Evaluating Technical Capabilities of LLMs

By Anthony Devito & Ryder Swanson
CPTS 475
Washington State University

Abstract

In a rapidly evolving landscape where new large language models (LLMs) emerge at an unprecedented pace, people and businesses alike have begun implementing them into their everyday workflows. Determining their domain-specific strengths and weaknesses has become a critical endeavor. To address this problem, we propose a targeted benchmarking framework using multiple-choice assessments to evaluate and compare leading LLMs across diverse subject areas. By analyzing accuracy rates, response patterns, and error types. Our approach gives information into model capabilities, helping researchers, developers, and organizations figure out the best model for their particular use case. The results of this work not only shed light on the specialized competencies of different models but also offer a practical methodology for matching AI solutions to specific needs, ultimately enhancing the integration of generative AI into real-world applications and workflows.

1. Introduction

Currently the field of generative AI is a wild west, identifying the best model for specific tasks is a challenging but essential problem for people everywhere. With new models coming out almost daily, it's difficult to determine which models excel in particular domains, such as mathematics, science, or history. This lack of clarity can hinder effective decision-making for researchers, developers, and organizations looking to deploy AI solutions.

To address this, we propose a benchmarking approach that uses a multiple-choice test to evaluate the performance of various large language models (LLMs). By analyzing their accuracy, identifying patterns in their responses, and understanding their strengths and weaknesses, we aim to determine which models are best suited for specific tasks.

This work builds on existing AI benchmarking efforts but takes a more focused approach by targeting task-specific performance in diverse knowledge domains. Our results provide insights into the capabilities of leading models and offer practical guidance for their application. Ultimately, we hope to bring clarity to this fast-moving space, enabling better alignment between AI tools and real-world needs.

2. Problem Definition

With seemingly every company coming out with their own AI model, and training them in their own unique way. It has become increasingly difficult for users to understand each model's specific capabilities and limitations. And with so many options available, it is hard to evaluate and compare these models effectively. Each model often excels in a

certain area, for example, some are highly proficient in areas such as writing, others specialize in technical tasks like coding or doing puzzles. This array of different strengths and weaknesses makes it necessary to figure out which models are best suited for specific tasks.

This problem is becoming increasingly important as AI becomes a bigger and bigger part of products and workflows across all sorts of industries. Businesses and researchers need to select models that maximize potential and align with their unique requirements to remain efficient. By addressing this challenge, we aim to provide a systematic approach to understanding AI model performance, which in return can enable companies to make informed decisions and unlock the full potential of generative AI in their applications and workflows.

3. Models/Algorithms/Measures

Although there are over 800 models that were state of the art, highly cited, or otherwise historically notable. We only used 19 models for our research, and with such a small sample size the models we picked needed to be intentional. We went for models that were the most popular, a mix of cost, and a mix of size. This way we cover the majority of the cases that someone who is looking into AI could run into.

The list is as follows:

- openai/gpt-3.5-turbo
- openai/gpt-4o-mini
- anthropic/claude-3-haiku
- anthropic/claude-3.5-sonnet
- google/gemini-flash-1.5
- google/gemma-2-27b-it
- meta-llama/llama-3.1-70b-instruct
- meta-llama/llama-3.1-8b-instruct
- meta-llama/llama-3-70b-instruct
- x-ai/grok-beta
- cohere/command-r
- nousresearch/hermes-3-llama-3.1-405b
- mistralai/mistral-nemo
- mistralai/mistral-tiny
- microsoft/wizardlm-2-8x22b
- qwen/qwen-2.5-72b-instruct

4. Implementation/Analysis

Our study was about testing the technical capabilities of 16 leading large language models (LLMs) across various domains. The dataset we analyzed included multiple-choice questions sourced from publicly available PDFs on topics such as college algebra, high school algebra, college and high school physics, college abstract algebra, and conceptual physics. We chose these because this would be at a similar level of knowledge that a freshmen year computer science student would expected to have.

To test the models, we wrote a Python script that implemented a zero-shot benchmarking approach. Each LLM was presented with multiple-choice questions such as, “How much current does a 60 W light bulb draw if it operates at a voltage of 120 V?” with options like 0.25 A, 0.5 A, and 2:00 AM. The models were scored based on their responses, with results saved and categorized into five categories.

