Non-Determinism of "Deterministic" LLM System Settings in Hosted Environments

Anonymous ACL submission

Abstract

LLM (large language model) users of hosted providers commonly notice that outputs can vary for the same inputs under settings expected to be deterministic. While it is difficult to get exact statistics, recent reports on specialty news sites and discussion boards suggest that among users in all communities, the majority of LLM usage today is through cloud-based APIs. Yet the questions of how pervasive non-determinism is, and how much it affects performance results, have not to our knowledge been systematically investigated. We apply five API-based LLMs configured to be deterministic to eight diverse tasks across 10 runs. Experiments reveal accuracy variations of up to 15% across runs, with a gap of up to 70% between best possible performance and worst possible performance. No LLM consistently delivers the same outputs or accuracies, regardless of task. We speculate about the sources of non-determinism such as input buffer packing across multiple jobs. To better quantify our observations, we introduce metrics focused on quantifying determinism, TARr@N for the total agreement rate at N runs over raw output, and TARa@N for total agreement rate of parsed-out answers. Our code and data will be publicly available at https://github.com/*Anonymous*.

1 Introduction

Large Language Models (LLM) perform well on many types of Natural Language Processing (NLP) or NLP-related tasks, including question answering (Robinson and Wingate, 2023), diverse types of reasoning (Qiao et al., 2023), and code generation (Jiang et al., 2024b). Their general applicability has resulted in their widespread adoption for diverse, high-stakes societal functions, such as information gathering in medicine (Shool et al., 2025) or law (Niklaus et al., 2024), financial planning (de Zarzà i Cubero et al., 2024), or manufacturing optimization (Du et al., 2025), to name a few. In tandem with these high stakes uses, there has been increasing attention to reliability (e.g., for Out-of-Distribution behavior (Liu et al., 2024; Du et al., 2022)), alongside other aspects of LLM trustworthiness (Shridhar et al., 2024; Chen and Mueller, 2024). Uncertainty in LLM output is an aspect of performance that could either degrade or bolster trust, depending on the level of transparency. The laudable practice of testing on benchmark datasets to demonstrate progress is counterbalanced by the frequent lack of uncertainty measures. Despite known uncertainty across different training runs of a model, it has become standard to report LLM results from a single run (Hendrycks et al., 2021; Suzgun et al., 2023; Wang et al., 2024; Gema et al., 2024; Rein et al., 2023), possibly due to cost and computational time restrictions. Benchmark results reported without measures of uncertainty (e.g., confidence intervals) therefore undermines reliability. In this paper, we examine another factor that introduces variance in benchmark results: non-determinism in hosted LLMs.

Many users of LLMs gain access to models that are hosted through APIs. It is difficult to get exact statistics, but recent information from specialty news sites and discussion boards suggests that among users in all communities, the majority of LLM usage today is through cloud-based APIs.[1] Many users of LLM APIs presumably expect model output to be deterministic when temperature=0. While some users may have observed a degree of non-determinism in this setting, there is little if any quantification of this variance. Throughout the paper, we refer to this behavior of output variance despite zero temperature as instability or non-determinism. We demonstrate an alarming

[1] E.g.: https://www.prnewswire.com/news-releases/study-finds-72-of-enterprises-plan-to-ramp-spending-on-genai-in-2025-302484025.html?utm_source=chatgpt.com; https://konghq.com/resources/reports/ai-and-api-adoption-challenges.

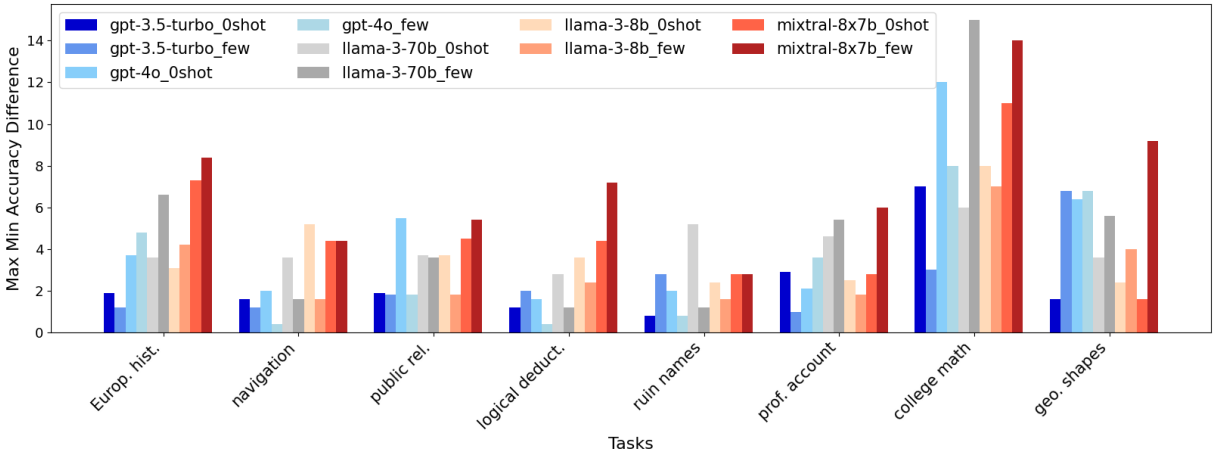

Figure 1: Percentage difference between maximum and minimum accuracy in 10 runs per model, for 5 models on 8 tasks with zero-shot and few-shot settings.

degree of variation across equivalent input runs with a varied collection of high performing API-based LLMs[2] under presumed deterministic settings. Our findings of up to 15% differences in accuracy across runs demonstrate there is far too much uncertainty in a realm where robust engineering is the expectation.

To quantify the problem of instability when temperature=0, we measure it in three LLM families (GPT, Llama, and Mixtral) on diverse tasks from two common benchmarks: Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021) and BIG-Bench Hard (BBH) (Suzgun et al., 2023). Figure 1 depicts differences between maximimum and minimum accuracies in multiple runs, showing that the degree of instability changes across model families, model sizes, tasks and settings. Therefore, performance instability can doubtless impact the ranking performance of systems. Our specific contributions include:

- Quantification of LLM system instability over 8 tasks randomly selected from two common benchmarks: BBH and MMLU.
- Two metrics, TARr@N (total agreement rate for raw data across N runs) and TARa@N (total agreement rate for parsed answer across N runs) for LLM system instability to capture the variability in answer accuracy and in the output word spans.
- Comparison across settings, including zero-shot and few-shot (3 for BBH, 5 for MMLU as in the standard settings).
- Correlation analyses of instability with accuracy, input length, and output length.
- Experiments on locally run LLMs that demonstrate the desired stability.
- Data from runs and source code.[3]

The remainder of the paper is structured into the following sections: Introduction, Related Work, Datasets, Methods, Experiments, Results, Discussion, and Conclusion.

