Robustness as an Emergent Property of Task Performance

Anonymous ACL submission

Abstract

Robustness is often regarded as a critical future challenge for real-world applications, where stability is essential. In this work, we question this assumption and explore the relationship between robustness and performance, hypothesizing that high performance in a task serves as a strong indicator of robustness. Through an empirical analysis of multiple models across diverse datasets and configurations (e.g., paraphrases, different temperatures), we find a strong positive correlation: as models approach high performance on a task, robustness is effectively achieved. This effect persists beyond "trivial robustness" expected from high success rates and holds across architectures. Our findings suggest that robustness is primarily driven by task-specific competence rather than inherent model-level properties, challenging current approaches that treat robustness as an independent capability. Thus, looking at the field from a high-level perspective, we may expect that as new tasks saturate model robustness on these tasks will emerge accordingly. This calls for a reduced focus on measuring and improving robustness, as it is likely to resolve naturally with performance gains.

1 Introduction

One man's trash is another man's treasure

Robustness – the ability of models to produce consistent outputs across prompt variations – remains an unsolved issue. This has important implications for real-world applications: even high-performing models can behave unpredictably under minor changes, undermining confidence in their reliability, especially in scenarios where stability is critical (Yang et al., 2024; Wang and Zhao, 2024; Ashury-Tahan et al., 2025).

Different models tend to learn tasks in a similar order, i.e., some tasks are generally easier to learn than others (Hacohen et al., 2020; Pliushch et al., 2022; Baldock et al., 2021). Once a task has been learned, the model can succeed over a wide range of specific questions. This suggests that as models internalize task representations, they may also become more robust, generalizing across different formulations of the *same* question in addition to learning entirely new ones. Thus, robustness (consistency over task formulations) may be strongly associated with performance (success over tasks).

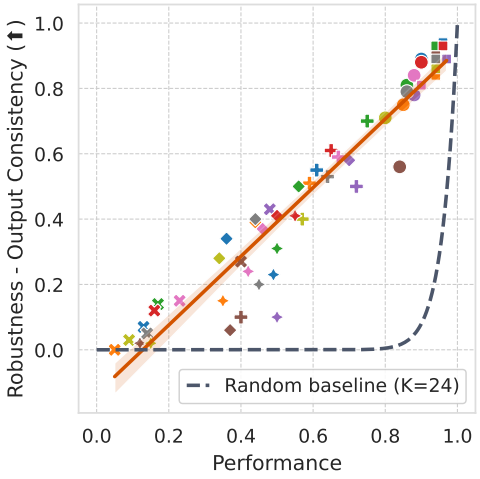


Figure 1: Linear regression of robustness on performance. Robustness increases slightly faster than performance, with a slope of 1.05. Performance explains 92.4% of robustness variance, indicating a strong predictive relationship. Colors denote models, and shapes represent datasets. Dashed gray line: random baseline, i.e., the probability of answering consistently across all example configurations, assuming per-configuration success probability equals the model's performance.

In this work, we examine the correspondence between performance and robustness. We argue and verify that **performance serves as a meaningful signal of model robustness**. When models demonstrate high success and questions are unquestionably easy for them, they are easy regardless of how they are presented. To explore this hypothesis, we analyze multiple models across diverse datasets and configurations.

Our findings reveal a strong positive correlation

between benchmark performance and model robustness, demonstrating that as model performance approaches the upper limits of a task, so does its resilience to inference variations. This phenomenon transcends the "trivial robustness" expected from high success rates and remains consistent across diverse model architectures (see Fig. 1).

Current approaches often focus on dedicated measures for model robustness. Our results challenge this approach, by demonstrating that *task-specific competence, rather than inherent model-level robustness*, is the primary driver of robust behavior.

Our work suggests that robustness can be viewed as a concomitant effect that tends to increase as a benchmark approaches saturation. Thus, as model saturation extends over time to new tasks, robustness on these tasks may emerge naturally.

2 Preliminaries

Let D denote a dataset, with examples $\{x_i \mid i \in \{1, \ldots, |D|\}\}$. Each example can be inferred using one of the configurations $v \in V$, denoted as x_i^v.

Let $m(x_i^v)$ represent the model prediction for x_i^v, and let $\text{score}(m(x_i^v))$ denote the benchmark evaluation score assigned to the prediction.

In what follows, we define the key terms used throughout our analysis.

Inference Configuration Configurations are chosen to reflect plausible real-world settings of diverse types, for which the model is expected to produce identical outputs, including surface form variations (paraphrasing), in-context modifications (modifying demonstrations and their quantity), generation parameter changes (varying temperature), and adversarial perturbations (such as noise addition). Formally, given an example x_i, we define a set of configurations $\{x_i^v \mid v \in V\}$, which share the same semantic meaning but differ in how they are presented to the model. The detailed configuration can be found in App. §A.

In our main results, all configurations are equally valid references and considered original. Otherwise, we denote x_i^o the original reference configuration. We calculate metric scores against each original reference and then average the results.

Model Capability The performance score (e.g., accuracy, F1) obtained by a model on the original version. We define the overall capability of a model m on dataset D as the average score across all examples:

$$\text{Capability}(m, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \text{score}(m(x_i^o))$$

Model Robustness (Output Consistency) Following previous work (Nalbandyan et al., 2025; Ackerman et al., 2024; Zhu et al., 2024; Habba et al., 2025; Ashury-Tahan et al., 2025), we define robustness at the example level as strict agreement of model predictions across configurations.

Given the set of configurations V, a model m is considered *robust* on example x_i if all configuration outputs are equivalent:

$$c_i = \mathbf{1}\left[m(x_i^{v_1}) \equiv m(x_i^{v_2}) \;\; \forall v_1, v_2 \in V\right]$$

Dataset-level robustness is the fraction of examples that are robust:

$$\text{OutputConsistency}(m, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} c_i$$

3 Experimental Setup

Our experiments include runs on 6 datasets and 9 models.From each dataset, we sample 100 examples and generate predictions under 24 different configurations. We then evaluate, for each model–dataset pair, both performance[1] and robustness across these configurations. More technical details are provided in App. §A.

Robustness Metrics In addition to our primary metric *Output Consistency*, we also evaluated two score-based metrics: (i) the standard deviation of scores across example configuration, and (ii) the performance drop rate across configurations. Metrics formal definitions are in Appendix §B.

Random Baseline We also compare our results to a random baseline, computed as the probability of consistent answers across all configurations, assuming that the success probability of each configuration equals the model's overall performance.

Contamination While contamination is a potential concern in evaluation studies, we aimed to minimize it in our work by focusing on *diverse* datasets and configurations, reducing the likelihood of exact overlap with pretraining corpora. Moreover, our analysis emphasizes consistency across all 24

[1]We validate it against existing benchmarks (see App §A.2).