Based on the chosen domain of topics, our five categories were set as complex calculation, multi-step problems, random string generation, random string identification, and general performance averages. Once the testing was completed, each model was categorized into the aforementioned categories. Externally we evaluated the benchmarking accuracy against human-like reasoning and computational abilities, as well as assessing alignment with expected statistical probabilities for random tasks.

5. Results and Discussion

The results from our testing shows some pretty consistent top tier performers such as, Google Gemini Flash 1.5, and Claude 3.5 Sonnet. Our hypothesis that LLM performance would vary by task was confirmed with our findings. For example in Multi-step, and Random String Identification Google Gemini Flash 1.5 performed extremely well (Figure 1). While in other tests such as complex calculations, Claude 3.5 Sonnet excelled (Figure 4).

Something that is interesting about our results is that no model performed better than just randomly guessing on certain benchmarking, such as Random String Identification. This is a novel discovery which, as far as we know, has not been discussed in relevant papers (Hopkins and Renda, 2023; Van Koevering and Kleinberg, 2024). This might highlight a weakness in LLMs ability to handle more abstract tasks. The results show that LLM use should be tailored to specific use cases rather than a one size fits all approach.

Although we did focus on having a great diversity of tasks and benchmarks that allowed us to evaluate LLMs across multiple dimensions. The zero-shot nature didn’t allow the AI to perform at the highest level due to the raw nature of the style. It did however allow us to get unbiased assessment of the models, but each model's output could be greatly improved. Things like feeding in previously solved questions of similar topic, or asking the model to solve the problem step by step would result in much better outcomes. The hypothesis is this would get the AI to perform at a more human level.

As well as due to the nature of the assignment, we ultimately couldn’t have a massive scope. We chose multiple choice questions because they are efficient for testing, but this may oversimplify certain domains, or fail to capture the nuanced reasonings LLMs are capable of in open-ended contexts. We tried to make up for this in our testing by doing an unconventional challenge by asking it about which strings were random. This made the AI look for patterns in the answers, which pushed the limits of LLM reasoning.

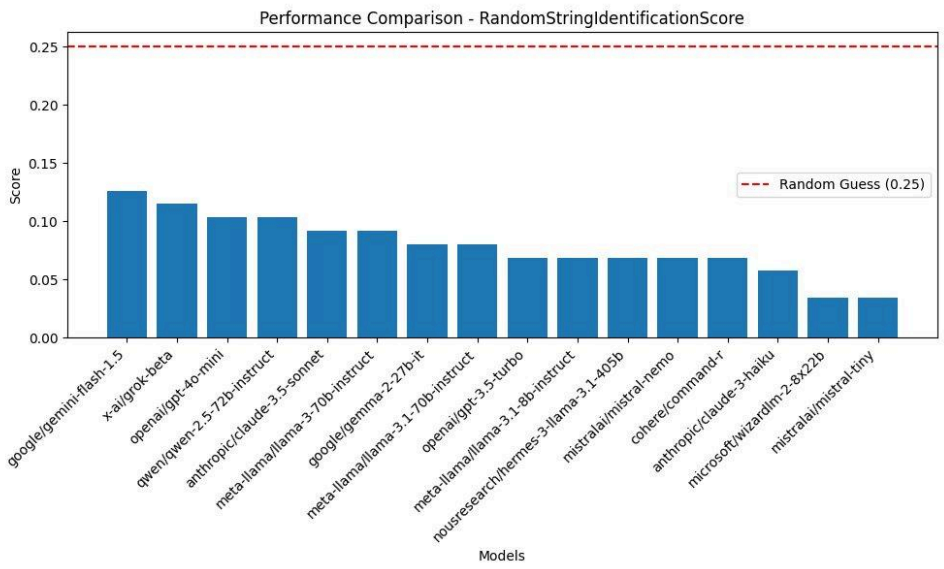

Figure 1: Performance of Random String Identification

Random String Identification, Google Gemini Flash 1.5 beats out the competition. This particular test matters in the case you are for example doing data analysis and you are looking to spot random non-patterns that can lead to uncovering trends, errors, or maybe fraud.

