GPT-4.

2 Related Work

Evaluating Computational Reasoning Capabilities of LLMs Substantial progress has been made by large language models, but evaluating their mathematical and computer science reasoning remains an evolving challenge. Various datasets have been introduced to assess LLMs' mathematical reasoning abilities (Ahn et al., 2024), and approaches such as graph-based verification have been proposed to enhance reasoning (Cao, 2024). However, significant gaps remain, particularly in the domain of the formal theory of computation, where evaluation benchmarks and models' capabilities are less developed (Li et al., 2024; Frieder et al., 2023).

Automated LLM Evaluation Research on automating LLM evaluations has gained momentum, proposing techniques like self-consistency checks, external truth verification, and adversarial probing to improve evaluation accuracy (Huang et al., 2024; Chiang and yi Lee, 2023). LLMs have also been used to calibrate and augment human raters for evaluating generated text (Zhang et al., 2024). Hybrid approaches that combine human and LLM evaluations for assessing written content offer new insights into human-AI collaboration (Ren et al., 2024). However, questions remain regarding the trustworthiness of LLMs as evaluators, prompting research into scalable meta-evaluation mechanisms, such as agent debate (Chern et al., 2024). Aligning LLM-assisted evaluations with human preferences continues to be an active area of exploration (Shankar et al., 2024).



(a) Category Distribution of TuringQ

(b) Difficulty Distribution of TuringQ

Figure 2: Category and Difficulty Level Distribution in the TuringQ Dataset

3 The TuringQ Dataset

TuringQ is a comprehensive dataset comprising 4,006 question-answer pairs covering undergraduate and graduate-level theory of computation problems, aligned with Sipser's framework (Sipser, 2013). The questions are categorized into four difficulty levels and seven main conceptual areas: Regular Languages, Theoretical Concepts, Context-Free Languages, Computability Theory, Countability Concepts, Complexity Theory, and Fundamental Concepts. Details are provided in Appendix Table 18. The difficulty levels were determined by domain experts, ensuring an even distribution across categories and a clear distinction between difficulty levels and conceptual categories. The distribution of the dataset by category and difficulty level is illustrated in Figure 2. Examples of dataset entries are provided in Appendix Table 17.

3.1 Data Collection

We curated a collection of questions from publicly available exam sets and homework solutions from 29 top-tier universities to ensure a high-quality dataset in in the theory of computation domain. The primary dataset consists of 2,155 carefully selected university exam and homework questions, ensuring fair distribution across various categories. Additionally, 61 question-answer pairs from reputable non-university resources were incorporated. To complement the academic questions, we developed a secondary set focusing on fundamental concepts, theorems, lemmas, and essential knowledge. Domain experts identified these topics, and the Claude 3 Sonnet model (Anthropic, 2024) was utilized to generate 1,790 question-answer pairs covering the core principles of Theory of Computation.

3.2 Question Type Distribution

The TuringQ dataset includes three main types of questions: objective, analytical, and subjective. **Objective** questions, such as true/false or multiple-choice, have clear and unambiguous answers. **Analytical** questions require problem-solving and logical reasoning, while **subjective** questions involve providing explanations or descriptions.

The dataset emphasizes analytical thinking, with 48.1% of the questions classified as analytical. Additionally, 32% of the questions are subjective, focusing on explanatory capabilities, while 19.9% are objective, assessing factual knowledge. This distribution ensures a comprehensive assessment of language models across diverse cognitive tasks within the theory of computation domain.

4 Experiments

For further evaluation and analysis, we employ a diverse set of language models: Llama-3-8B-Instruct (Dubey et al., 2024), Llama-2-7B-chat-hf (Touvron et al., 2023), Mistral-7B-Instruct (Jiang et al., 2023), GPT-4-32k (OpenAI, 2023), Gemma-7B-it, and Gemma-2B-it (Team et al., 2024). To assess these models, we curated a stratified sample of 500 questions from the TuringQ dataset, maintaining the original distribution across difficulty levels and categories. This approach ensures a representative subset for our comparative analysis.

4.1 Evaluation Metrics

To assess the LLM performance, the evaluator assigns a score on a 1-to-4 scale, with higher values indicating superior quality. We then use binary accuracy, which classifies responses as valid (3-4) or invalid (1-2) and calculates the percentage of valid responses. To measure how well the LLM evaluator's scoring aligns with the human evaluator, we use two metrics: binary alignment and exact alignment. Binary alignment checks if both evaluators consider a response to be valid or invalid, while exact alignment checks whether the scores are exact matches.