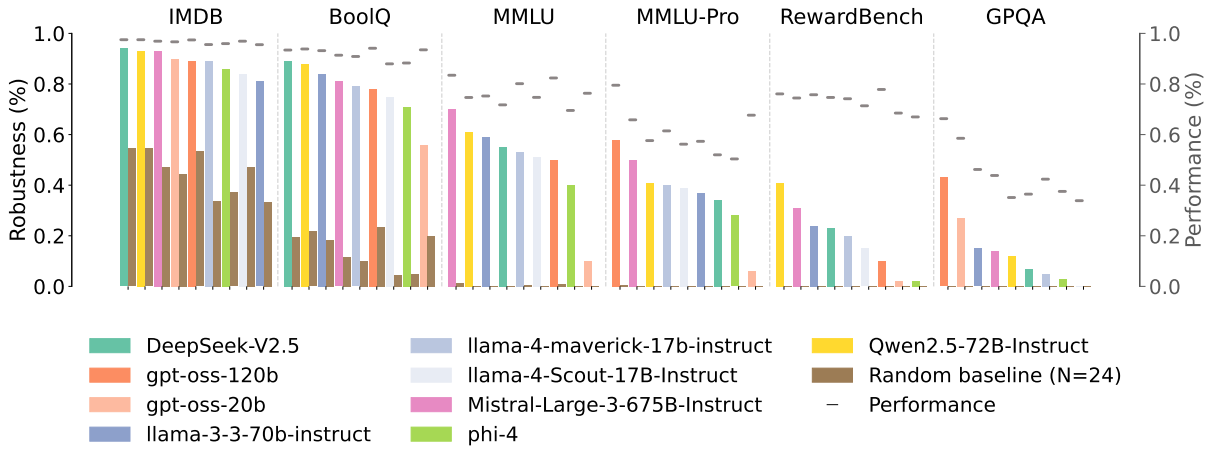


Figure 2: Robustness rate per dataset, computed using the output consistency metric, in relation to overall performance (represented by dashed lines). Robustness rises as benchmarks saturate. Camel bars represent the random baseline consistency probability with similar success rates, which is approximately 0 when performance is below ~80%. Additional robustness metrics analysis can be found in App. §C.

varied configurations, making it unlikely that contamination alone explains the observed robustness patterns. Representative evidence can be seen in the detailed aggregation of consistency patterns provided in App. §C.2, which shows that across all datasets and models there is no consistent tendency to succeed on any particular configuration (i.e., no dominance of a specific STD value); instead, the behavior follows a long-tail distribution.

4 Results

This section presents our main findings, demonstrating a strong positive correlation between performance and consistency in model outputs.

In this section we reflect the robustness results based on the output consistency metric (§2), noting that all metrics exhibit strong correlation. Additional results for the other metrics are provided in Appendix C.

4.1 Key Findings

Figure 2 illustrates the relationship between performance and robustness across models by dataset. This view reveals that, among our selected models, IMDB and BoolQ appear saturated, while GPQA remains challenging. Below, we present our findings.

Higher performance is associated with a greater proportion of consistent answers This suggests that the ability to solve a dataset also reflects an ability to generalize across configurations. For example, all models on IMDB achieve performance between 95% and 97%, and their robustness is comparatively high, ranging from 81% to 94%. This is not trivial, as illustrated in Figure 2, and becomes evident when comparing consistency to a random baseline with a similar overall success rate, which in the case of IMDB case achieves only between 33% and 54%.

Models outperform random baselines across benchmark regimes Models substantially exceed the random baseline on all benchmarks, underscoring robustness that is not explained by chance. For the four datasets on the right of Figure 2, the random baseline achieves zero robustness (as performance is lower than 80%). Even in cases of very high success probability (IMDB, BoolQ), the gap remains significant, 43.4% and 62.9% on average, respectively. An extreme case is `llama-4-Scout-17B-Instruct` on BoolQ, which outperforms the random baseline by 70.4%.

Model-specific factors are comparatively weaker Our analysis shows that although architecture and design matter, their effect on consistency is modest relative to the strong performance–robustness trend in Figure 1. Nevertheless, models can differ in inherent robustness; for example, `gpt-oss-120b` achieves a significantly higher robustness score than `gpt-oss-20b` on datasets where their performance is similar (e.g., on BoolQ, a 1-point performance difference results in 120B achieving 78% robustness compared to 56% for 20B; similarly, on MMLU, a 6-point performance gap corresponds to a 40% robustness difference).

Our results show few outliers, which hint that we observe little contamination. In contamination, one

would expect the original to be often high when paraphrases are not.

4.2 Additional Analysis

We also measure the distribution of model robustness behavior using the per-example standard deviation (STD) (see Appendix B). This distribution provides a more nuanced perspective on model consistency. It offers additional evidence that inherent model robustness is limited, as the STD distribution is similar across models within the same benchmark, i.e., their graph trends follow a comparable pattern.

Moreover, it reveals that as models achieve higher performance, their consistency increasingly exhibits a long-tail pattern: most examples show low STD (high consistency), while a smaller subset falls into the tail with higher STD values. Graphs and additional details can be found in Appendix §C.2.

4.3 Statistical Analysis

We performed an ablation study using ANOVA to confirm that our findings are not driven by arbitrary choices in the experimental setup. The results indicate that parameter choices have only a minor impact on performance. More details in Appendix §C.3.

5 Related Work

Saturation Progress Evaluation benchmarks for LLMs has become increasingly saturated (Bengio et al., 2025; Reuel et al., 2024; HAI, 2023; Ott et al., 2022; Bengio et al., 2025). Studies show rapid early gains followed by plateaus as models near perfect scores, often accelerated by data overlap (Sainz et al., 2023), underscoring the pace and capabilities of current systems. Recent work addresses saturation by introducing harder benchmarks, weighted metrics, or progressively challenging evaluations (Ivanov and Volkov, 2025; Mirzadeh et al., 2024; Etzine et al., 2025; Bradley, 2024). While these works raise the challenges in this phenomenon, such as evaluation limitations, or try to suggest solutions, other perspectives on what saturation entails for models remain underexplored.

Robustness LLM robustness has been researched over the years, with many studies highlighting brittleness and sensitivity to input variations, each focusing on a specific task (Alzahrani et al., 2024), domain (Ashury-Tahan et al., 2025), input perturbation type (Mizrahi et al., 2024), or robustness as a phenomenon and how to address it (Kumar and Mishra, 2025). However, these works have treated robustness as an isolated phenomenon. Lunardi et al. found robustness correlates with consistency, aligning with our results, though their focus was on whether benchmark scores reflect robustness. Finally, Ding et al. (2018) tied robustness to input distribution, which does not have to be task-specific, an observation that aligns with our findings.

Generalization and Learning Order A line of work in NLP and vision explores the generalization process of LLMs, showing that the order of learning and generalization tends to repeat across different architectures and training regimes (Hacohen et al., 2020; Pliushch et al., 2022; Baldock et al., 2021; Choshen et al., 2022; Edamadaka et al., 2025). Recent studies also reveal striking similarities in learned parameters (Kaushik et al., 2025). These works support our findings: models tend to generalize in a similar order, then performance on a learned task is expected to transfer to varied versions of it.

6 Discussion

While model robustness is often studied in isolation, here we take a broader perspective. We find that robustness is mainly tied to overall task performance, where high performance corresponds to robust and consistent model behavior. Interestingly, intrinsic signals from the model itself are relatively weak.