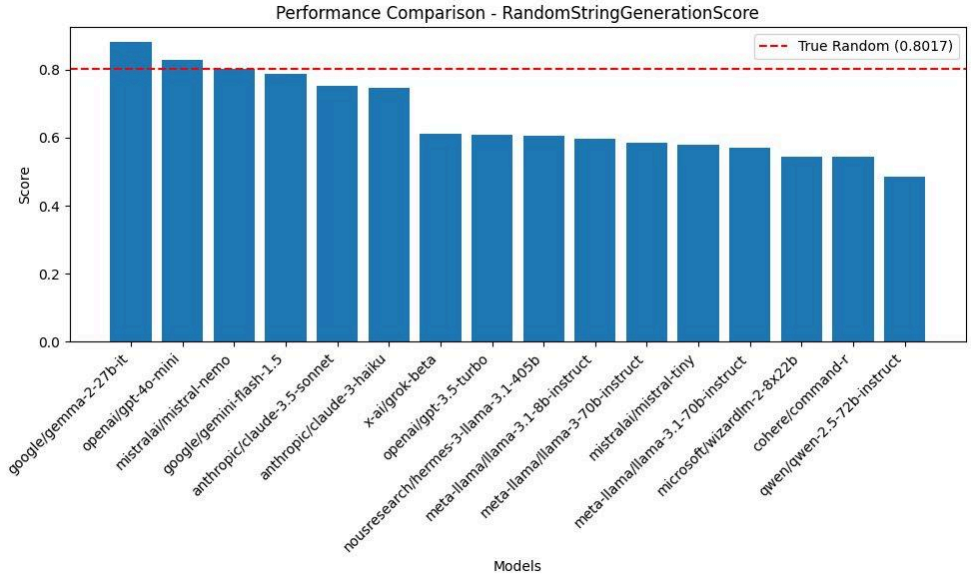

Figure 2: Performance of Random String Generation

Random String Generation, randomness is surprisingly important in technology. Secure systems like password managers, cryptographic algorithms, and token generation rely on randomness to ensure security. Random strings are also essential in generating unique identifiers for files, database entries, or sessions for the web. Models capable of generating high-quality random strings can assist in testing or even creating secure systems in everyday tools. In this case, Google Gemma 2-27b is your go to model.