4.2 AI-Driven Assessment

We employed the Llama3-8B model to generate responses using direct and Chain of Thought (CoT) prompts (Wei et al., 2023). To standardize evaluation, we created the AutoGrade-TQ prompt, enabling LLMs to score responses on a 1-4 scale. Three domain experts independently scored the responses for ground-truth evaluations, resulting in substantial agreement (Fleiss' Kappa $\kappa = 0.742$). Final scores were derived from majority voting. Detailed statistical measures are provided in Appendix Table 5, and specific prompts used are outlined in Appendix Table 16. The Llama3-8B model performed well as an evaluator, achieving 77.8% binary alignment, with CoT-generated responses consistently receiving higher scores.

4.3 Model Specialization

We fine-tuned Llama3-8B to create Llama3-8B-ft-TuringQ using our comprehensive dataset of detailed answers. Our approach combined Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023), Parameter-Efficient Fine-Tuning (PEFT) (Xu et al., 2023), and Supervised Fine-Tuning (SFT)[2]. We utilized three TuringQ-derived datasets: training (3,006 instances), validation (500 instances), and test (500 instances), generated through stratified sampling based on difficulty and category. Our process incorporated quantization and low-rank adaptation to optimize performance within computational constraints. Detailed setup and hyperparameters are provided in Appendix A.1.

5 Results

5.1 Performance Evaluation

We evaluated seven LLMs, including our fine-tuned model, Llama3-8B-ft-TuringQ, using the TuringQ test set. The evaluation involved two prompts: a Chain-of-Thought prompt to elicit responses from the LLMs and an AutoGrade-TQ prompt for automatic scoring. To establish a benchmark, three human annotators rated each answer, with the final human rating determined by majority vote. These human ratings served as a standard against which we compared the performance of the LLM evaluator and were used to assess the accuracy of LLM performance. Interestingly, as illustrated in Table 1, our fine-tuned model achieved an average binary accuracy of 81.2%, representing a 10% improvement over its base model, and performed comparably to GPT-4, which reached 84.8% accuracy, despite using limited resources.

Model	H Acc	L Acc	H Score	L Score
GPT-4	84.8%	82.4%	3.18	3.28
Llama3-8B-ft	81.2%	76.0%	3.08	2.98
Llama3-8B	71.2%	73.8%	3.05	3.03
Mistral-7B	76.4%	70.4%	2.92	2.99
Llama2-7B	72.6%	70.8%	2.85	3.02
Gemma-7B	68.2%	72.2%	2.76	3.02
Gemma-2B	59.4%	65.2%	2.59	2.87

Table 1: Performance Comparison of LLMs on the TuringQ Test Set: Mean Score and Binary Accuracy Evaluated by Humans (H) vs. Llama3-8B (L)

The results showed partial alignment between the LLM evaluator, Llama3-8B, and human evaluators. We observed that LLMs, when acting as evaluators, tended to overrate responses from weaker models and underrate those from stronger models compared to human evaluators. This suggests that LLMs may face challenges in accurately distinguishing between high- and low-quality answers. In the evaluation of true/false questions, the agreement between the LLM evaluator and human experts was notably strong, with our fine-tuned model achieving 80% binary alignment, while GPT-4 reached 78%. The remaining 20% discrepancy likely stems from instances where the LLM evaluator introduced its own reasoning during evaluation, highlighting an area for further investigation. Table 14 provides detailed statistical measures.

5.2 Category-Specific Performance Analysis

The category-specific analysis of the TuringQ dataset revealed contrasting trends between human evaluations and LLM evaluations. In the LLM evaluations, model performance was consistent across each category, with minimal variation. The best average performance was achieved in the 'complexity theory' category, with an accuracy of 81.51%, while the lowest performance was observed in the 'theoretical concepts' category, at 68.57% (details are provided in Appendix Tables 6 and 10). In contrast, human evaluations showed a different trend. The 'theoretical concepts' category achieved the highest scores across all metrics, with a significant gap compared to other categories. The average accuracy for this category was 89.93%, while the lowest performance was in the 'countability con-

[2]https://huggingface.co/docs/trl/en/sft_trainer

cepts' category, with an accuracy of 50.23% (details are provided in Appendix Tables 7 and 11). Notably, our fine-tuned model outperformed the base model in every category, demonstrating improved performance across all aspects of the theory of computation.

5.3 Impact of Difficulty Levels on Model Performance

A notable limitation of the LLM evaluator is highlighted in the analysis of difficulty levels. When evaluated by humans, the mean accuracy for level-3 questions was 48.91%, while the mean accuracy for axiomatic questions was significantly higher at 89.93% (details are provided in Appendix Tables 9 and 13). This outcome aligns with expectations, indicating that LLMs tend to perform better on easier questions. Conversely, when the LLM served as an evaluator, the results were reversed. For level-3 questions, the evaluator assigned a mean accuracy of 77.5%, whereas the mean accuracy for axiomatic questions dropped to 68.57% (details are provided in Appendix Tables 8 and 12).