At a higher level, our work relates to the dynamics of saturation, where tasks become progressively easier for models over time. Together with the observed link between performance and robustness, it suggests that, without explicitly addressing it, *as models advance, robustness may increasingly cease to be a primary bottleneck*. This perspective aligns with prior findings that models tend to generalize on tasks in a predictable order.

These patterns matter for applications where robustness is as critical as accuracy (e.g., healthcare, safety), informing evaluation and deployment strategies.

Our findings indicate that perceived model robustness often reflects task-specific competence rather than inherent model properties, calling for a reduced focus on measuring and improving robustness in isolation.

7 Limitations

Scope of Tasks Our experiments are focused on the classification task to ensure comparability of results; however, this comes at the cost of reduced generalizability, as the findings may only partially apply to other tasks or domains.

Model Behavior Our analysis and conclusions are based on a diverse set of models. While the observed behaviors are consistent across this set, they may not generalize under substantial shifts in model architectures or training paradigms.

Evaluated Models Due to cost constraints, we did not include closed-source models in our evaluation. Consequently, caution should be exercised when generalizing these results to all model types.

References

Samuel Ackerman, Ella Rabinovich, Eitan Farchi, and Ateret Anaby Tavor. 2024. A novel metric for measuring the robustness of large language models in non-adversarial scenarios. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2794–2802, Miami, Florida, USA. Association for Computational Linguistics.

Norah Alzahrani, Hisham Abdullah Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yusef Almushaykeh, Faisal Mirza, Nouf Alotaibi, Nora Altwairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. *Preprint*, arXiv:2402.01781.

Shir Ashury-Tahan, Yifan Mai, Rajmohan C, Ariel Gera, Yotam Perlitz, Asaf Yehudai, Elron Bandel, Leshem Choshen, Eyal Shnarch, Percy Liang, and Michal Shmueli-Scheuer. 2025. The mighty torr: A benchmark for table reasoning and robustness. *Preprint*, arXiv:2502.19412.

Robert J. N. Baldock, Hartmut Maennel, and Behnam Neyshabur. 2021. Deep learning through the lens of example difficulty. *Preprint*, arXiv:2106.09647.

Yoshua Bengio, Sören Mindermann, Daniel Privitera, Tamay Besiroglu, Rishi Bommasani, Stephen Casper, Yejin Choi, Philip Fox, Ben Garfinkel, Danielle Goldfarb, Hoda Heidari, Anson Ho, Sayash Kapoor, Leila Khalatbari, Shayne Longpre, Sam Manning, Vasilios Mavroudis, Mantas Mazeika, Julian Michael, and 77 others. 2025. International ai safety report. *Preprint*, arXiv:2501.17805.

William F. Bradley. 2024. Enhancing llm evaluations: The garbling trick. *ArXiv*, abs/2411.01533.

Leshem Choshen, Guy Hacohen, Daphna Weinshall, and Omri Abend. 2022. The grammar-learning trajectories of neural language models. *Preprint*, arXiv:2109.06096.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *Preprint*, arXiv:1905.10044.

Gavin Weiguang Ding, Kry Yik-Chau Lui, Xiaomeng Jin, Luyu Wang, and Ruitong Huang. 2018. On the sensitivity of adversarial robustness to input data distributions. In *International Conference on Learning Representations*.

Sathya Edamadaka, Soojung Yang, Ju Li, and Rafael Gómez-Bombarelli. 2025. Universally converging representations of matter across scientific foundation models. *Preprint*, arXiv:2512.03750.

Bryan Etzine, Masoud Hashemi, Nishanth Madhusudhan, Sagar Davasam, Roshnee Sharma, Sathwik Tejaswi Madhusudhan, and Vikas Yadav. 2025. Revitalizing saturated benchmarks: A weighted metric approach for differentiating large language model performance. In *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*, pages 511–523, Albuquerque, New Mexico. Association for Computational Linguistics.

Eliya Habba, Ofir Arviv, Itay Itzhak, Yotam Perlitz, Elron Bandel, Leshem Choshen, Michal Shmueli-Scheuer, and Gabriel Stanovsky. 2025. Dove: A large-scale multi-dimensional predictions dataset towards meaningful llm evaluation. *Preprint*, arXiv:2503.01622.

Guy Hacohen, Leshem Choshen, and Daphna Weinshall. 2020. Let's agree to agree: Neural networks share classification order on real datasets. *Preprint*, arXiv:1905.10854.

Stanford HAI. 2023. Ai benchmarks hit saturation. https://hai.stanford.edu/news/ai-benchmarks-hit-saturation. Accessed: July 22, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Igor Ivanov and Dmitrii Volkov. 2025. Resurrecting saturated llm benchmarks with adversarial encoding. *Preprint*, arXiv:2502.06738.

Prakhar Kaushik, Shravan Chaudhari, Ankit Vaidya, Rama Chellappa, and Alan Yuille. 2025. The universal weight subspace hypothesis. *Preprint*, arXiv:2512.05117.

Pankaj Kumar and Subhankar Mishra. 2025. Robustness in large language models: A survey of mitigation strategies and evaluation metrics. *Preprint*, arXiv:2505.18658.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. *Preprint*, arXiv:2403.13787.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, and 31 others. 2023. Holistic evaluation of language models. *Preprint*, arXiv:2211.09110.

Riccardo Lunardi, Vincenzo Della Mea, Stefano Mizzaro, and Kevin Roitero. 2025. On robustness and reliability of benchmark-based evaluation of llms. *Preprint*, arXiv:2509.04013.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Iman Mirzadeh, Keivan Alizadeh-Vahid, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *ArXiv*, abs/2410.05229.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt llm evaluation. *Preprint*, arXiv:2401.00595.

Grigor Nalbandyan, Rima Shahbazyan, and Evelina Bakhturina. 2025. Score: Systematic consistency and robustness evaluation for large language models. In *North American Chapter of the Association for Computational Linguistics*.

Simon Ott, Adriano Barbosa-Silva, Kathrin Blagec, Janina Brauner, and Matthias Samwald. 2022. Mapping global dynamics of benchmark creation and saturation in artificial intelligence. *Nature Communications*, 13.

Iuliia Pliushch, Martin Mundt, Nicolas Lupp, and Visvanathan Ramesh. 2022. When deep classifiers agree: Analyzing correlations between learning order and image statistics. *Preprint*, arXiv:2105.08997.

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.

Anka Reuel, Amelia F. Hardy, Chandler Smith, Max Lamparth, Malcolm Hardy, and Mykel J. Kochenderfer. 2024. Betterbench: Assessing ai benchmarks, uncovering issues, and establishing best practices. *ArXiv*, abs/2411.12990.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier López de Lacalle, and Eneko Agirre. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. In *Conference on Empirical Methods in Natural Language Processing*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

Yuqing Wang and Yun Zhao. 2024. Rupbench: Benchmarking reasoning under perturbations for robustness evaluation in large language models. *ArXiv*, abs/2406.11020.

Zeyu Yang, Zhao Meng, Xiaochen Zheng, and Roger Wattenhofer. 2024. Assessing adversarial robustness of large language models: An empirical study. *ArXiv*, abs/2405.02764.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Zhenqiang Gong, and Xing Xie. 2024. Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. *Preprint*, arXiv:2306.04528.