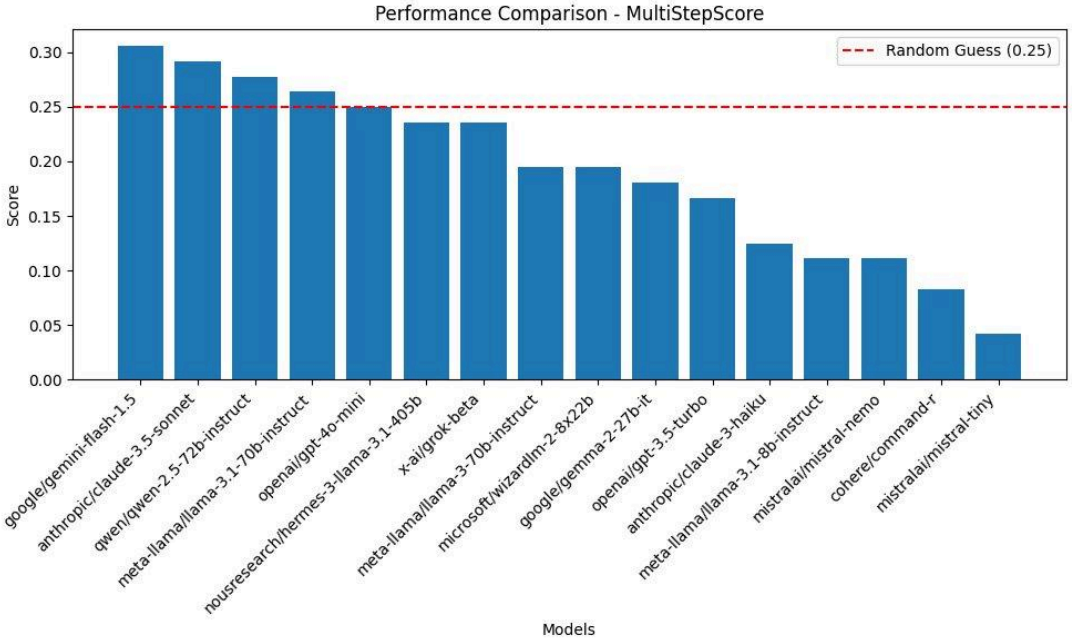

Figure 3: Performance of Multi-step problem solving

Multi-step problems show how the models solve problems with multiple steps, problems in our everyday life aren't about solving one thing at a time. This requires the model to break down a problem into smaller tasks that add up to solve the bigger solution. This would be where Google Gemini Flash 1.5 shines. This could be used for things such as project management, or customer service because these require resolving complex inquiries that require understanding the problem, retrieving relevant information, and presenting a step-by-step resolution.

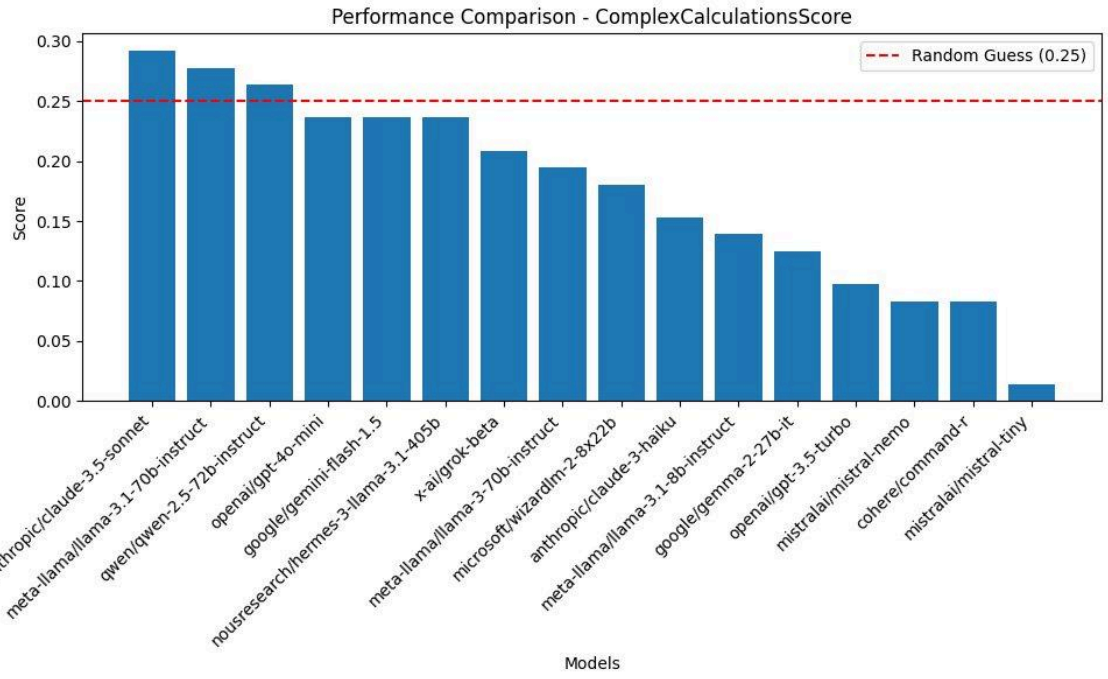

Figure 4: Performance of Complex Calculations

Complex calculations are everywhere in our daily lives, even if we don't notice them. From your smartphone to physics simulations in video games, these rely on quick and accurate math. In professional settings, engineers, scientists, and analysts often need precise calculations for things like designing structures, analyzing data, or forecasting trends. Claude 3.5 Sonnet excels in this area, and could help automate and simplify those tasks, making tools smarter and saving people more time.