This scoring discrepancy likely stems from the fundamental difference between LLM and human evaluation approaches. The LLM's rigid, pattern-matching assessment tends to favor complex answers, even when partially incorrect, while undervaluing simpler yet accurate responses. In contrast, human evaluators employ a holistic approach, considering comprehension and intent, thus more accurately assessing both simple correct answers and flawed complex ones.

5.4 Accuracy Breakdown by Data Source

We evaluated model performance across diverse data sources, including both university and non-university origins. Our fine-tuned model and GPT-4 demonstrated distinct strengths, particularly excelling in non-university and synthetic questions. For a comprehensive analysis, please refer to Appendix A.2.

5.5 Out-of-Domain Performance

To assess the generalization capabilities of our model and investigate potential overfitting, we evaluated Llama3-8B-ft-TuringQ on the challenging out-of-domain MATH Dataset (Hendrycks et al., 2021), which comprises competition-level mathematics problems. Our analysis involved a stratified sample of 500 questions from the MATH test split, allowing us to compare the performance of both the base Llama3-8B model and our fine-tuned version. We engaged human experts to evaluate the model's answers, and the results are shown in Table 2.

Score	Llama3-8B	Llama3-8B-ft
1	47.20%	44.40%
2	15.20%	17.40%
3	4.20%	4.60%
4	33.40%	33.60%

Table 2: Score Distribution of Human Evaluation: Llama3-8B vs. Llama3-8B-ft on MATH Test subset

The fine-tuned model achieved a 0.6% increase in binary accuracy, indicating that specialized fine-tuning did not compromise generalization. Instead, these results suggest that enhanced computational theory understanding may improve mathematical reasoning capabilities. Performance varied across MATH categories (Table 15), with improvements in 'Prealgebra' and 'Intermediate Algebra' but slight declines in 'Number Theory' and 'Precalculus'. This varied performance across mathematical domains highlights the complex relationship between computational theory training and general mathematical problem-solving abilities, warranting further investigation into the transfer of knowledge between these domains.

6 Conclusion

We introduced TuringQ to evaluate the reasoning capabilities of large language models (LLMs) in the domain of computation theory, encompassing four difficulty levels and seven core concepts. Our evaluation involved various open-source LLMs and GPT-4, utilizing Chain of Thought prompting and assessments from human experts. Additionally, we developed an automated evaluation system using a large language model, demonstrating both its potential and limitations. Fine-tuning a Llama3-8B model on TuringQ significantly improved its grasp of computation theory, as well as its performance on out-of-domain tasks such as algebra. This work establishes a valuable benchmark for assessing LLMs' understanding of computational theory. Evaluating comprehension of formal languages is essential for gauging the depth of LLMs' reasoning abilities, marking a significant advancement toward developing LLMs into effective problem solvers.

7 Ethics Statement

The TuringQ dataset comprises publicly available exams and homework questions from renowned universities worldwide, obtained from the internet. Each source is duly cited in the dataset's metadata, and no question has been extracted without acknowledgment of the original source. After data collection, we reviewed and enhanced some answers to maintain the dataset's high quality and ensure its value as a resource. This enhancement process did not involve any bias or alteration of the original content or answers.

For the theoretical concepts, we utilized the Claude 3 Sonnet model to generate answers for specified theorems and lemmas. Subsequently, we checked and edited the model-generated answers to ensure the absence of bias, hallucinations, or errors in our work.

In gathering solutions from non-university sources, we made efforts to include diverse, reliable references, such as computer science portals and textbooks. As the theory of computation and theoretical computer science is an evolving and complex field, we have included answers that reflect our current understanding, particularly regarding P, NP, and open problems. We acknowledge that as our knowledge progresses, some open questions in our dataset may require updates to their answers. However, to the best of our current knowledge, this dataset is up to date.

8 Limitations

This study encountered several limitations that future research should address. Firstly, computational resource constraints hindered our ability to utilize larger language models. Consequently, we focused on smaller yet powerful models that were more feasible for our research scope.

Evaluating descriptive questions posed a significant challenge. While we attempted various methods for assessing these questions, incorporating more extensive human evaluation would be beneficial. Although this approach is more resource-intensive and time-consuming, it could yield valuable insights into model performance.