A Experimental Setup

A.1 Experimental Design

Building on the definitions above, we conduct experiments on both saturated and less saturated benchmarks, incorporating different configurations.

The Data We conducted our experiments using the following benchmarks: IMDB (Maas et al., 2011), BoolQ (Clark et al., 2019), MMLU (Hendrycks et al., 2021), MMLU-Pro (Wang et al., 2024), GPQA (Rein et al., 2023), and RewardBench (Lambert et al., 2024). The datasets were chosen to share a similar classification task and use the same exact-match accuracy metric, ensuring comparability.

The Models For each dataset, we evaluated the capability and robustness of 9 open-weight models and 6 model families, all listed in Table A.2. These models were selected as open-weight representatives from different model families, allowing us to analyze how performance on a benchmark relates to robustness behavior. Each model was evaluated using all perturbation types described below.

Configurations Following previous work (Habba et al., 2025; Mizrahi et al., 2024; Alzahrani et al., 2024), we implement our experiments using the following variations with exact parameter values provided in Table A.1:

1. **Paraphrases**: An LLM judge paraphrased the original prompt, and the resulting text was used as an alternative template.
2. **Number of Demonstrations**: We varied the number of in-context examples provided.
3. **Random Noise**: Random patterns were added as prefixes, suffixes, or space replacements of varying lengths to introduce noise.
4. **Model Temperature**: Inference was performed under different temperature settings.

In total, we apply 24 configurations, each representing a unique combination of these variations to assess the robustness of model performance. The use of multiple configurations helps mitigate contamination concerns that could affect our results.

Evaluation Since our experiments focused on classification tasks, we evaluated results using exact match between the gold answer and the model output. We instructed the model to output the final answer only. Prior to output comparison, both strings were normalized (i.e., lowercasing, and stripping whitespace).

Required Computation We sampled 100 examples from each dataset and evaluated each model on all samples across 24 configurations. This yields 14,400 inferences per model and 129,600 total inferences across nine models.

A.2 Performance Verification with External Benchmarks

To sanity-check our results, we compared the performance scores we have got and presented in Table A.2 against published scores from external sources. A practical challenge is coverage: our model list includes several newer models, whereas some benchmarks (e.g., IMDB, BoolQ) are *older* and are not consistently reported for recent models. Accordingly, we focus our cross-check on widely reported tasks such as MMLU, MMLU-Pro and GPQA. The results presented below indicate a positive correlation between the scores and only minor differences, despite variations in evaluation runs between our setup and theirs.

Llama models. We compared our measurements with the scores reported on the official model cards and community evaluations (e.g., the Llama 4 Maverick model card on Hugging Face[2]). Overall, the numbers are closely aligned (Table A.3).

HELM Capabilities We used the HELM (Liang et al., 2023) capabilities leaderboard, which reports results for MMLU-Pro and GPQA on some of our reported models. The results are similar to ours (see Table A.4). One difference that may explain their slightly better scores is that they ran the evaluation with chain-of-thought (CoT) prompting, whereas we requested a final answer only.

[2] https://huggingface.co/meta-llama/Llama-4-Maverick-17B-128E-Instruct

Configuration Parameter	Number of variations	Values
Paraphrases	2	[A paraphrase created with an LLM for each dataset]
Number of demonstrations	2	2, 4
Random noise	3	no noise; replace prompt spaces with another character (e.g., TAB); add a random string at the beginning and end (70 chars)
Model temperature	2	0.2, 0.6

Table A.1: Configuration parameters in the experimental setup. We ran the experiments using each of the 24 unique configurations shown in the table.

Model	IMDB	BoolQ	MMLU	RewardBench	MMLU-Pro	GPQA
gpt-oss-120b	.97	.94	.82	.78	.80	.66
gpt-oss-20b	.97	.94	.76	.69	.68	.68
Mistral-Large-3-675B-Instruct	.97	.91	.84	.75	.66	.44
llama-4-maverick-17b-instruct	.96	.91	.80	.74	.62	.42
llama-3-3-70b-instruct	.95	.93	.75	.76	.57	.46
Llama-4-Scout-17B-Instruct	.97	.88	.75	.71	.56	.34
Qwen2.5-72B-Instruct	.97	.94	.74	.76	.57	.35
DeepSeek-V2.5	.97	.93	.72	.75	.52	.36
phi-4	.96	.88	.70	.67	.50	.38
Average	.97	.92	.76	.73	.61	.45

Table A.2: Benchmark performance summary.

Model	MMLU	MMLU-Pro
Llama 4 Maverick 17B	80 (85)	62 (62)
Llama 4 Scout	75 (79)	56 (58)
Llama 3.1 70B	75 (79)	57 (53)

Table A.3: Our measured scores (outside parentheses) versus published scores (in parentheses). Minor differences are expected due to evaluation details (e.g., prompt templates, decoding settings).

Model	MMLU-Pro	GPQA
Llama 4 Scout	75 (74)	56 (50)
Llama 4 Maverick 17B	80 (81)	62 (65)
Qwen2.5-72B-Instruct	57 (63)	35 (42)
gpt-oss-120b	80 (79)	66 (68)
gpt-oss-20b	68 (74)	68 (59)

Table A.4: Our measured scores (outside parentheses) versus published scores (in parentheses). Minor differences are expected due to evaluation details (e.g., running with CoT).

B Robustness Metrics

B.1 Robustness Main Metric

While it is common practice to measure robustness using scores, we find it somewhat less ideal to reflect both performance and robustness using the exact same numbers. Instead, we rely primarily on the model's outputs in the paper, which offer a more direct reflection of its ability to maintain consistent predictions under meaning-preserving variations.

It is important to note that this choice *does not affect the validity* of our results: all robustness metrics we examined yielded similar trends.

B.2 Other Robustness Metrics

We used the following robustness metrics to complement our main output-based measure and provide a broader perspective on model behavior under different configurations:

Standard Deviation (STD) We compute the standard deviation of the model's scores across the original and perturbed versions of each input. Formally, for a given input x_i and its variants $\{x_i^v \mid v \in V\}$, we calculate:

$$\sigma_{x_i} = \text{STD}\left(\{\text{score}(m(x_i^v)) \mid v \in V\} \right)$$

A value of 0 indicates perfectly consistent behavior across perturbations, while higher values reflect increased variability in the model's responses.

Performance Drop Rate (PDR) We follow (Zhu et al., 2024) and calculate the relative drop in performance when the model is evaluated on perturbed inputs compared to the original test set. We define:

$$\rho_{x_i}(\%) = \frac{1}{|V|} \sum_{v \in V} [1 - \frac{\text{score}(m(x_i^o))}{\text{score}(m(x_i^v))}]$$

$$PDR(m, D) = \frac{1}{|D|} \sum_{x_i \in D} \rho_{x_i}$$

Here, performance is measured using the primary evaluation metric. A higher percentage indicates a greater sensitivity to perturbations.