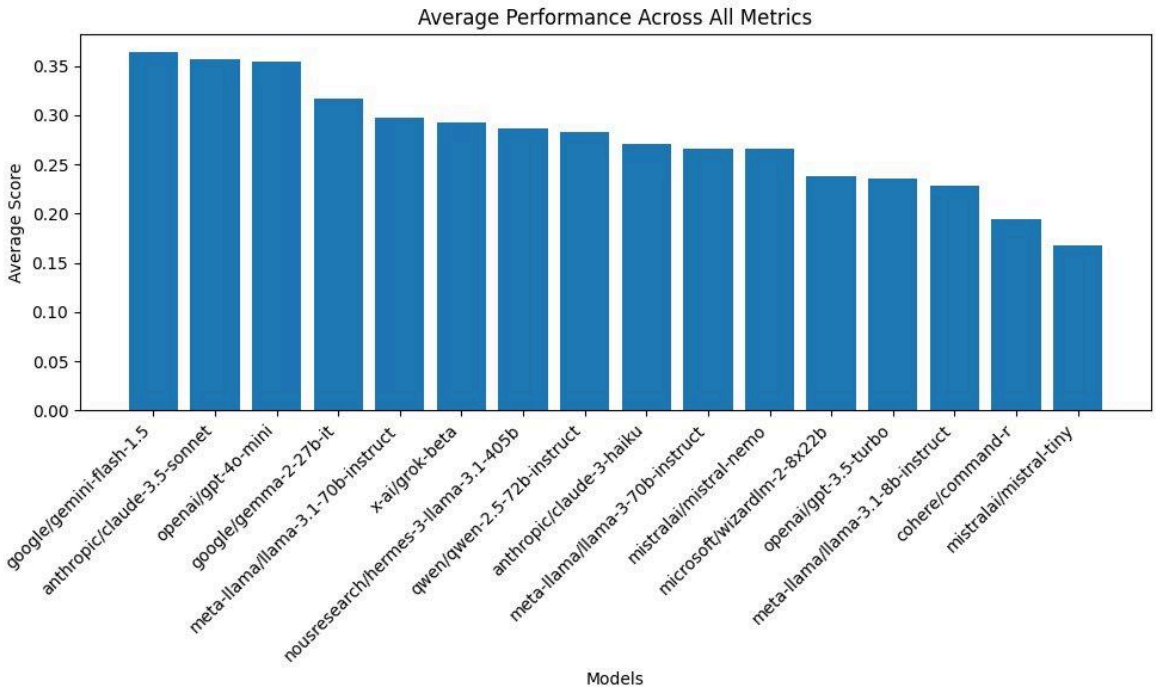

Figure 5: Average Performance

Maybe you have a problem that spans multiple domains, real world problems often do, you'd want to use Google Gemeni Flash 1.5 as it scored the highest in average performance across all metrics. This model in particular balances strong general performance with specialization that can handle a wide variety of tasks without switching tools. Things like writing reports while simultaneously generating data visualizations, or writing code but being able to communicate the code to a non-technical individual. This is actually really exciting because Google Gemeni Flash 1.5 is the smallest model on the list. It's actually 33 times cheaper than the popular GPT-4o to run but has a 700% greater context at 1 million tokens. In order for models to truly get better for practical use they need to get smaller in size or they will get too expensive to run on a day to day basis.