2 Related Work

To the best of our knowledge, no work systematically investigates LLM instability given the same inputs and configurations (zero-shot and few-shot) with maximally deterministic hyperparameters for hosted LLMs. However, there is relevant work on both robustness of evaluation results in general, and on instability of hosted LLMs. Biderman et al. (2024) introduce a standard evaluation toolkit for LLMs and suggest best practices for reproducibility, but do not discuss instability. Works on the robustness of machine learning (ML) models with trivial changes to the input include (Sehwag et al., 2019; Freiesleben and Grote, 2023; Hancox-Li, 2020; Rauber et al., 2017). The (Song et al., 2024) paper, which mentions instability, analyzes the effect of temperature, sampling strategy, repetition penalty, and alignment algorithms on performance evaluation. Findings include that LLMs have some variance in the output that should be taken into account in evaluation benchmarks. However, they use a temperature of 1, thereby introducing the variability that our study seeks to minimize. Ouyang et al.

[2]API-based LLMs refer to the usage of LLMs through APIs such as OpenAI API or Together API.

[3]Download URL will be provided if paper is accepted.

Task	Description	Size	Options
BBH: navigation	does path end at start	250	2
BBH: ruin names	humorous edit of a band or movie title	250	4
BBH: geometric shapes	shape given SVG format	250	10
BBH: logical deduct. 3 objects	order of 3 objects given constraints	250	3
MMLU: h. s. Europ. hist.	*identical*	165	4
MMLU: college math	*identical*	100	4
MMLU: prof. accounting	*identical*	282	4
MMLU: public rel.	media theory, crisis mgmt., etc.	110	4

Table 1: Eight tasks from BBH and MMLU with brief descriptions, and numbers of examples and answer options.

(2025) present an instability analysis of a single model, ChatGPT, with varying temperatures on the one task of code generation. Lastly, Holtzman et al. (2020) mention freedom in text generation which might lead to different outputs for the same inputs, but they do not talk about the parameters that affect this behaviour.

3 Datasets

To ensure that our investigation of instability includes diverse NLP tasks, we selected tasks from two widely used multiple-choice benchmarks: Beyond the Imitation Game Benchmark Hard (BBH) (Suzgun et al., 2023), with 27 diverse tasks from mathematics, commonsense reasoning and other domains; Measuring Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021), with 57 tasks across disciplines including the humanities, social sciences, and STEM areas. To balance diversity against computational resources, we randomly selected four subtasks from each benchmark. Table 1 lists the tasks we selected, number of examples, and number of multiple-choice options.

4 Methods

The subsections here discuss the LLM temperature parameter, the models we chose, and our metrics.

4.1 Controlling LLM Determinism

The temperature hyperparameter controls the degree of determinism. Equation 1 shows the probability of word i where T is temperature $\in [0, 1]$ and y_i is the LLM logit:

$$\frac{e^{\frac{y_i}{T}}}{\sum_{j=1}^{N} e^{\frac{y_j}{T}}} \quad (1)$$

Theoretically, when $T = 0$, the LLM should produce the same output given the same prompt, and T can be raised to diversify outputs. As shown in Figure 1, utilization of LLMs through APIs leads to variable output at $T = 0$.

4.2 Models

We chose five top performing models from different families and with varying sizes: GPT-3.5 Turbo (Brown et al., 2020), GPT-4o (OpenAI et al., 2024), Llama-3-70B-Instruct (Meta, 2024), Llama-3-8B-Instruct (Meta, 2024), and Mixtral-8x7B-Instruct (Jiang et al., 2024a).

4.3 Metrics

To quantify instability, we report three metrics based on accuracy that capture accuracy extremes within a set of runs in a given experimental condition (model × dataset; see below). We also report median accuracy; we do not report means and standard deviations because the distributions in runs for a given condition are not normal (see below). Additionally, we present two key metrics that are variants of Total Agreement Rate@N (TAR@N): the percentage of test set questions across N runs where generated answers are all identical, *regardless of whether the answer was correct.* This gives six measures per condition:

1. TARr@N (TAR@N for the **raw** model response) LLM responses are string equivalent.
2. TARa@N (TAR@N for the parsed **answer**) The parsed answers are the same, e.g., "The answer is a)" is the same as "a) is the answer."
3. The best possible accuracy over N runs (BestAcc), which is the maximum possible accuracy that could be extracted from N runs. For each question, if there is a run in which the answer is correct, that question is marked as correctly answered.
4. The worst possible accuracy over N runs (WorstAcc), which is the minimum possible

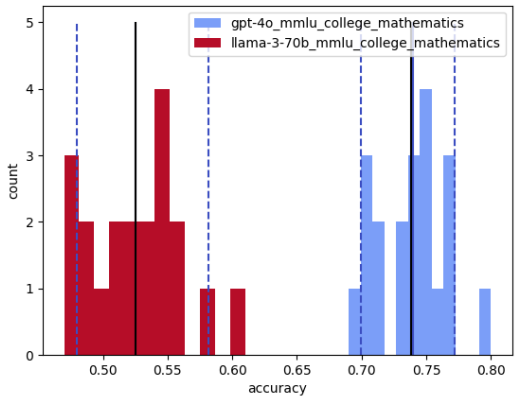

Figure 2: Accuracy over 20 identical runs on college math, temperature=0, top-p=1. Median in blue, mean in black with dashed 5% and 95% quantiles.

accuracy that could be extracted from N runs. For each question, if there is a run in which the answer is incorrect, that question is marked as incorrectly answered.

5. Maximum-minimum accuracy difference across N runs (max-min-diff). Note that because it represents the largest gap in N runs, it is not the same as the difference between BestAcc and WorstAcc.
6. Median accuracy over N runs.

The TARr@N score is very strict, since any character variation will result in a disagreement. Thus in principle, it is possible for the same set of runs to have 100% TARa@N and 0% TARr@N.

To examine the distributional behavior of accuracy scores, we did 20 few-shot runs of GPT-4o and Llama-3-70b on college math, two of the more unstable conditions. The results in Figure 2 clearly show non-normal distributions, with mean and median values far from the mode. A Kolmogorov-Smirnov normality test (Massey Jr, 1951) rejected the normal hypothesis with a p-value $< 10^{-9}$.

4.4 Correlation Analyses

In addition to reporting measures of instability, we also investigate how independent the measures are using Spearman's rank correlation test. As part of this analysis, we include median input length and median output length as possible correlates.

5 Experimental Conditions

For our investigation of instability, we perform experiments on models without fine-tuning in both zero-shot and few-shot prompting (without Chain-of-Thought (CoT) (Wei et al., 2022)). Regarding the number of examples for few-shot, we use the standard settings of 3-shot for BBH tasks, and 5-shot for MMLU tasks.