C Results

We provide the results of 2 additional robustness metrics described in App. §B, and show they correlate well with our main results in the paper. While the primary metric used in the paper in output consistency which is string-based, these metrics provide another aspects that are score based.

C.1 PDR Trends

Calculating the PDR at the example level and averaging across models reveals a pattern similar to our main result as can be seen in Figure B.1. Here, the bars illustrate how far performance can vary per instance, essentially indicating where score consistency is lacking, which complements what we examined in Figure 2, where the bars reflected output consistency. In this figure, they represent performance variability rather than stability. We observe the same dataset ordering as before, but reversed from left to right: GPQA remains the least robust and IMDB the most robust. The relationship with overall performance is also evident when comparing the bar trend to the dashed trend.

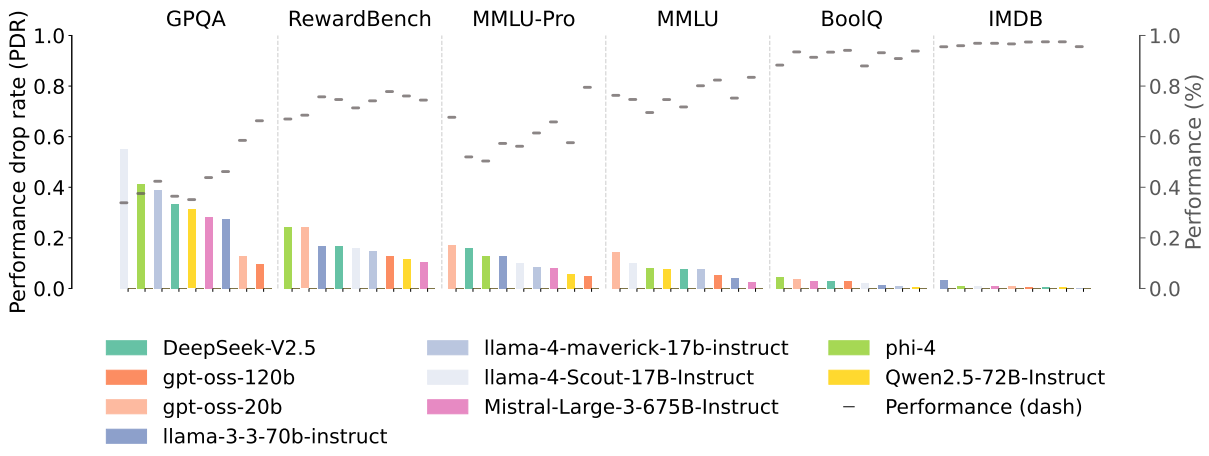


Figure B.1: Average PDR reveals a similar trend: lower values indicate smaller performance drops across perturbations, signaling higher robustness and more consistent model behavior. The dashed lines indicates the performance.

C.2 Model-Dataset Full STD Distributions

To provide a more granular perspective on our results, we extended the STD metric by calculating the full distribution of per-example STD across models. These distributions reveal detailed patterns of model consistency.

While an STD of 0 indicates a consistent score, it is important to distinguish between success consistency and failure consistency. In our setup, success consistency means producing the same correct answer across all configurations, whereas consistent failure can occur due to different incorrect answers, which does not necessarily indicate robust behavior. Therefore, in the figures, we not only reflect the STD score but also differentiate whether an STD of 0 corresponds to all failures or all successes.

Considering only successful cases, the results closely mirror what we presented in Figure 2. While that figure reflected output consistency, here we focus on score consistency. It is also evident that as robustness decreases, the STD distribution becomes flatter, whereas higher performance leads to an increasingly long-tail distribution.

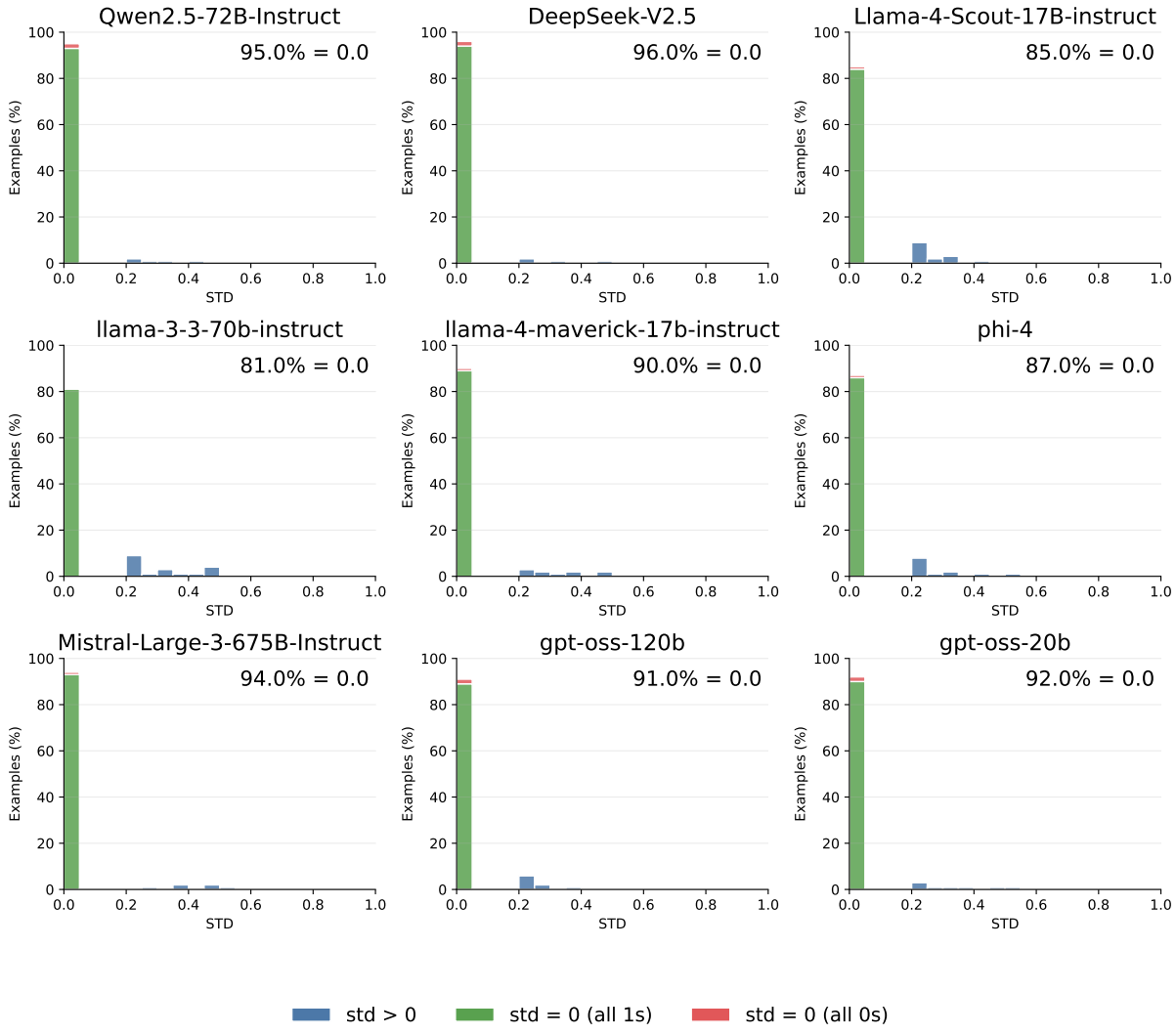


Figure C.1: Per-example STD distribution for each model on the IMDb dataset.