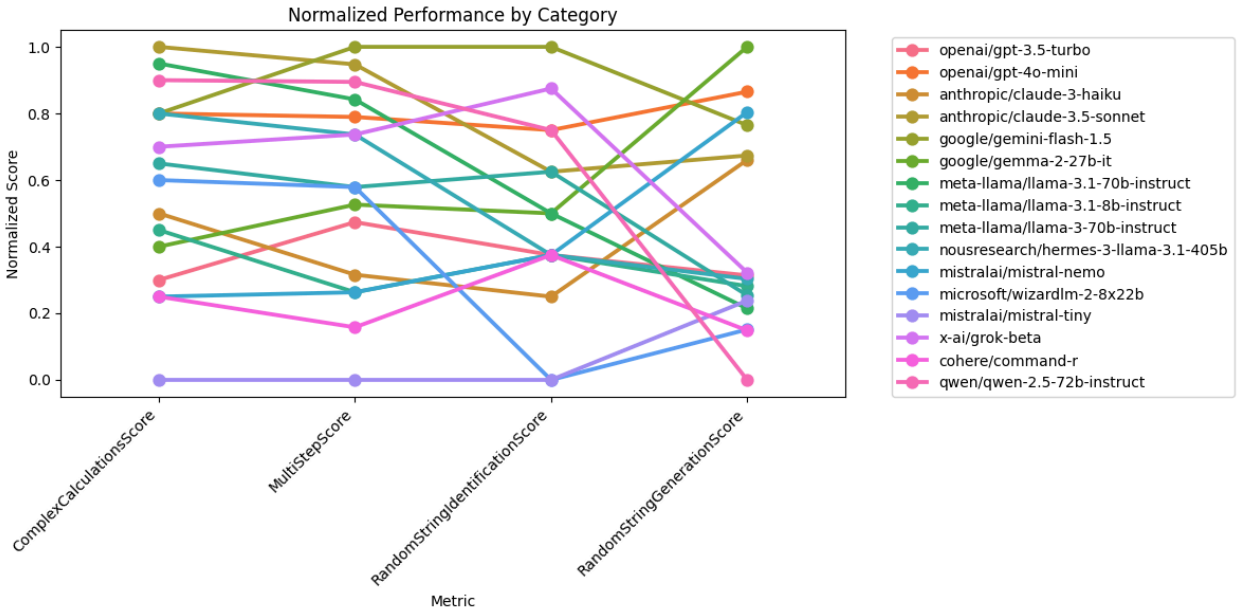

Figure 6: Normalized Performance by Category

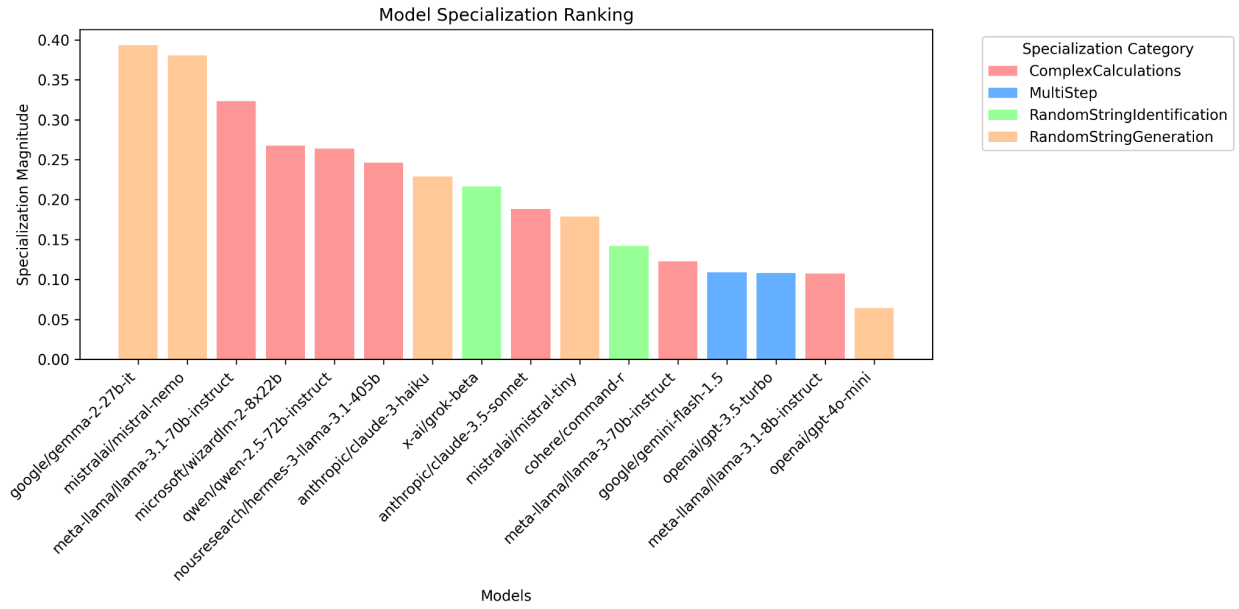

Figure 7: Model Specialization Ranking

In many high level applications you might want a model to perform well on one single task, preferring efficiency over a ‘jack of all trades’ model. This is why we measured the specialization of each model. Notably from that chart,

google/gemma-2-27b-it emerged as the most specialized model for random string generation, outperforming other models in this task. In the complex calculations category, meta-llama/llama-3.1-70b-instruct demonstrated exceptional capabilities, highlighting its strength in mathematical problem-solving. For random string identification, x-ai/grok-beta stood out as the most specialized model, showcasing its ability to distinguish between random and non-random strings. Finally, google/gemini-flash-1.5 excelled in multistep problems, demonstrating its capacity to break down complex tasks into manageable steps. These findings underscore the importance of understanding the unique strengths and weaknesses of each LLM, as different models excel in different areas.

