TuringQ: Benchmarking AI Comprehension in Theory of Computation

Anonymous ACL submission

Abstract

We present TuringQ, to the best of our knowledge, the first effort to evaluate the reasoning capabilities of large language models (LLMs) in the theory of computation. TuringQ consists of 4,006 question-answer pairs spanning undergraduate and graduate-level problems collected from a diverse set of universities. It covers three difficulty levels and six main concepts, including a valuable subset of axioms and essential theoretical concepts. We evaluated various open-source LLMs and GPT-4 using Chain of Thought prompting and human expert assessment. Additionally, we explored an automated LLM-Judge, demonstrating its potential to compete with human precision. We show that fine-tuning an LLaMA-3B model on TuringQ improves its reasoning ability. TuringQ serves as both a benchmark and a fine-tuning resource for enhancing LLM reasoning in this complex domain. Our comparative analysis reveals insights into LLM performance, contributing to advancements in AI comprehension of theoretical computer science. [1]

1 Introduction

The reasoning and comprehension capabilities of large language models across complex domains are crucial due to their recent vast number of applications (Guo et al., 2023). As LLMs grow in capability, robust benchmarks are needed to accurately assess their performance, especially in domains requiring deep understanding and logical reasoning (Brown et al., 2020; Ling et al., 2024). While efforts like BIG-Bench (Srivastava et al., 2022) have introduced multi-task benchmarks across various domains, a dedicated dataset to assess LLM performance on theoretical concepts and problems in the theory of computation has been notably absent. Assessing comprehension in formal languages is particularly important to understand the depth of LLMs' reasoning abilities. This can be a significant step toward developing LLMs into effective problem solvers in complex domains (Bender and Koller, 2020).

TuringQ provides a robust platform to rigorously assess and compare the reasoning capabilities of different LLMs on complex theoretical domains, driving advancements in enhancing their skills for tackling intricate computational concepts and contributing to the development of more capable and reliable AI systems (Radford et al., 2019; Yang et al., 2023). Moreover, a strong grasp of theory of computation principles is crucial for LLMs as these foundational concepts underpin modern computing systems. Enhancing LLM comprehension in this domain can unlock their potential for reasoning about computational problems, analyzing algorithms, and potentially contributing to the development of new computational models and methodologies (Sipser, 2006). 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 three 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-TuringQ prompt using Llama-3-8b, we investigate the potential for automating the evaluation process, thereby reducing the time and cost associated with manual grading.

[1] The dataset, code, and fine-tuned model will be made publicly available upon publication.