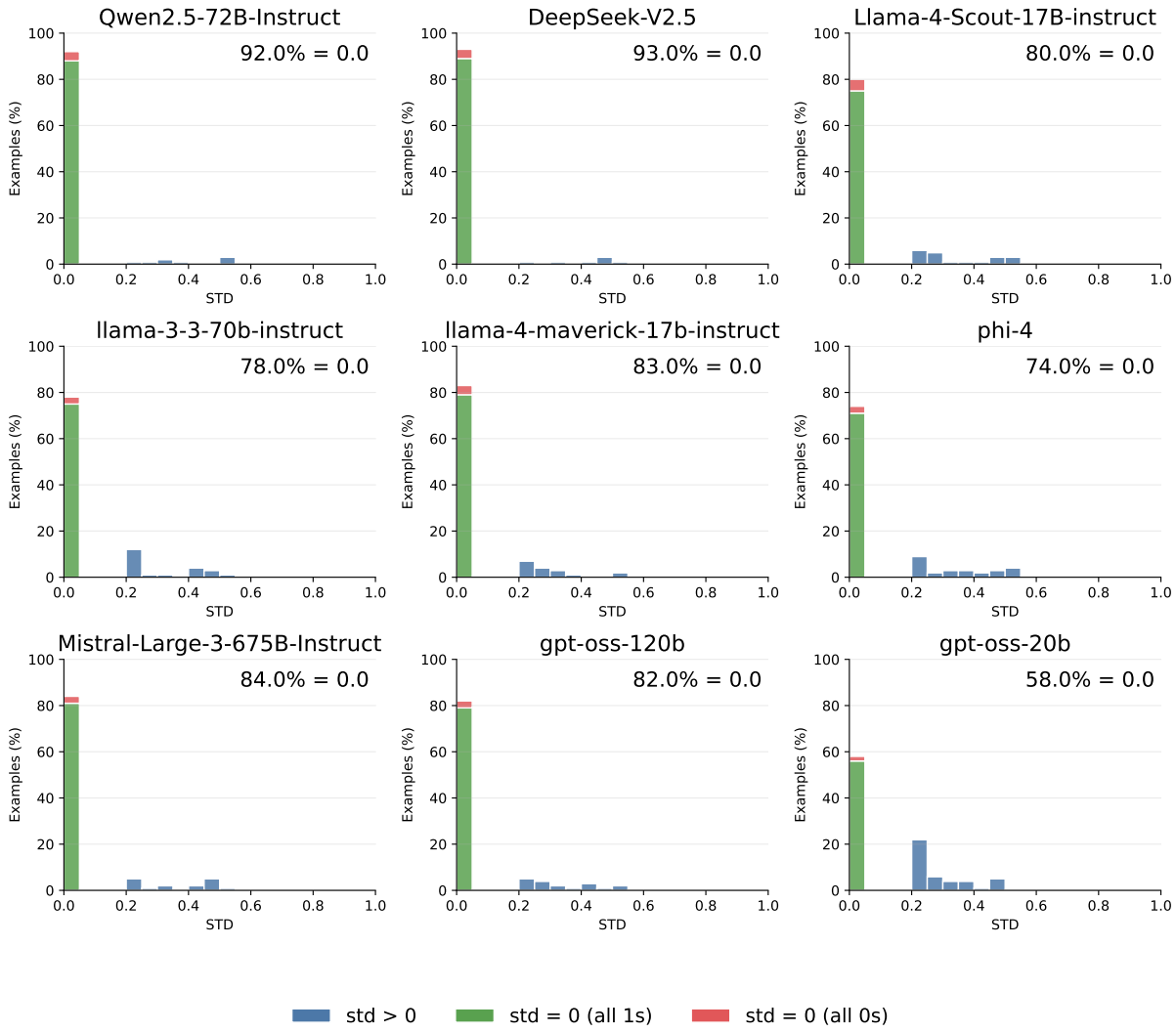


Figure C.2: Per-example STD distribution for each model on the BoolQ dataset.