6. Related Work

While there have been studies on the ability of large language models (LLMs) to generate random numbers, such as "Can llms generate random numbers? evaluating llm sampling in controlled domains" by Hopkins and Renda (2023) and "How Random is Random? Evaluating the Randomness and Humaness of LLMs' Coin Flips" by Van Koevering and Kleinberg (2024), our work diverges from these studies by focusing on an LLM's ability to detect random strings from non-random strings. This unique aspect of our approach sets us apart from existing research in the field.

Furthermore, we are not aware of any papers that have benchmarked LLMs to identify their domain-specific strengths and weaknesses, which is a critical component of our study. Our framework provides a systematic approach to understanding LLM capabilities, enabling the selection of the most suitable model for specific tasks. As shown in Figure 7, our results demonstrate that certain models excel in specific categories, such as XAI Grok in random string identification and Meta LLaMA 3.1 70b in complex calculations. This highlights the importance of understanding the specialized capabilities of each model, rather than relying on a one-size-fits-all approach.

By exploring the technical capabilities of leading LLMs across diverse subject areas, our study provides a comprehensive understanding of their capabilities and limitations, ultimately enabling better alignment between AI tools and real-world needs.

7. Conclusion

In this report we explored the capabilities of what the popular LLMs are across various domains. We did this through our proprietary benchmarking framework that evaluates their performance across a variety of different domains.

Our findings demonstrated that LLMs have clear domain-specific strengths and weaknesses, which confirms our hypothesis that their effectiveness varies by task. Models like Google Gemini Flash 1.5 excelled in general performance, multi-step reasoning, and random string identification, while Claude 3.5 Sonnet stood out in complex calculations. However, the inability of models to outperform random guessing in certain areas, such as random string identification, shows their current limitations in handling abstract tasks.

Our framework provided an unbiased examination through a zero-shot benchmarking methodology, but we also recognized the potential for improvement by integrating step-by-step problem-solving techniques or providing specific training on

each task. While multiple-choice questions simplified testing, they may have failed to capture the nuanced reasoning capabilities of LLMs in open-ended contexts.

Our research shows the importance of tailoring LLM selection to specific use cases, as no single model currently offers a universal solution, Google Gemini Flash 1.5 gets pretty close though. As AI continues to integrate into real-world workflows, efforts to refine benchmarking methods and expand task diversity will be critical to furthering our understanding of model capabilities (Sainz et al., 2023). Broadening our scope in the future could look like testing additional models, incorporating more open-ended tasks, or exploring fine-tuning strategies to bridge gaps in LLM reasoning and abstraction, such as combining LLMs with domain experts (Ge et al., 2024). By spending a little more time in these areas, we can figure out how to enhance the utility of LLMs and better align them with the needs of users and industries alike.

Appendix

GitHub Link - https://github.com/aanthoonyy/Evaluating-Technical-Capabilities-of-LLMs

Works Cited

Hopkins, Aspen K., and Alex Renda. "Can llms generate random numbers? evaluating llm sampling in controlled domains." Sampling and Optimization in Discrete Space (SODS) ICML 2023 Workshop, 2023.

Van Koevering, Katherine, and Jon Kleinberg. "How Random is Random? Evaluating the Randomness and Humaness of LLMs' Coin Flips." *arXiv preprint arXiv:2406.00092* (2024).

Ge, Yingqiang, et al. "Openagi: When llm meets domain experts." *Advances in Neural Information Processing Systems* 36 (2024).

Sainz, Oscar, et al. "Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark." *arXiv preprint arXiv:2310.18018* (2023).