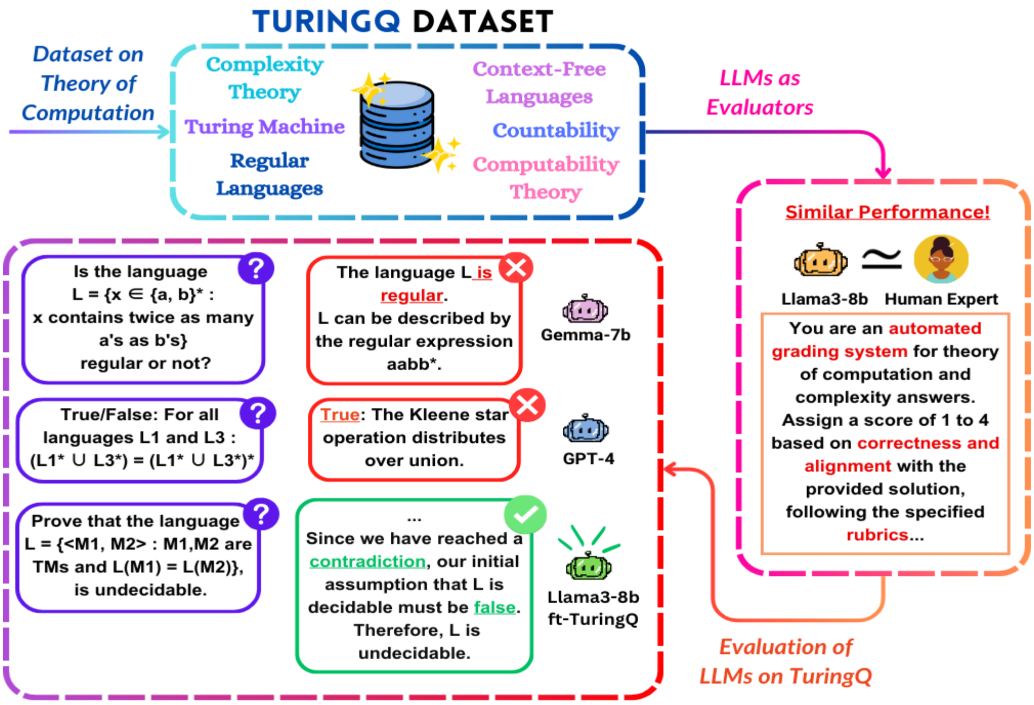


Figure 1: **TuringQ Dataset and its Evaluation Framework.** This diagram presents the TuringQ dataset, a comprehensive resource for theory of computation, and illustrates the automated assessment of LLMs using Llama3-8b. It showcases sample questions, LLM responses, and their evaluation by the AI evaluator.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

3. **Llama3-8b-ft-TuringQ Model**: We fine-tuned a large language model on the TuringQ dataset, creating Llama3-8b-ft-TuringQ, a model specialized for theory of computation reasoning. Through comprehensive evaluation using custom metrics, we provide a comparative analysis of LLM performance across different TuringQ categories, shedding light on their ability to tackle complex queries relative to human performance.

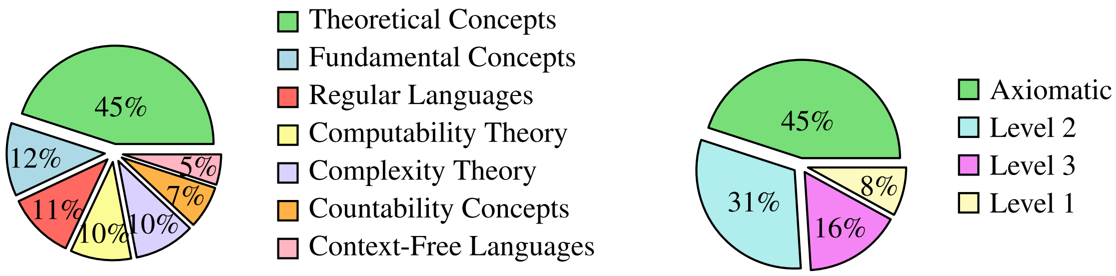
2 Related Works

Evaluating Reasoning Capabilities of LLMs Large Language Models have shown remarkable progress, but evaluating their mathematical and computer science reasoning capabilities is still an evolving field (Frieder et al., 2023; Li et al., 2024; Ahn et al., 2024). While various datasets have been introduced to assess mathematical reasoning abilities (Ahn et al., 2024), and approaches like graph-based verification have been proposed to enhance reasoning (Cao, 2024), the theory of computation domain awaits similar advancements.

Automated LLM Evaluation Automated evaluation of large language models is an active area of research. Various techniques, such as self-consistency, truth-checking against external data, and adversarial probing, have been proposed to enable LLMs to evaluate their own outputs (Huang et al., 2024). Parallel studies have explored using LLMs to calibrate and augment human raters for evaluating text generation outputs (Zhang et al., 2024). The combination of LLM evaluations with human grading for written assessments has also been investigated, providing a novel perspective on human-AI collaboration (Ren et al., 2024). However, the trustworthiness of LLMs for evaluation has been questioned, leading to proposals for scalable meta-evaluation of LLMs as evaluators via agent debate (Chern et al., 2024). Additionally, research has focused on aligning LLM-assisted evaluation of LLM outputs with human preferences (Shankar et al., 2024). These works contribute to the understanding and enhancement of automated LLM evaluators.