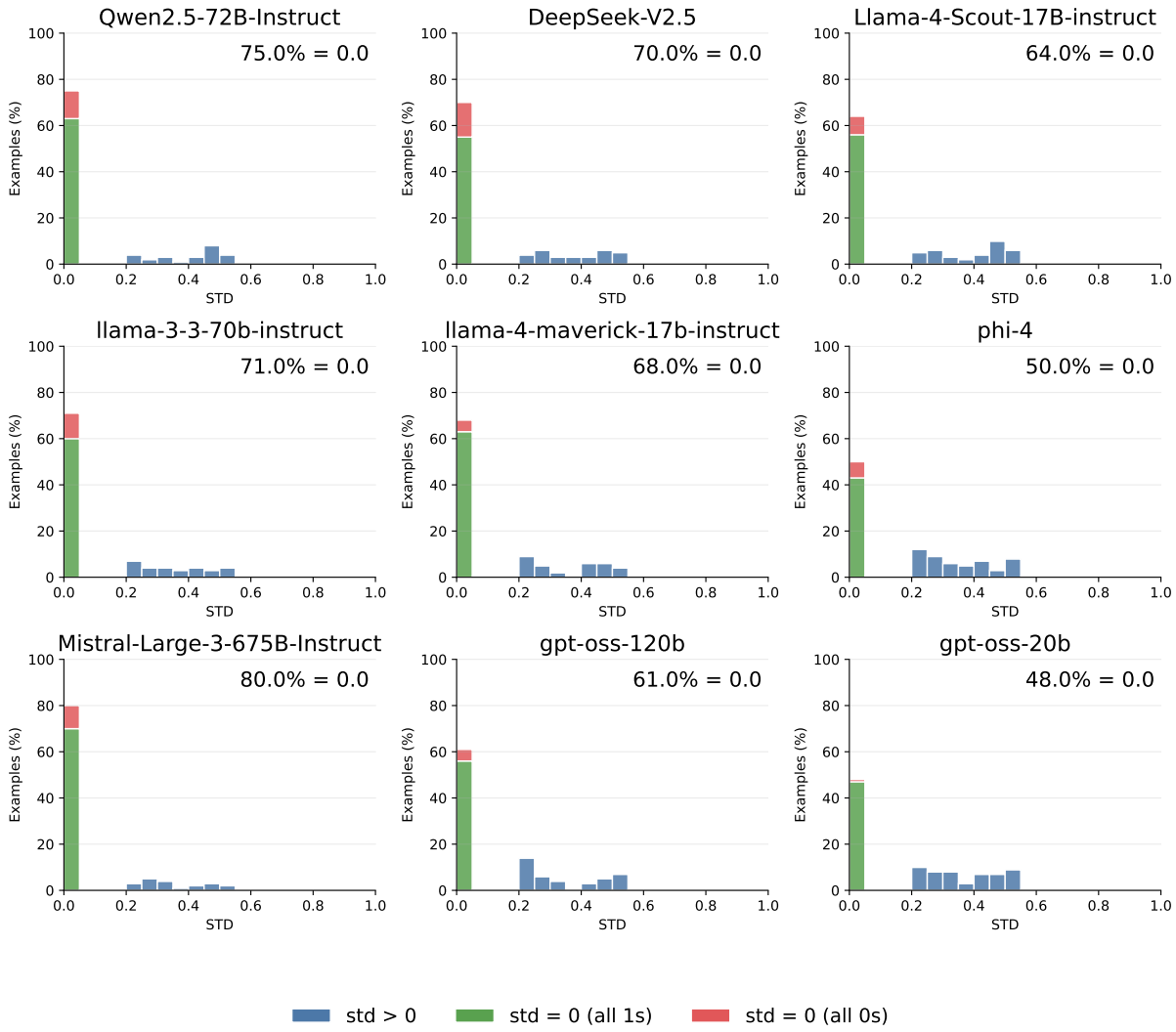


Figure C.3: Per-example STD distribution for each model on the MMLU dataset.

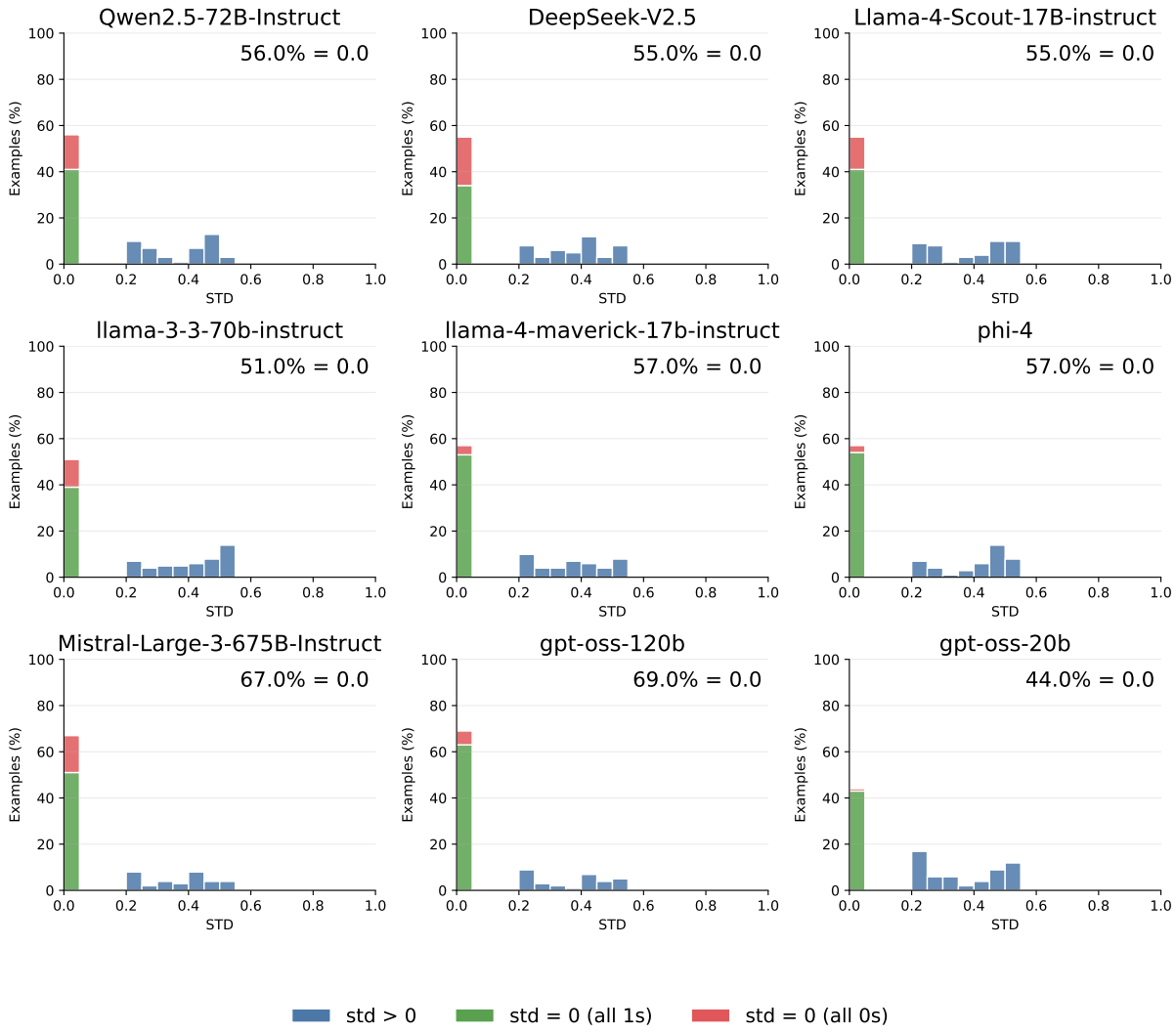


Figure C.4: Per-example STD distribution for each model on the MMLU-Pro dataset.