All runs use the same compute infrastructure, inputs, and configurations. However, we should note that we do not have any control of the compute infrastructure on the API-side. We set temperature at 0, top-p at 1, and we fix the seed.

Our eight datasets, five base models and two settings (zero/few-shot) yield eighty conditions. For each condition, we performed ten runs.

6 Results

Here, we report our two types of results. Overall results on the instability measures show that all five models have a high degree of instability with respect to both the raw output and the task accuracies. The correlation analyses show that instability increases with output length, and that lower instability correlates with median accuracy for the few-shot setting.

6.1 Instability Results

Figure 1 summarizes the extremes observed across our eight datasets for the five models in zero-shot and few-shot settings. The y-axis is the percentage difference between the minimum and maximum accuracies (max-min-diff) in ten runs for each condition. Notably, there are 5-15% differences on some tasks.

The top of Table 2 reports BestAcc, median accuracy and WorstAcc in the few-shot conditions for our five models (zero-shot results show a similar degree of non-determinism, with varying consistency across conditions, see Table 3 in Appendix A.2). The lower half of the table reports TARa@10 and TARr@10. When there is a gap between BestAcc and WorstAcc > 10, there is often very low TARr@10 (e.g., GPT3.5 on geometric shapes, logical deduction, ruin names; GPT4o on public relations, European history professional accounting, college math). Notably, TARr@10 is typically fairly low, and there is a lot of variation across models and datasets. Unsuprisingly, TARa@10 can be much higher than TARr@10, following from the fact that TARr@N is a very strict metric (see above).

Figure 3 shows the TARr@10 for each task and model in a few-shot setting (for zero-shot

Task	gpt3.5	gpt4o	llama8b	llama70b	mixtral8-7b
Accuracy Results					
navigation	96.8, 95.6, 93.2	98.8, 98.8, 98.4	82.0, 80.2, 78.0	95.2, 94.6, 93.6	84.4, 79.0, 71.6
geo. shapes	72.4, 59.6, 46.8	82.4, 68.4, 53.6	49.2, 40.6, 32.8	67.2, 57.0, 47.2	54.4, 27.8, 08.8
logical deduct.	88.8, 81.6, 75.2	100., 100., 99.6	95.6, 90.2, 81.2	98.0, 96.4, 95.2	87.6, 75.0, 64.0
public rel.	75.5, 69.1, 65.5	80.0, 76.4, 73.6	63.6, 61.8, 61.8	67.3, 60.5, 53.6	58.2, 48.2, 36.4
Europ. hist.	83.6, 81.2, 78.2	89.1, 81.5, 72.1	74.5, 67.0, 59.4	61.8, 50.3, 41.2	65.5, 51.5, 35.8
ruin names	72.0, 58.0, 44.8	93.2, 90.8, 88.4	68.4, 66.8, 64.4	89.2, 87.2, 84.4	78.8, 67.6, 55.6
prof. account	52.5, 50.9, 48.9	89.0, 74.5, 57.8	48.2, 45.4, 44.0	78.0, 67.2, 55.3	67.0, 39.0, 13.1
college math	39.0, 38.0, 34.0	88.0, 69.0, 44.0	50.0, 22.5, 04.0	85.0, 54.5, 22.0	75.0, 31.5, 03.0
TAR Results					
navigation	96.4, 46.0	99.6, 46.0	96.0, 86.0	98.4, 64.0	84.8, 50.0
geo. shapes	62.8, 25.2	63.2, 00.0	58.8, 27.6	66.4, 18.0	12.0, 02.4
logical deduct.	84.4, 34.8	99.6, 36.8	85.2, 50.0	97.2, 49.6	74.8, 16.4
public rel.	87.3, 82.7	92.7, 37.3	96.4, 73.6	81.8, 17.3	62.7, 10.9
Europ. hist.	94.5, 70.9	81.2, 09.1	82.4, 07.3	73.3, 22.4	55.2, 23.6
ruin names	66.0, 05.6	95.2, 00.0	88.4, 47.6	94.4, 10.8	70.4, 24.8
prof. account	91.1, 76.6	66.7, 04.6	89.0, 52.1	69.5, 00.0	23.4, 00.7
college math	89.0, 76.0	50.0, 00.0	22.0, 00.0	25.0, 00.0	07.0, 00.0

Table 2: BestAcc, Median Accuracy, WorstAcc on top; TARa@10, TARr@10 on bottom, for the few-shot conditions (3 for BBH, 5 for MMLU, see section 5). Results are in terms of percentages.

scores, see Figure 12 in Appendix A.2). GPT-3.5 Turbo has lower TARr@10 (less instability) than other models, and Llama-3-70B often has very low TARr@10.

Figure 4 shows TARa@10 for each condition in a few-shot setting (see Figure 11 in Appendix A.2 for zero-shot). While the TARa@10 results show less instability than TARr@10, they are still far from perfect and show task-specific results. The high scores for the navigation task indicate that leaderboards on this task can be expected to be more reliable. On the other hand, the more scattered results for the college math and professional accounting tasks indicate that results reported on these tasks are not as robust.

6.2 Correlation Analyses

We perform a Spearman rank correlation analysis on all pairs of the following metrics: TARa@10, TARr@10, max-min-diff, median accuracy, median input length, and median output length. Heat map results are shown in Figure 5 for the few-shot and zero-shot prompted models. Here we define accuracy as the median accuracy over the 10 runs with the same model and dataset setup. Input length and output length are median word counts split by space, calculated over the input and output of each LLM experiment setup. We split the words by space instead of using a particular tokenizer.

The results show a strong to moderate negative correlation between the output length and TARa@10, as well as between the output length and TARr@10 in few-shot/zero-shot settings. Note this is also consistent with the positive correlation of output length with max-min-diff. These correlations mean that as an LLM's output length increases, the instability of the output increases, resulting in more diverse natural language responses as well as in the actual multiple choice answer prediction. The strong negative correlation between LLM output length and instability could motivate those using LLMs in hosted environments to restrict the max generation tokens to control the instability. We also see a strong positive correlation between median accuracy and TARa@10 in the few-shot setting. This indicates that when the LLM is more accurate it becomes more deterministic for multiple choice selections. Additionally, in the few-shot setting, there is a moderate negative correlation between the output length and median accuracy, which indicates that restricting max generation tokens may improve both determinisim and accuracy. This is in parallel with the findings in (Zhang et al., 2024).

In addition to general correlations, we also look at correlation maps per model to see how general findings apply to each.[4] We find that all models are

[4]These correlation map figures are in Appendix A.1.