3 The TuringQ Dataset

TuringQ is a comprehensive dataset comprising 4,006 question-answer pairs covering undergraduate and graduate-level theory of computation prob-



(a) Category Distribution of TuringQ

(b) Difficulty Distribution of TuringQ

Figure 2: Category and Difficulty level Distribution of TuringQ

lems. The questions are categorized into three difficulty levels and seven main conceptual areas: Regular Languages, Theoretical Concepts, Context-Free Languages, Computability Theory, Countability Concepts, Complexity Theory, and Fundamental Concepts, as detailed in Table 9. 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 based on category and difficulty level is illustrated in Figure 2. Examples of dataset entries are provided in Table 8.

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 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.

4 Experiments

For further evaluation and analysis, we employ a diverse set of language models: Llama-3-8B-Instruct (Meta, 2024), Llama-2-7b-chat-hf (Touvron et al., 2023), Mistral-7B (Jiang et al., 2023), Gemma-7b-it (Team et al., 2024), and GPT-4-32k (OpenAI, 2023). 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 AI-Driven Assessment

We used Llama-3-8b to generate responses using direct and Chain of Thought (CoT) prompts (Wei et al., 2023). To evaluate LLMs as assessors, we developed the 'AutoGrade-TQ' prompt, guiding models to score answers on a 1-4 scale. Three in-house domain experts provided ground-truth evaluations with substantial inter-rater agreement (Fleiss' Kappa $\kappa = 0.742$). Majority votes were derived from their scores. Models evaluated both CoT and simple answers. Analysis suggests LLMs can be effective evaluators, with Llama3-8b achieving 77.8% binary accuracy. Key findings include:

CoT answers generally received higher scores, improving performance in open-source models. GPT-4 showed the lowest alignment with human evaluators. GPT-4 led in 4-level accuracy (49%), while Llama3-8b led in 2-level accuracy (77.8%). Llama3-8b and human evaluators' average scores for CoT answers were nearly identical. Full prompt details and statistics are presented in Tables 2 and 7.

4.2 Model Specialization

We fine-tuned the Llama3-8b model, resulting in Llama3-8b-ft-TuringQ, using our extensive dataset of detailed answers to enhance its performance on specific tasks. Our approach combined Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023), a Parameter-efficient Fine-tuning (PEFT) technique (Xu et al., 2023), and Supervised Fine-Tuning (SFT)[2]. We utilized three datasets derived from TuringQ for fine-tuning: a training set (3,006 instances), a validation set (500 instances), and a

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

test set (500 instances), generated using stratified sampling based on difficulty level and category for balanced representation. Our fine-tuning process incorporated advanced techniques like quantization and low-rank adaptation to optimize performance within computational constraints. Despite limitations, we achieved high-quality results, and further fine-tuning could yield better performance. Setup and hyperparameters are detailed 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. Assessment utilized a Chain-of-Thought prompt and an AutoGrader prompt for automatic evaluation. We measured performance using score and binary accuracy metrics. The score metric quantifies response quality, while binary accuracy classifies answers as valid or invalid based on the score, providing a more comprehensive assessment of answer correctness. As shown in Table 1, Llama3-8b-ft-TuringQ increased binary accuracy by 2.2%, a significant improvement given the computational resources used. This enhancement primarily resulted from an increase in responses with a score of 3. While performances across models were similar, GPT-4 only slightly outperformed others despite its superior capabilities, highlighting the challenges LLMs face with TuringQ questions. Figure 4 shows the score distribution for each model.

5.2 Category-Specific Performance Analysis

Analysis of the seven categories in the TuringQ dataset revealed consistent model performance across categories without drastic differences. Contrary to expectations, the "theoretical concepts" category did not yield the highest scores, potentially due to its more descriptive manner compared to other categories. The best performance was observed in the context-free languages category. GPT-4 exhibited exceptional performance in the "Countability" concepts category, achieving 90.9% accuracy—23.2% higher than the average binary accuracy of open-source models (Table 5). The fine-tuned model outperformed Llama3-8b in every category except theoretical concepts, where it showed a 5% decrease. In context-free languages, it demonstrated a substantial 22% increase compared to Llama3-8b (Figure 3).