While our dataset effectively captures the essential categories and fundamentals of the theory of computation, it lacks coverage of more applied tasks, such as code generation. Future research could investigate how fine-tuned, specialized models impact performance in related domains like code generation, reasoning, and mathematical problem-solving. It would be particularly interesting to explore the extent to which domain-specific fine-tuning may affect a model's general capabilities.

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *Preprint*, arXiv:2402.00157.

Anthropic. 2024. Introducing the next generation of claude.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, et al. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Lang Cao. 2024. Graphreason: Enhancing reasoning capabilities of large language models through a graph-based verification approach. *Preprint*, arXiv:2308.09267.

Steffi Chern, Ethan Chern, Graham Neubig, and Pengfei Liu. 2024. Can large language models be trusted for evaluation? scalable meta-evaluation of llms as evaluators via agent debate. *Preprint*, arXiv:2401.16788.

Cheng-Han Chiang and Hung yi Lee. 2023. A closer look into automatic evaluation using large language models. *Preprint*, arXiv:2310.05657.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Simon Frieder, Luca Pinchetti, Alexis Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, et al. 2023. Mathematical capabilities of chatgpt. *Preprint*, arXiv:2301.13867.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Supryadi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, and Deyi Xiong. 2023. Evaluating large language models: A comprehensive survey. *Preprint*, arXiv:2310.19736.

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. *Preprint*, arXiv:2103.03874.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Hui Huang, Yingqi Qu, Jing Liu, Muyun Yang, and Tiejun Zhao. 2024. Self-evaluation of large language model based on glass-box features. *Preprint*, arXiv:2403.04222.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, et al. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Xiaoyuan Li, Wenjie Wang, Moxin Li, Junrong Guo, Yang Zhang, and Fuli Feng. 2024. Evaluating mathematical reasoning of large language models: A focus on error identification and correction. *Preprint*, arXiv:2406.00755.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, et al. 2024. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *Preprint*, arXiv:2305.18703.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Cheng Ren, Zachary Pardos, and Zhi Li. 2024. Human-ai collaboration increases skill tagging speed but degrades accuracy. *Preprint*, arXiv:2403.02259.

Shreya Shankar, J. D. Zamfirescu-Pereira, Björn Hartmann, Aditya G. Parameswaran, and Ian Arawjo. 2024. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. *Preprint*, arXiv:2404.12272.

Michael Sipser. 2013. *Introduction to the Theory of Computation*, third edition. Course Technology, Boston, MA.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint*. ArXiv:2206.04615 [cs, stat].

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, et al. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, et al. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *Preprint*, arXiv:2312.12148.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *Preprint*, arXiv:2304.13712.

Mozhi Zhang, Mianqiu Huang, Rundong Shi, Linsen Guo, Chong Peng, et al. 2024. Calibrating the confidence of large language models by eliciting fidelity. *Preprint*, arXiv:2404.02655.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

A Appendix

A.1 Fine-tuning Setup and Hyperparameters

Our fine-tuning approach for the Llama3-8B model combined Quantized Low-Rank Adaptation (QLoRA), a Parameter-Efficient Fine-Tuning (PEFT) method, with Supervised Fine-Tuning (SFT) using the SFTTrainer from *HuggingFace's trl library*[3]. QLoRA, as a PEFT technique, allows for task-specific tuning without modifying all model parameters, while SFT provides a framework for supervised learning on our specific task. LoRA (Low-Rank Adaptation) freezes the LLM's weights and injects trainable rank-decomposition matrices (Hu et al., 2021). QLoRA extends this by incorporating quantization techniques, further reducing memory usage while maintaining or improving model performance. We configured the PEFT settings with the following hyperparameters:

- Alpha: 64
- Dropout rate: 0.05
- Optimizer: 'paged_adamw_8bit'
- Learning rate: 5e-6
- Learning rate scheduler: cosine
- Number of epochs: 3
- Max steps: 4000
- Batch size: 4 (for both training and evaluation)
- Gradient accumulation steps: 2

Evaluation was performed at every step, with results logged for detailed performance tracking. We employed quantization via the *BitsAndBytes method*[4], setting the compute data type to bfloat16 and loading the model in 4-bit with a quantization type of "nf4". This configuration enabled double quantization, potentially improving the efficiency of our model training. Our approach, combining QLoRA, SFT, and quantization techniques, allowed us to achieve high-quality results despite computational constraints.

[3] https://huggingface.co/docs/trl/en/index
[4] https://huggingface.co/docs/bitsandbytes/main/en/index

A.2 Accuracy Breakdown by Data Source

To provide a more nuanced understanding of model performance, we conducted an analysis of accuracy across different data sources in our test set. This analysis encompassed both intra-university comparisons and a broader inter-university analysis.