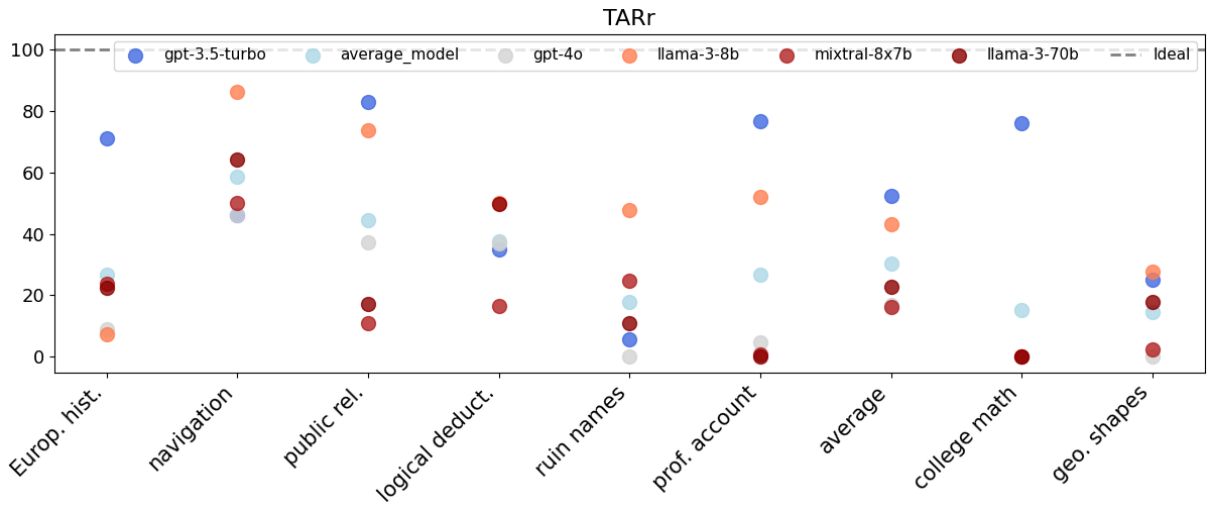

Figure 3: TARr@10 for each model in the few-shot setting. Dataset colors have been chosen to distinguish them by relatively challenging (increasingly dark red hues) versus relatively easy (increasingly dark blue hues).

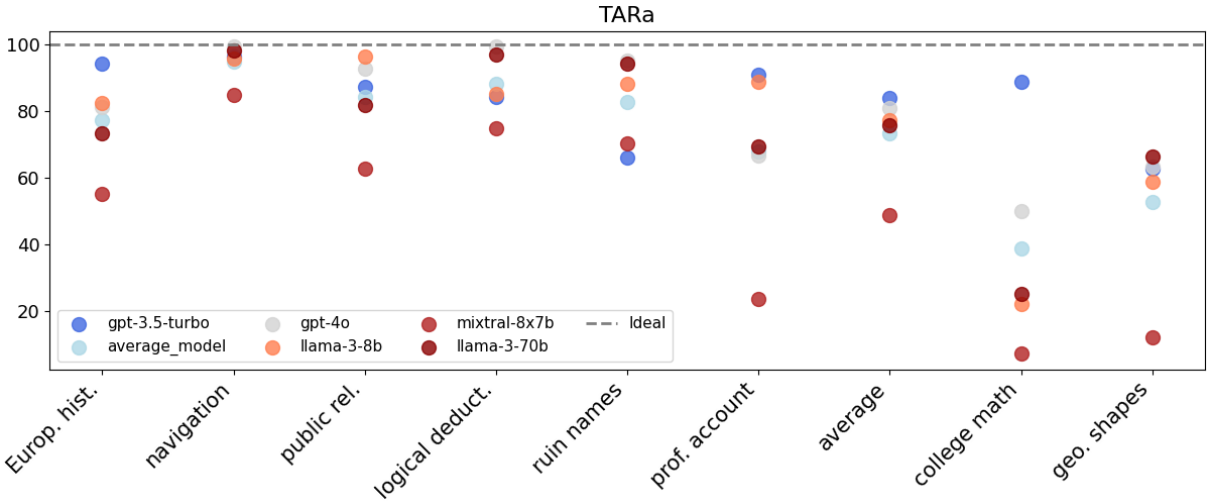

Figure 4: TARa@10 for each task in the few-shot setting. Models colors have been chosen to distinguish them by relatively low performing (increasingly dark red hues) versus relatively high performing (increasingly dark blue hues).

more stable when they generate shorter responses. Notably, Mixtral and Llama-3 models are more stable when they are more accurate in the few-shot setting, but the effect varies in the zero-shot setting. Last but not least, in the few-shot setting GPT-3.5 is more stable when the input is longer, but this effect shows up less in the zero-shot setting.

7 Discussion

Theoretically, at 0 temperature the LLMs should be deterministic given the same input, with values of 100% for TARa@10 and TAR@10, the same values for BestAcc and WorstAcc, and 0% difference in the minimum and maximum values across all tasks. Our results show that zero temperature is far from deterministic for API usage of LLMs. The TARr@10 scores show that hosted LLMs are not stable at the string level in the $T = 0$ setting, while the TARa@10 scores show they are far more deterministic at the parsed answer level. String variation does not affect a human reader much because we can extract the same answer even if the output format is different, but a downstream system that needs to parse the LLM response can be affected significantly when the format or pattern is different. This should be taken into account when using hosted LLMs.

TARa@10 values are much more consistent than TARr@10, yet still lead to high instablity of up to 15%, as shown in Figure 1. One caveat is that

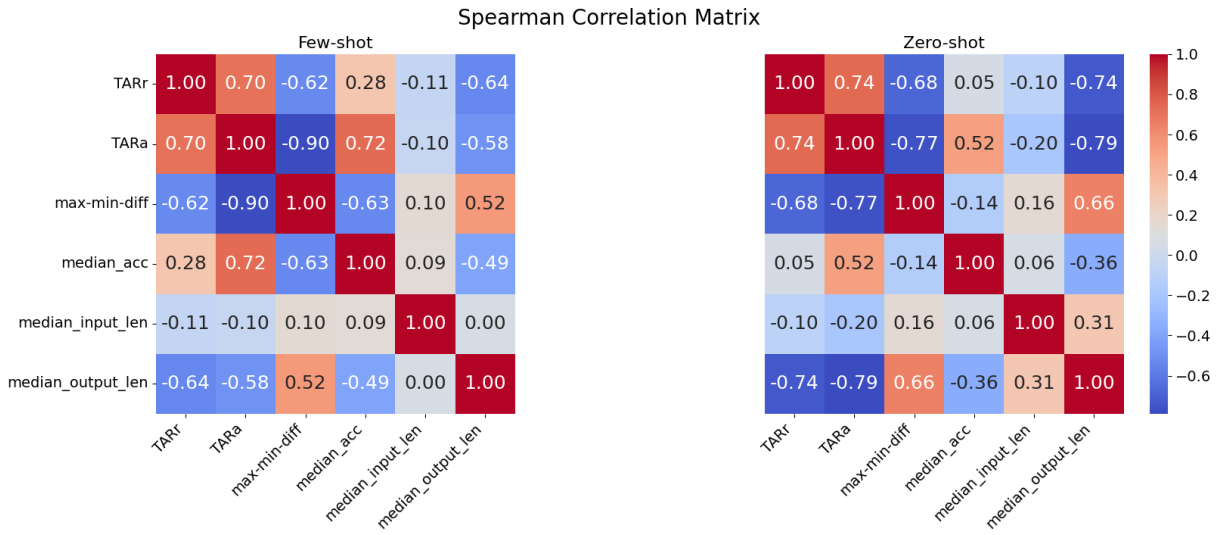

Figure 5: Spearman correlation matrices for pairs of metrics in the few-shot (left) and zero-shot (right) settings.