Model	Mean Score	Binary Accuracy
GPT-4	3.276	82.40%
Llama3-8b-ft-TuringQ	2.984	76.00%
Llama3-8b	3.030	73.80%
Gemma-7B	3.022	72.20%
LLaMA-2-7B	3.020	70.80%
Mistral-7B	2.986	70.40%
Gemma-2B	2.872	65.20%

Table 1: Comparative Performance Metrics of Language Models on the TuringQ Test Set

5.3 Impact of Difficulty Levels on Model Performance

The TuringQ dataset's difficulty levels were validated by domain experts, acknowledging the inherent subjectivity of difficulty assessments. Interestingly, our findings contradict conventional human expectations regarding question difficulty. Questions labeled as Level 3 and Level 2 achieved higher average scores (3.17) than Level 1 questions (2.95) and Axiom-level questions (2.90). Binary accuracy metrics further corroborate these findings, with the highest accuracies observed in Level 3 and Level 2 questions (Tables 4 and 6). This unexpected performance pattern across difficulty levels suggests a potential misalignment between human-perceived difficulty and the capabilities of language models in this domain.

6 Conclusion

We presented TuringQ to evaluate the reasoning capabilities of large language models (LLMs) in the theory of computation covering three difficulty levels and six main concepts, including key axioms and theoretical concepts. We evaluated various open-source LLMs and GPT-4 using Chain of Thought prompting and human expert assessment, and explored an automated LLM-Judge, demonstrating its potential to compete with human precision. Fine-tuning an LLaMA-3B model on TuringQ improved its reasoning ability. This effort provides a valuable benchmark for evaluating LLM understanding and could also be used as an educational resource. Assessing comprehension in formal languages was crucial for understanding the depth of LLMs' reasoning abilities, representing a significant step toward developing LLMs into effective problem solvers in complex domains.

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 labeled in the dataset's metadata, and no question has been extracted without mentioning 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 the specified theorems and lemmas. We believe this approach could benefit the TuringQ dataset. Subsequently, we checked and edited the model-generated answers to ensure the absence of bias, hallucinations, or errors in our work.

Regarding non-university sources, we made efforts to gather solutions from diverse, reliable sources, including computer science portals and books. As the theory of computation and theoretical computer science is an advancing and complex field, we have included answers that are accurate based on our current knowledge, particularly concerning P and NP, and open problems. We acknowledge that as our understanding progresses, some open questions in our dataset may require updates to their answers. However, to the best of our present knowledge, this dataset is up-to-date.

8 Limitations

The present study encountered several limitations that future research should address. Firstly, computational resource constraints hindered our ability to utilize larger language models with 70 billion or more parameters. Instead, we focused on smaller yet powerful models that were more feasible for our research scope. These resource constraints also impacted the fine-tuning process, limiting the Llama3-8b-ft-TuringQ model to only three epochs of fine-tuning, which may have curtailed its potential performance. Consequently, future studies should explore extended training periods and alternative fine-tuning approaches using the TuringQ dataset to fully leverage its capabilities.

Evaluating descriptive questions posed a significant challenge. While we developed two metrics for evaluating descriptive questions, incorporating more extensive human evaluation would be beneficial. Although this approach is more resource-intensive and time-consuming, it could provide valuable insights into model performance. Additionally, the development and implementation of new, more comprehensive evaluation metrics would be beneficial for assessing model capabilities.

Our dataset effectively captures the essential categories and fundamentals of theory of computation. However, 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. Further study into the broader implications and potential trade-offs of such fine-tuning on large language models is encouraged.

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.

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

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.

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.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

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. 2006. *Introduction to the Theory of Computation*, second edition. Course Technology.

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: Linear
- Number of epochs: 3 (due to computational limitations)
- 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.