A.2.1 Intra-University Performance Analysis

We examined performance across four institutions with similar data distributions: New Jersey Institute of Technology, The University of Texas at Austin, UC San Diego, and University of Washington. Binary accuracies based on human evaluator scores for our fine-tuned model and GPT-4 were as follows:

Institution	Fine-tuned Model	GPT-4
New Jersey Institute of Technology	76.19%	85.71%
The University of Texas at Austin	71.43%	76.19%
UC San Diego	61.11%	77.78%
University of Washington	90.00%	60.00%

Table 3: Intra-university Performance Comparison

A.2.2 Inter-University Performance Analysis

We also compared performance between university-sourced questions and those from non-university or synthetic sources. Binary accuracies from human ratings showed:

Source	Fine-tuned Model	GPT-4
Non-university	95.65%	96.74%
University	72.78%	77.85%

Table 4: Inter-university Performance Comparison

It is important to note that these results may be influenced by factors such as difficulty levels, question types, and categories.

	Average	MSE	Variance	Correlation	Binary Alignment	Exact Alignment
Llama2-7B	3.494	1.758	1.4979	0.1169	0.6800	0.3440
Llama2-7B-CoT	3.456	1.656	1.4928	0.0478	0.7040	0.3520
Llama3-8B	2.858	1.746	1.7301	0.1772	0.6400	0.3180
Llama3-8B-CoT	3.032	1.268	1.2676	0.3408	0.7780	0.3520
Gemma-2B	3.2969	2.068	1.9737	0.1400	0.6784	0.3753
Gemma-2B-CoT	3.4854	2.006	1.8295	0.1463	0.7050	0.4121
Gemma-7B	3.1674	1.678	1.6520	0.0479	0.6801	0.2733
Gemma-7B-CoT	3.3162	1.524	1.4479	0.0355	0.7084	0.3203
Mistral-7B	3.454	1.538	1.3171	0.3474	0.7260	0.4520
Mistral-7B-CoT	3.374	1.686	1.5823	0.2632	0.7120	0.4620
GPT-4	2.69	1.390	1.3036	0.5103	0.7000	0.4880
GPT-4-CoT	2.366	2.106	1.6354	0.3906	0.6080	0.3980
Human	2.984	-	-	-	-	-
Human-CoT	3.052	-	-	-	-	-

Table 5: Statistical Measures of LLM Performance as Evaluators on the TuringQ Test Set: Direct and Chain-of-Thought (CoT) Prompt Answers of Llama3-8b

Category	llama3-8B	Llama3-8B-ft-TuringQ	Gemma-2B	Gemma-7B	llama2-7B	Mistral-7B	GPT-4
Complexity Theory	**3.1**	3.1	3.0	3.2	3.1	3.2	3.4
Computability Theory	**3.1**	**3.3**	3.1	**3.3**	3.2	**3.3**	3.4
Context-Free Languages	2.8	**3.3**	**3.2**	**3.3**	**3.4**	3.1	3.4
Countability Concepts	2.9	3.2	2.8	2.9	3.2	2.8	**3.6**
Fundamental Concepts	**3.1**	3.1	3.0	3.1	3.3	2.9	3.2
Regular Languages	**3.1**	3.0	3.0	3.2	3.2	3.1	3.4
Theoretical Concepts	3.0	2.8	2.7	2.9	2.8	2.9	3.2

Table 6: Comparative Analysis of Mean Scores Across Categories: Evaluated by Llama3-8B

Category	llama3-8B	Llama3-8B-ft-TuringQ	Gemma-2B	Gemma-7B	llama2-7B	Mistral-7B	GPT-4
Complexity Theory	2.9	3.0	2.4	2.5	2.8	2.8	3.0
Computability Theory	2.6	3.0	2.3	2.5	2.5	2.5	2.8
Context-Free Languages	2.5	2.9	2.1	2.5	2.6	2.8	2.7
Countability Concepts	2.4	2.8	2.4	2.4	2.6	2.2	2.7
Fundamental Concepts	2.8	3.1	2.6	2.8	3.0	3.1	**3.6**
Regular Languages	2.5	2.7	2.0	2.2	2.4	2.5	3.1
Theoretical Concepts	**3.5**	**3.3**	**3.0**	**3.1**	**3.1**	**3.2**	3.3

Table 7: Comparative Analysis of Mean Scores Across Categories: Evaluated by Human Expert