our answer extraction system has many hard-coded parts, which reduces the generality of the system. Therefore, we have no guarantee that raw outputs will lead to the exactly the same results for our various accuracy metrics, if the experiments are repeated.

Theoretically, the maximum-minimum accuracy difference (max-min-diff) should be 0%. All LLMs here demonstrate considerable variation on this metric. Mixtral-8x7b on college math is 72% (75% - 3%) for a particularly bad example on suggesting a truly random element in the generative process driving the minimum value to 0%. This instability lowers confidence in the reliability of reporting only a single number in LLM benchmarks. We encourage reporting maximum-minimum scores across runs to have a more robust comparison of LLM systems.

7.1 Implications for Practical Engineering

Although the use of multiple GPUs introduces some randomness (Nvidia, 2024; Dror et al., 2019), it can be eliminated by setting random seeds, so that AI models are deterministic given the same input. In that case, performance errors could be attributed to the model's generalization capability (e.g., under-/over-fitting). However, engineering optimizations to run LLMs faster, such as continuous batching, chunk prefilling, or prefix caching, might lead to non-deterministic behavior. Since many of the models are close-sourced (GPT-3-5, GPT-4o), and all are hosted behind APIs we don't control, we can only speculate about the reason for this behavior. In order to support this line of reasoning, we ran Llama3-8b on our local GPUs without any optimizations, yielding deterministic results. This indicates that the models and GPUs themselves are not the source of non-determinism.

Additionally, we fine-tuned GPT-3.5 using two-fold cross validation. Although the results indicate that fine-tuning helps reduce instability, we hypothesize that a fine-tuned model cannot be shared across users and as such, our tasks were the only ones being run. Hence, fine-tuning itself may not be the only reason for reduced instability.

Non-deterministic AI brings new challenges to developers, especially in commercial applications:

- The usage of unit tests for AI functions is limited because of non-determinism.
- Low stability might also increase the potential for inexplicable errors that are very different from human mistakes such as responding as "none of the above" when the task is a multiple choice selection.
- Instability of the format of the outputs can result in downstream parser failures.
- One of the most important effects is in system complexity that has to handle gracefully "usually correct but this time wrong" results. Zipfian distributions are commonly seen in applied AI systems where the frequency of an input/category is inversely related to its rank in count sorted order $frequency \propto 1/rank$). Testing tends to concentrate on the frequent events, potentially resulting in user confidence

that the resulting system is stable for the common inputs. However, the lack of stability shown here undermines the entire foundation of this confidence, especially if mistakes are costly.

8 Conclusion

We have made a systematic analysis of the determinism of hosted LLMs with the temperature hyperparameter value that should maximize it. Our results show that such systems can be highly nondeterministic with $T = 0$. Furthermore, we find that these LLMs rarely produce the same response ten times given the same input; the parsed answer is often more stable. Note that the observation that instability results are not normally distributed makes it more difficult to measure the resulting uncertainty. Lastly, instability is highly variable across tasks for the same model, and across models for the same task.

Other questions about instability remain to be explored. For instance, how can we reduce the instability of hosted LLM systems during training or inference time (e.g., adding a meta prompt to indicate the model is only allowed to answer with a single letter)? Second, how can the instability of hosted LLM systems be taken into account in business products? Third, how should we communicate with decision-makers about instability? Last but not least, more analysis could be done to see if there is any correlation between the stability and specific types of errors, such as false positives and false negatives.

Limitations

Our experiments are limited to 8 datasets and multiple choice questions. Further, we only experimented with 5 LLM systems. However, given the overal pattern we have observed, we believe that the findings likely apply to other datasets and LLMs.

References

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, and 1 others. 2024. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiuhai Chen and Jonas Mueller. 2024. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5186–5200, Bangkok, Thailand. Association for Computational Linguistics.

Irene de Zarzà i Cubero, Joaquim de Curtò i Díaz, Gemma Roig, and Carlos T Calafate. 2024. Optimized financial planning: Integrating individual and cooperative budgeting models with llm recommendations. *AI*, 5(1):91–114.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. Association for Computational Linguistics.

Kaze Du, Bo Yang, Keqiang Xie, Nan Dong, Zhengping Zhang, Shilong Wang, and Fan Mo. 2025. Llm-manuf: An integrated framework of fine-tuning large language models for intelligent decision-making in manufacturing. *Advanced Engineering Informatics*, 65:103263.

Xuefeng Du, Zhaoning Wang, Mu Cai, and Sharon Li. 2022. Towards unknown-aware learning with virtual outlier synthesis. In *International Conference on Learning Representations (ICLR)*.

Timo Freiesleben and Thomas Grote. 2023. Beyond generalization: a theory of robustness in machine learning. *Synthese*, 202(4):109.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, and 1 others. 2024. Are we done with MMLU? *arXiv preprint arXiv:2406.04127*.

Leif Hancox-Li. 2020. Robustness in machine learning explanations: does it matter? In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 640–647, New York, NY, USA. Association for Computing Machinery.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024a. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. 2024b. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515*.

Bo Liu, Li-Ming Zhan, Zexin Lu, Yujie Feng, Lei Xue, and Xiao-Ming Wu. 2024. How good are LLMs at out-of-distribution detection? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8211–8222, Torino, Italia. ELRA and ICCL.

Frank J Massey Jr. 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Meta. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date. https://ai.meta.com/blog/meta-llama-3.

Joel Niklaus, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis, and Daniel Ho. 2024. MultiLegalPile: A 689GB multilingual legal corpus. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15077–15094, Bangkok, Thailand. Association for Computational Linguistics.

Nvidia. 2024. Floating point and ieee 754 compliance for nvidia gpus. https://docs.nvidia.com/cuda/floating-point/index.html.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Shuyin Ouyang, Jie M. Zhang, Mark Harman, and Meng Wang. 2025. An empirical study of the non-determinism of ChatGPT in code generation. *ACM Transactions on Software Engineering and Methodology*, 34(2).