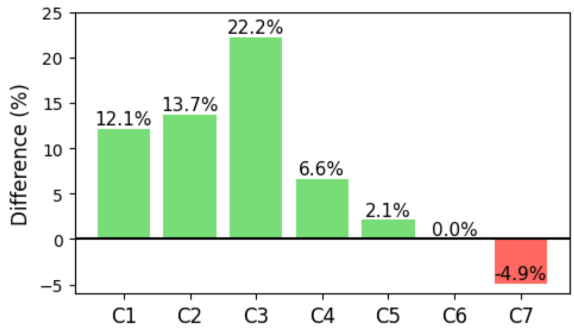


Figure 3: Bar chart showing the difference in binary accuracy (%) between Llama3-8b-ft-TuringQ and the Llama3-8b across various TuringQ categories. Categories C1 (Countability Concepts), C2 (Computability Theory), C3 (Context-Free Languages), C4 (Fundamental Concepts), and C5 (Complexity Theory) demonstrate positive accuracy gains for Llama3-8b-ft-TuringQ compared to Llama3-8b, indicating performance improvements after fine-tuning. C6 (Regular Languages) exhibits no change in accuracy and C7 (Theoretical Concepts) has a minor decrease in performance.

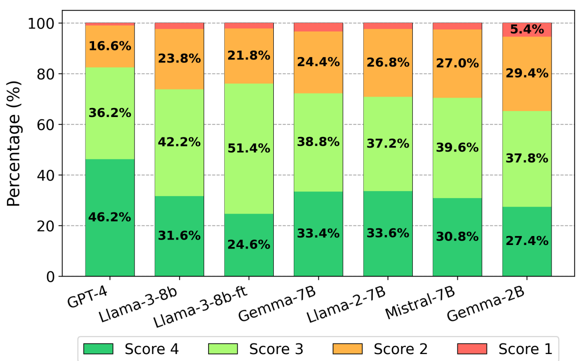


Figure 4: Score Distribution Across Models on the Test Split of the TuringQ Dataset

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

	Average	MSE	Variance	Correlation	2-Class Acc	4-Class Acc
Llama-2-7b	3.494	1.758	1.4979	0.1169	0.6800	0.3440
Llama-2-7b-CoT	3.456	1.656	1.4928	0.0478	0.7040	0.3520
Llama-3-8b	2.858	1.746	1.7301	0.1772	0.6400	0.3180
Llama-3-8b-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 2: Statistical Measures of LLM Performance as Evaluators on the TuringQ Test Set

Category	llama3-8b	Llama3-8b-ft-TuringQ	Gemma-2b	Gemma-7b	llama2-7b	Mistral-7b	GPT4
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 3: Comparative Analysis of Mean Scores Across Models by Category

Difficulty	llama3-8b	Llama3-8b-ft-TuringQ	Gemma-2b	Gemma-7b	llama2-7b	Mistral-7b	GPT4
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 4: Comparative Analysis of Mean Scores Across Models by Difficulty Level

Category	llama3-8b	Llama3-8b-ft-TuringQ	Gemma-2b	Gemma-7b	llama2-7b	Mistral-7b	GPT4
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 5: Comparative Analysis of Mean Binary Accuracy Across Models by Category

Difficulty	llama3-8b	Llama3-8b-ft-TuringQ	Gemma-2b	Gemma-7b	llama2-7b	Mistral-7b	GPT4
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 6: Comparative Analysis of Mean Binary Accuracy Across Models by Difficulty Level

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 Thoughts approach: Chain of Thoughts: 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 Thoughts for logical soundness and completeness. Use clear and concise language, avoiding unnecessary jargon.
AutoGrade-TQ	You are an automated grading system for evaluating s 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 ccs 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 7: 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 context-free grammars 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 8: Sample Instances from the TuringQ Dataset

Category	Discription
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 for regular languages 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, consisting of a set of production rules that specify 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, Church-Turing thesis, 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 9: Detailed Analysis and Interpretation of the TuringQ Dataset Categories