Difficulty	llama3-8B	Llama3-8B-ft-TuringQ	Gemma-2B	Gemma-7B	llama2-7B	Mistral-7B	GPT-4
Axiomatic	3.0	2.8	2.7	2.9	2.8	2.9	3.2
Level 1	2.9	3.0	2.9	2.9	**3.2**	2.8	3.0
Level 2	**3.1**	**3.2**	3.0	**3.2**	**3.2**	**3.1**	3.4
Level 3	3.0	**3.2**	**3.1**	**3.2**	3.1	**3.1**	**3.5**

Table 8: Comparative Analysis of Mean Scores Across Difficulty Levels: Evaluated by Llama3-8B

Difficulty	llama3-8B	Llama3-8B-ft-TuringQ	Gemma-2B	Gemma-7B	llama2-7B	Mistral-7B	GPT-4
Axiomatic	**3.6**	**3.3**	**3.0**	**3.1**	**3.1**	**3.2**	3.3
Level 1	2.7	3.2	2.5	2.8	2.8	3.1	**3.5**
Level 2	2.7	3.0	2.3	2.6	2.8	2.7	3.1
Level 3	2.6	2.6	2.1	2.2	2.4	2.4	2.8

Table 9: Comparative Analysis of Mean Scores Across Difficulty Levels: Evaluated by Human Expert

Category	llama3-8B	Llama3-8B-ft-TuringQ	Gemma-2B	Gemma-7B	llama2-7B	Mistral-7B	GPT-4
Complexity Theory	**81.2%**	83.3%	75.0%	**83.3%**	81.2%	**81.2%**	85.4%
Computability Theory	74.5%	88.2%	**76.5%**	78.4%	76.5%	80.4%	84.3%
Context-Free Languages	66.7%	**88.9%**	74.1%	74.1%	81.5%	74.1%	77.8%
Countability Concepts	66.7%	78.8%	60.6%	63.6%	75.8%	60.6%	**90.9%**
Fundamental Concepts	72.1%	78.7%	68.9%	73.8%	**82.0%**	65.6%	77.0%
Regular Languages	75.4%	75.4%	73.7%	71.9%	75.4%	70.2%	84.2%
Theoretical Concepts	74.0%	69.1%	57.0%	69.1%	61.0%	68.2%	81.6%

Table 10: Comparative Analysis of Mean Binary Accuracy Across Categories: Evaluated by Llama3-8B

Category	llama3-8B	Llama3-8B-ft-TuringQ	Gemma-2B	Gemma-7B	llama2-7B	Mistral-7B	GPT-4
Complexity Theory Concepts	60.4%	75%	45.8%	52.1%	64.6%	68.8%	72.9%
Computability Theory	58.8%	70.6%	41.2%	54.9%	56.9%	56.9%	66.7%
Context-Free Languages	48.1%	66.7%	29.6%	48.1%	51.9%	66.7%	59.3%
Countability Concepts	45.5%	63.6%	45.5%	45.5%	54.5%	39.4%	57.6%
Fundamental Concepts	62.3%	78.7%	54.1%	67.2%	75.4%	83.6%	95.1%
Regular Languages	52.6%	64.9%	31.6%	42.1%	52.6%	54.4%	80.7%
Theoretical Concepts	**90.1**%	**94.2**%	**80.7**%	**87.4**%	**87.4**%	**92.8**%	**96.9**%

Table 11: Comparative Analysis of Mean Binary Accuracy Across Categories: Evaluated by Human Expert

Difficulty	llama3-8B	Llama3-8B-ft-TuringQ	Gemma-2B	Gemma-7B	llama2-7B	Mistral-7B	GPT-4
Axiomatic	74.0%	69.1%	57.0%	69.1%	61.0%	68.2%	81.6%
Level 1	65.9%	68.3%	68.3%	68.3%	78.0%	61.0%	68.3%
Level 2	**78.2%**	**84.0%**	71.2%	75.6%	**79.5%**	73.7%	85.3%
Level 3	68.8%	83.8%	**75.0%**	**76.2%**	77.5%	**75.0%**	**86.2%**

Table 12: Comparative Analysis of Mean Binary Accuracy Across Difficulty Levels: Evaluated by Llama3-8B

Difficulty	llama3-8B	Llama3-8B-ft-TuringQ	Gemma-2B	Gemma-7B	llama2-7B	Mistral-7B	GPT-4
Axiomatic	**90.1**%	**94.2**%	**80.7**%	**87.4**%	**87.4**%	**92.8**%	**96.9**%
Level 1	51.2%	80.5%	48.8%	63.4%	65.9%	80.5%	90.2%
Level 2	59.6%	73.7%	46.2%	56.4%	67.3%	66.0%	75.6%
Level 3	51.2%	60.0%	31.2%	40.0%	45.0%	48.8%	66.2%