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5368–5393, Toronto, Canada. Association for Computational Linguistics.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. 2017. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *Preprint*, arXiv:2311.12022.

Joshua Robinson and David Wingate. 2023. Leveraging large language models for multiple choice question answering. In *The Eleventh International Conference on Learning Representations*.

Vikash Sehwag, Arjun Nitin Bhagoji, Liwei Song, Chawin Sitawarin, Daniel Cullina, Mung Chiang, and Prateek Mittal. 2019. Analyzing the robustness of open-world machine learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, AISec'19, page 105–116, New York, NY, USA. Association for Computing Machinery.

Sina Shool, Sara Adimi, Reza Saboori Amleshi, Ehsan Bitaraf, Reza Golpira, and Mahmood Tara. 2025. A systematic review of large language model (llm) evaluations in clinical medicine. *BMC Medical Informatics and Decision Making*, 25(1):117.

Kumar Shridhar, Koustuv Sinha, Andrew Cohen, Tianlu Wang, Ping Yu, Ramakanth Pasunuru, Mrinmaya Sachan, Jason Weston, and Asli Celikyilmaz. 2024. The ART of LLM refinement: Ask, refine, and trust. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5872–5883, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. 2024. The good, the bad, and the greedy: Evaluation of LLMs should not ignore non-determinism. *arXiv preprint arXiv:2407.10457*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances*

in neural information processing systems, 35:24824–24837.

Yusen Zhang, Sarkar Snigdha Sarathi Das, and Rui Zhang. 2024. Verbosity ≠ veracity: Demystify verbosity compensation behavior of large language models. *Preprint*, arXiv:2411.07858.

A Appendix

A.1 Correlation Matrices Per Model

A.2 Zero-shot Results

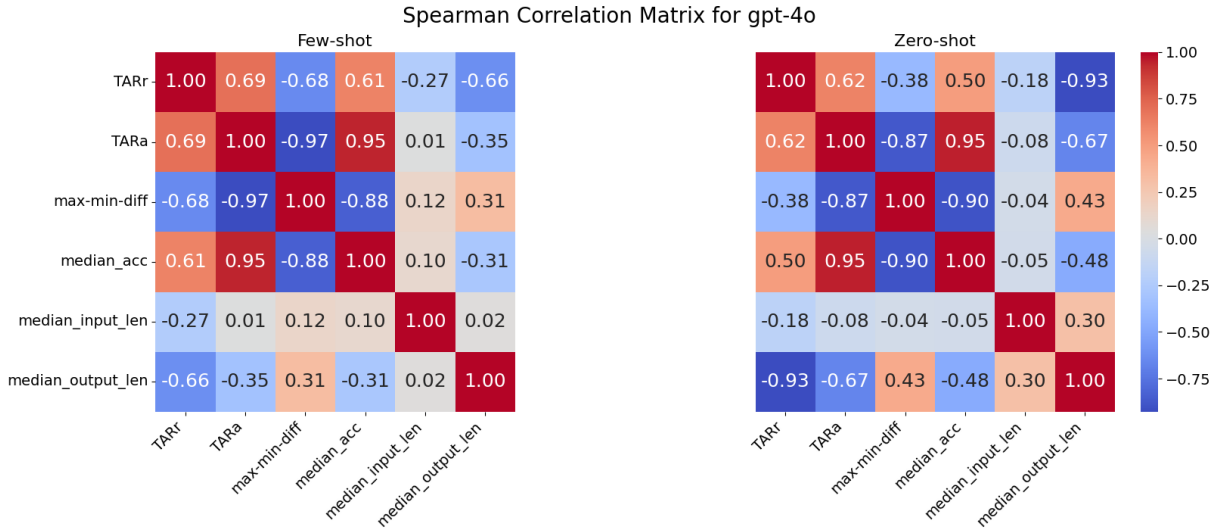

Figure 6: Spearman correlation matrices for GPT-4o for pairs of metrics in the few-shot (left) and zero-shot settings (right).

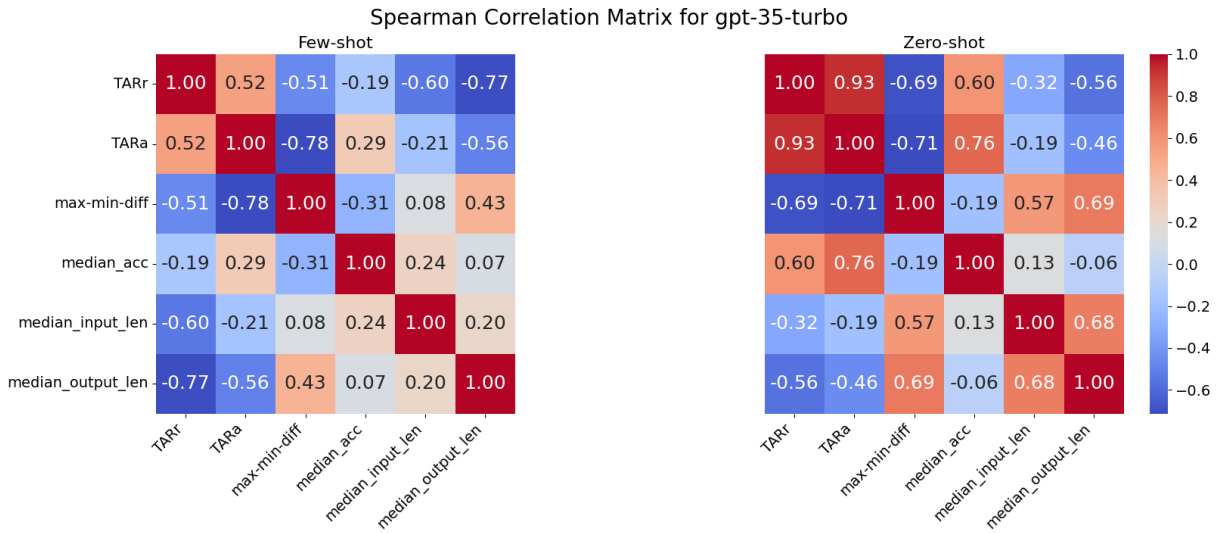

Figure 7: Spearman correlation matrix for GPT-3.5-turbo between metrics in few-shot setting (on the left) and zero-shot setting (on the right).