Table 13: Comparative Analysis of Mean Binary Accuracy Across Difficulty Levels: Evaluated by Human Expert

Model	MSE	Variance	Correlation	Binary Alignment	Exact Alignment
GPT-4	1.08	1.07	0.19	0.77	0.49
Llama3-8B-ft	1.11	1.10	0.18	0.72	0.41
Gemma-2B	1.72	1.64	0.10	0.56	0.29
Gemma-7B	1.58	1.51	0.11	0.64	0.35
Llama2-7B	1.49	1.46	0.11	0.61	0.33
Mistral-7B	1.30	1.29	0.16	0.66	0.40
Llama3-8B	1.27	1.27	0.34	0.74	0.33

Table 14: Statistical Measures: Human Evaluator vs. LLM Evaluator for Each Model

Type	Llama3-8B	Llama3-8B-ft	Delta
Number Theory	32.08%	28.30%	-3.78%
Prealgebra	58.62%	62.07%	+3.45%
Precalculus	27.27%	21.82%	-5.45%
Geometry	31.25%	33.33%	+2.08%
Intermediate Algebra	14.29%	18.68%	+4.39%
Algebra	55.46%	56.30%	+0.84%
Counting & Probability	23.40%	21.28%	-2.12%

Table 15: Comparative Analysis of Mean Binary Accuracy Across Categories on the MATH Test Set: Evaluated by Human Expert

Chain of Thought	You are a knowledgeable AI assistant specialized in Theory of Computation and Complexity. You will be answering questions related to this domain. To provide a clear and structured response, you will follow the Chain of Thought approach: Chain of Thought: 1. Analyze the question and identify core concepts, algorithms or problems. 2. Build a step-by-step solution approach, stating assumptions, defining variables/notations, and listing intermediate steps. 3. For proofs or complex calculations, show work explicitly, using relevant theorems, lemmas, or properties. 4. For true/false statements, provide clear justification or counterexample. 5. Review your Chain of Thought for logical soundness and completeness. Use clear and concise language, avoiding unnecessary jargon.
AutoGrade-TQ	You are an automated grading system for evaluating answers in the field of theory of computation and complexity. Your task is to assign a score (1, 2, 3, or 4) to a given answer based on its correctness and alignment with the provided solution, following the rubrics outlined below. Rubrics: Level 4 (Excellent): - Answer is completely correct and aligns perfectly with the provided solution. - Proofs, descriptions, true/false justifications, and calculations match the solution with no errors or omissions. - Demonstrates a comprehensive understanding of the concepts. Level 3 (Good): - Answer is mostly correct, with only minor deviations or omissions compared to the provided solution. - Proofs, descriptions, justifications, and calculations are largely accurate but may have a few minor flaws - Shows a strong grasp of the key concepts. Level 2 (Flawed): - Answer has some significant differences or incorrect elements compared to the provided solution. - Proofs, descriptions, justifications, and calculations contain several errors or omissions, but the core approach is partially valid. - Demonstrates a basic understanding of the concepts but lacks depth. Level 1 (Poor): - Answer deviates substantially from the provided solution. - Proofs, descriptions, justifications, and calculations are mostly incorrect or entirely missing crucial components. - Exhibits a lack of understanding of the fundamental concepts. Please note that the length of the answer should not be a factor in determining the score. The focus should be solely on the correctness and alignment with the provided solution. Given Answer: Answer Solution: Solution Based on the rubrics and the provided solution, assign a score (1, 2, 3, or 4) to the given answer.

Table 16: Prompts Employed for Automated Grading and Answer Generation via Chain of Thought Reasoning