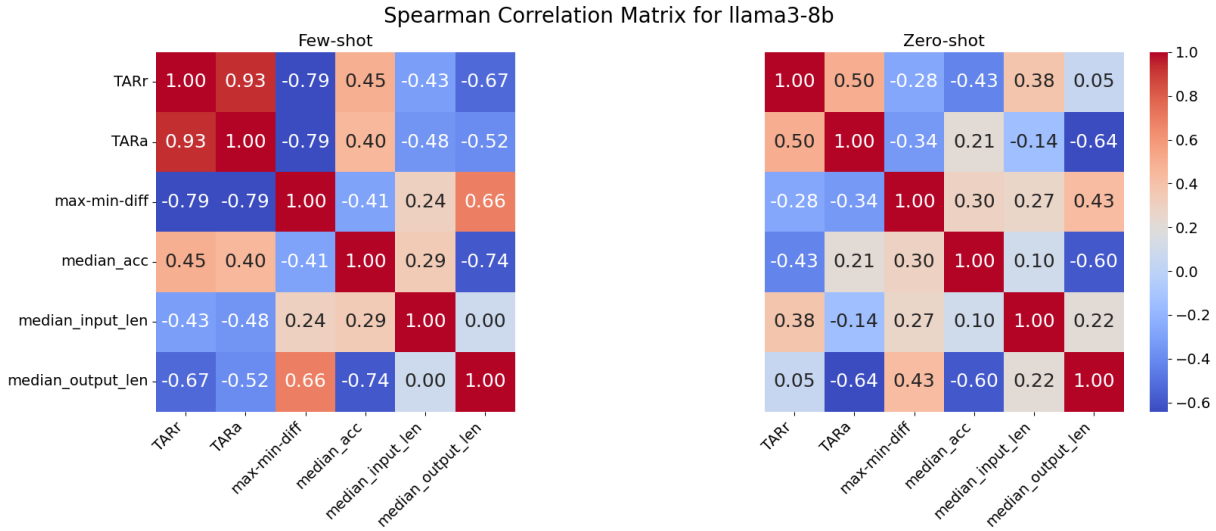

Figure 8: Spearman correlation matrix for Llama-8b between metrics in few-shot setting (on the left) and zero-shot setting (on the right).

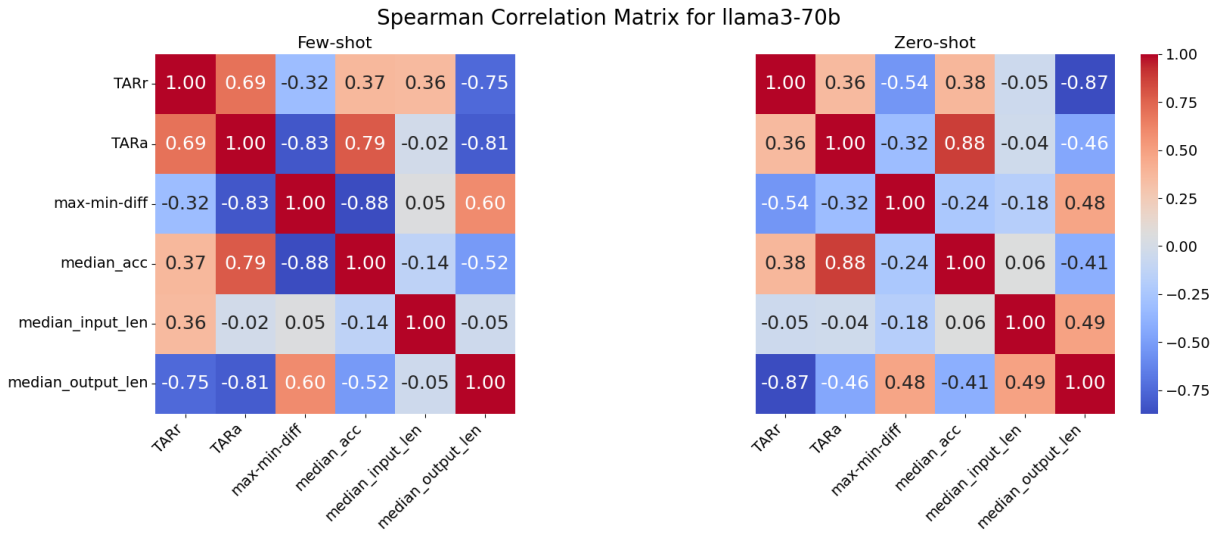

Figure 9: Spearman correlation matrix for Llama-70b between metrics in few-shot setting (on the left) and zero-shot setting (on the right).

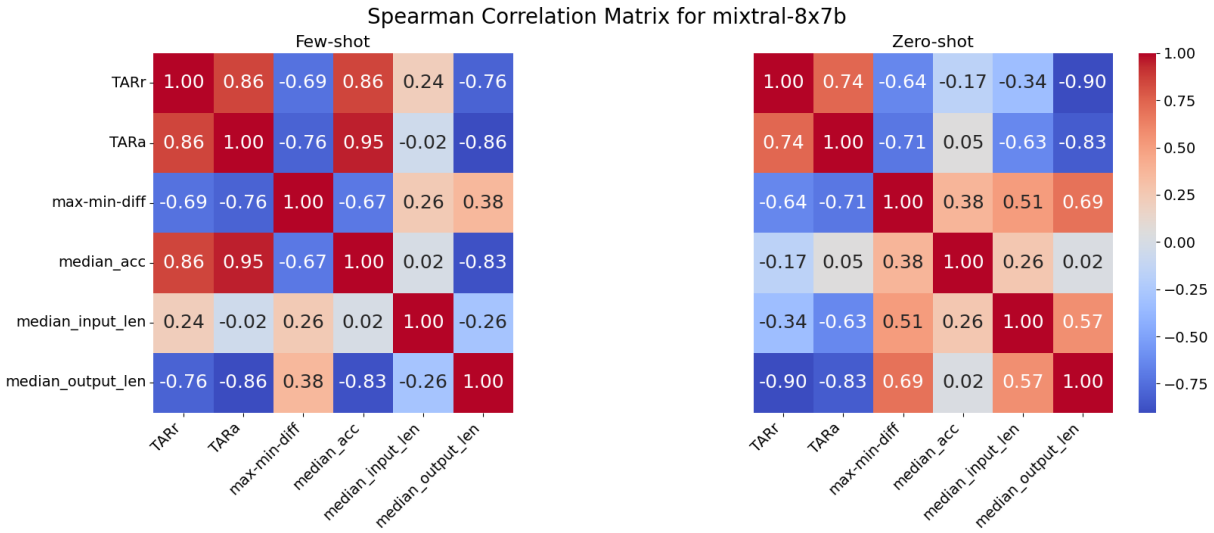

Figure 10: Spearman correlation matrix for Mixtral-8x7b between metrics in few-shot setting (on the left) and zero-shot setting (on the right).