Question	Answer
Show that the following is not regular. L = {ww : w $\in \{a, b\}^*$} *Difficulty: Level 2* *Category: Regular Expression* *Source: The University of Texas at Austin*	L = {ww : w $\in \{a, b\}^*$}. We'll use the pumping lemma. Don't get confused by the use of the variable w both to define L and as the name for the string we will choose to pump on. As is always the case, the only real work we have to do is to choose an appropriate string w. We need one that is long enough (i.e., |w| $\geq$ N). And we need one with firm boundaries between regions. So let's choose w = aˆNbaˆNb. Since |xy| $\leq$ N, we know that y must occur in the first a region. Clearly if we pump in any additional a's, the two halves of w will no longer be equal. Therefore L is not regular.
Give a context-free grammar that generate the following language. {w $\in \{0, 1\}^*$, the length of w is odd and the middle symbol is 0} *Difficulty: Level 2* *Category: Context-Free Languages* *Source: New Jersey Institute of Technology*	G = (V,Σ,R,S) with set of variables V = {S}, where S is the start variable; set of terminals Σ = {0,1}; and rules S $\rightarrow$ 0S0 | 0S1 | 1S0 | 1S1 | 0
Consider the language L = {ww^R}. Describe a two tape Turing machine to accept L. *Difficulty: Level 3* *Category: Computability Theory* *Source: The University of Texas at Austin*	The two tape machine works as follows: If the input is ϵ, accept. If not, copy the input to the second tape and record in the state that you have processed an even number of characters so far. Now, start the first tape at the left end and the second tape at the right end. Check that the symbols on the two tapes are the same. If not, reject. If so, move the first tape head to the right and the second tape head to the left. Also record that you have processed an odd number and continue, each time using the state to keep track of whether you've seen an even or odd number of characters so far. When you reach the end of the input tape, accept if you've seen an even number of characters. Reject if you've seen an odd number. (The even/odd counter is necessary to make sure that you reject strings such as aba.)
Let A be the set {x, y, z} and B be the set {x, y}. What is A × B? *Difficulty: Level 1* *Category: Fundamental Concepts* *Source: Sipser Book*	A x B = {(a, b) : a $\in$ A and b $\in$ B} = {(x, x), (x, y), (y, x), (y, y), (z, x), (z, y)}
What is the pumping lemma for regular languages? *Difficulty: Axiomatic* *Category: Theoretical Concepts* *Source: Claude3 + human*	The pumping lemma for regular languages states that if A is a regular language, then there exists a number p such that for any string s in A with length greater than or equal to p, there exist strings x, y, and z where s can be written as xyz, satisfying the following conditions: (1) xy^iz belongs to A for each i greater than or equal to 0, (2) y is not an empty string, and (3) the length of xy is less than or equal to p.

Table 17: Sample Instances from the TuringQ Dataset

Category	Description
Regular Languages	Regular languages are a class of formal languages that can be defined by regular expressions or recognized by finite automata with a finite number of states. Key topics in this category include determining whether a given language is regular or not, often employing techniques like the pumping lemma or constructing regular expressions. Additionally, concepts like deterministic finite automata (DFAs) and nondeterministic finite automata (NFAs) are fundamental, addressing the recognition of strings in regular languages through state transitions based on an input alphabet.
Context-Free Languages	A context-free language is a formal language that can be precisely defined by a context-free grammar, which consists of a set of production rules specifying how strings of symbols can be derived or generated, regardless of the context in which the symbols appear. Key concepts in the study of context-free languages include context-free grammars themselves, the processes of derivation and parse trees for visualizing derivations, as well as techniques for proving whether a given language is context-free or not.
Computability Theory	Computability Theory is a branch of theoretical computer science that deals with the limitations and capabilities of computational models, particularly in determining which problems are computationally solvable and which are not. Core concepts include Turing machines, decidability, Turing-recognizable languages, the Church-Turing thesis, and undecidability.
Complexity Theory	Complexity Theory is a branch of computer science that classifies computational problems based on their inherent difficulty and resource requirements. It analyzes time and space complexity using notations like Big O, and categorizes problems into complexity classes such as P, NP, NP-Complete, and PSPACE. Key concepts include polynomial time solvability, NP-Completeness for hardest problems in NP, and reducibility for relating problem complexities.
Countability Concepts	Countability concepts revolve around distinguishing between countable and uncountable sets, as well as characterizing the sizes of infinite sets. Key ideas include countable vs. uncountable sets, cardinal numbers and infinite cardinals, bijections and enumeration techniques, diagonalization methods for proving uncountability, the notion of cardinality as a measure of set size, and combinatorial principles like combinations and permutations. These concepts from set theory, combinatorics, and measure theory are crucial for understanding the nature of infinity.
Fundamental Concepts	Fundamental Concepts are the essential and introductory topics, including Set Theory, Propositional and Predicate Logic, and Relations. Set Theory covers sets, operations, and relations. Logic encompasses logical operators, truth tables, well-formed formulas, and quantifiers. Relations involve properties like reflexivity, symmetry, transitivity, equivalence relations, and partitions.
Theoretical Concepts	Theoretical Concepts in the theory of computation comprise the principles, theorems, rigorous proofs, lemmas, and auxiliary results that constitute the backbone of the field. These concepts lay the groundwork, illuminate pivotal results through meticulous derivations, and foster a profound understanding by elucidating connections and delineating boundary conditions. Mastering these Theoretical Concepts equips one with a robust theoretical foundation.

Table 18: Details and Interpretation of the TuringQ Dataset Categories