Task	gpt3.5	gpt4o	llama8b	llama70b	mixtral8-7b
Accuracy Results					
navigation	67.2, 64.8, 61.6	94.8, 92.0, 88.8	88.4, 73.0, 54.0	94.0, 88.0, 78.4	66.0, 57.6, 48.0
geo. shapes	16.8, 15.2, 13.6	76.0, 56.8, 30.4	24.4, 18.8, 12.0	44.4, 21.6, 6.4	29.6, 27.0, 24.8
logical deduct.	52.8, 50.8, 48.8	100.0, 98.6, 96.0	72.4, 62.8, 55.6	95.6, 92.2, 87.6	70.0, 59.6, 49.6
public rel.	66.4, 65.0, 61.8	81.8, 75.5, 66.4	28.2, 25.0, 19.1	39.1, 26.4, 13.6	57.3, 46.8, 35.5
Europ. hist.	75.2, 74.5, 72.7	76.4, 65.2, 55.2	38.8, 34.2, 30.3	41.2, 27.9, 19.4	66.1, 56.1, 45.5
ruin names	67.2, 65.6, 65.2	85.2, 83.2, 80.0	54.8, 50.6, 45.6	67.6, 60.0, 51.2	38.0, 34.4, 30.4
prof. account	60.3, 53.2, 47.5	84.0, 72.0, 58.5	36.2, 29.1, 25.5	54.6, 38.7, 24.8	42.9, 28.9, 20.2
college math	54.0, 32.0, 15.0	85.0, 59.0, 41.0	55.0, 34.0, 17.0	77.0, 58.0, 40.0	57.0, 31.5, 13.0
TAR Results					
navigation	94.4, 94.4	91.6, 15.2	65.2, 9.2	83.2, 4.8	77.6, 3.2
geo. shapes	91.6, 91.6	45.6, 0.8	60.4, 31.2	39.2, 5.6	90.4, 83.6
logical deduct.	92.8, 90.4	96.8, 7.6	80.4, 37.6	92.0, 16.4	74.4, 14.0
public rel.	92.7, 86.4	83.6, 38.2	82.7, 46.4	56.4, 0.9	61.8, 10.0
Europ. hist.	94.5, 94.5	74.5, 17.0	77.6, 41.2	53.9, 6.1	63.6, 19.4
ruin names	95.6, 97.2	93.6, 27.6	86.8, 26.8	79.2, 11.6	82.4, 20.8
prof. account	81.9, 49.3	71.3, 4.3	77.0, 44.0	57.8, 2.1	48.2, 4.3
college math	46.0, 10.0	50.0, 0.0	45.0, 3.0	54.0, 0.0	29.0, 2.0

Table 3: BestAcc, Median Accuracy, WorstAcc on top; TARa@10, TARr@10 on bottom, for the zero-shot conditions. Results are in terms of percentages.

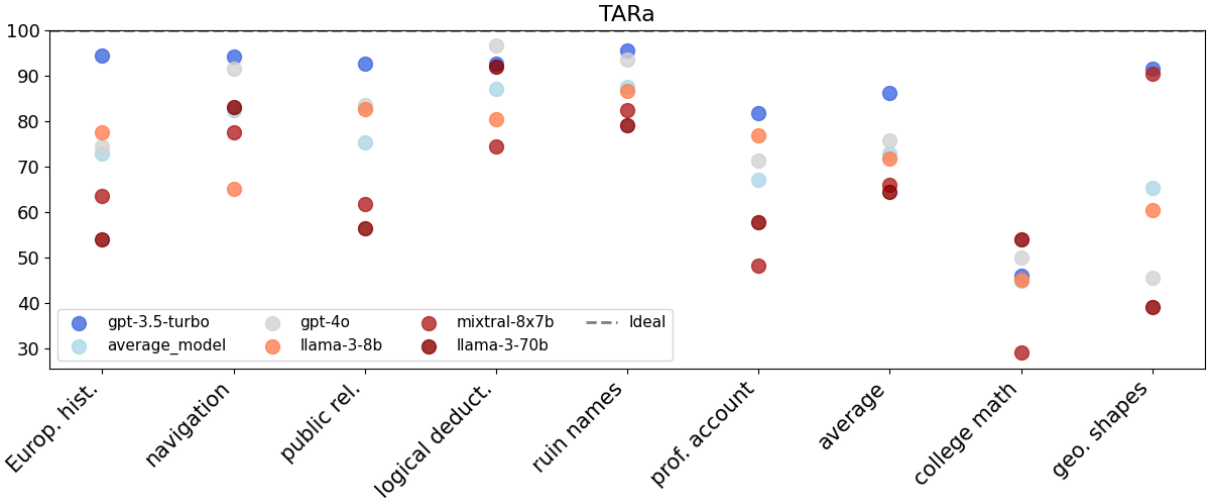

Figure 11: TARa@10 for each task in the zero-shot setting. Model colors have been chosen to distinguish them by relatively low performing (increasingly dark red hues) versus relatively high performing (increasingly dark blue hues).

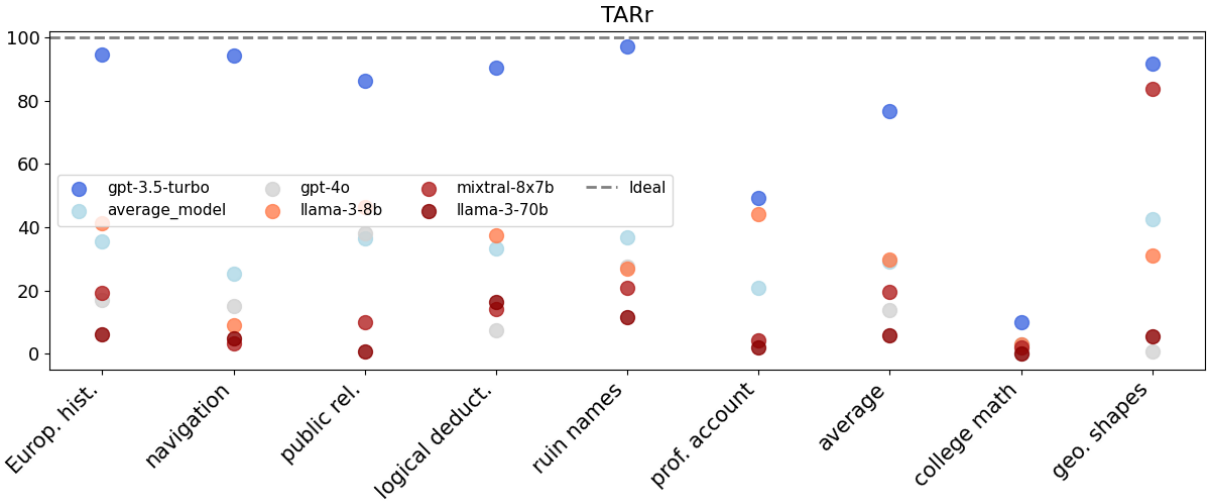

Figure 12: TARr@10 for each model in the zero-shot setting. Dataset colors have been chosen to distinguish them by relatively challenging (increasingly dark red hues) versus relatively easy (increasingly dark blue hues).