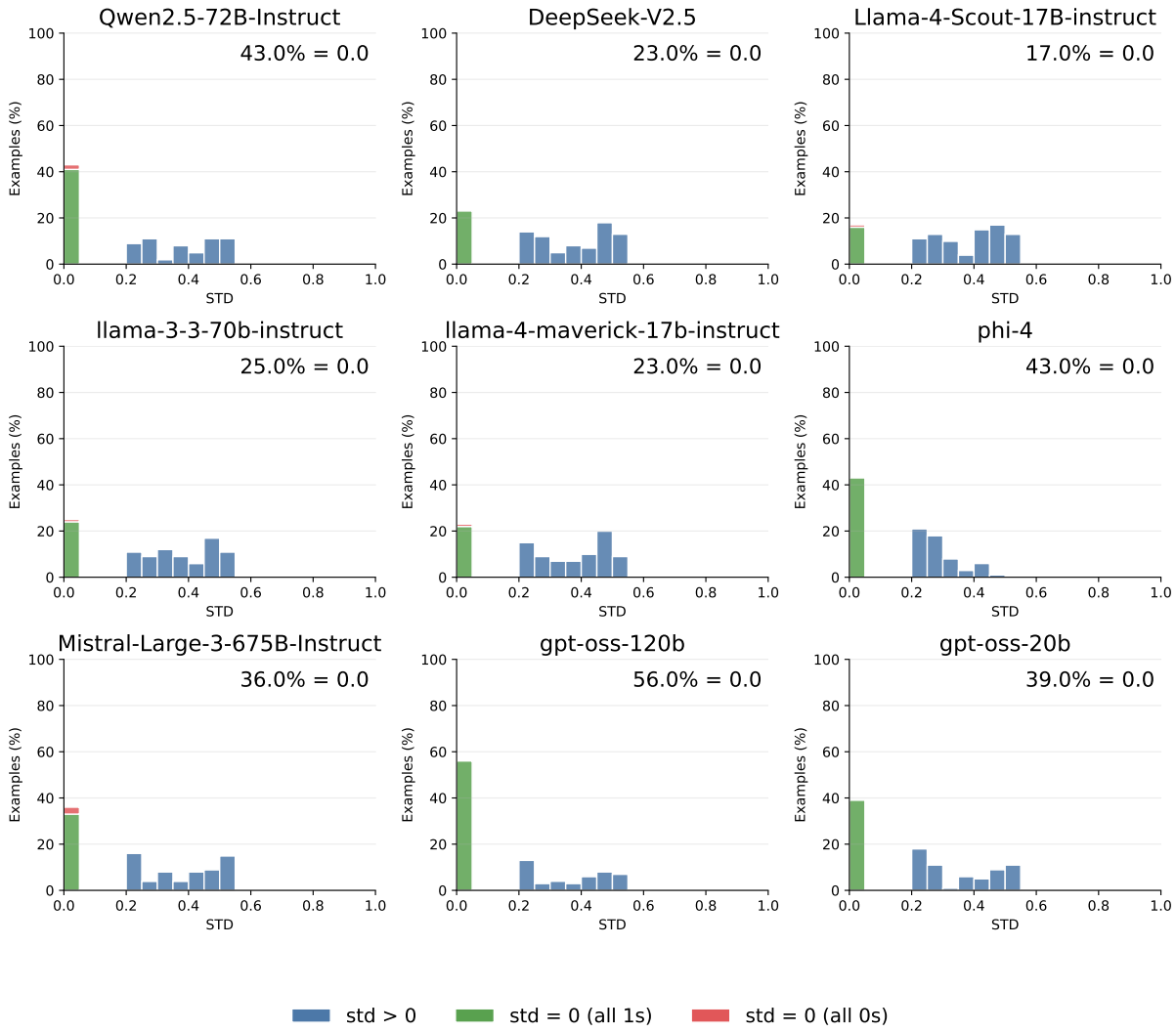


Figure C.5: Per-example STD distribution for each model on the RewardBench dataset.

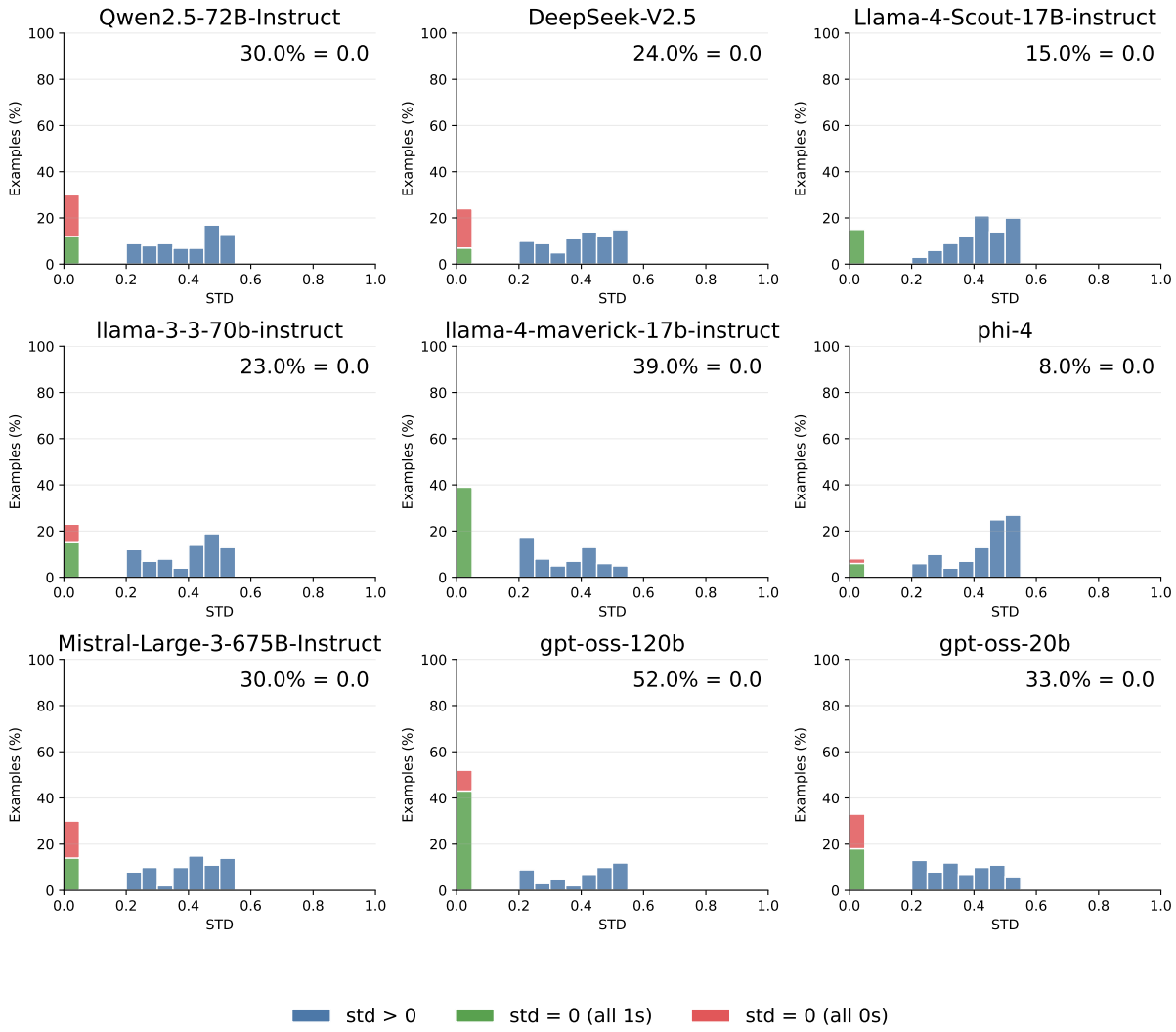


Figure C.6: Per-example STD distribution for each model on the GPQA dataset.

C.3 Statistical Testing

We analyzed the influence of four experimental factors (number of demos, prompt variation, template, and temperature) on performance across the datasets using both Type II and Type III ANOVA. Type II ANOVA evaluates each factor after accounting for all other factors, assuming no interaction terms, while Type III ANOVA additionally considers the presence of interactions and tests each factor after adjusting for all other factors and interactions. These tests assess whether different choices for each parameter significantly affect the results.

Overall, the findings indicate that parameter choices have only a minor impact on performance. As shown in Table C.1 and Table C.2, the majority of p-values (approximately 70%) are not significant. While prompt variation and number of demos exhibit statistically significant differences in some datasets (e.g., RewardBench, MMLU and MMLU-Pro), their effect sizes remain very small (<0.005), suggesting limited practical impact.

dataset	factor	F	p-value	eta_sq	partial_eta_sq	sum_sq
BoolQ	C(num_demos)	0.245	0.620	1.135e-05	1.137e-05	0.018
BoolQ	C(prompt_variation)	12.338	4.412e-06	0.001	0.001	1.848
BoolQ	C(template_used)	2.289	0.130	1.059e-04	1.060e-04	0.171
BoolQ	C(temprature)	0.022	0.881	1.030e-06	1.032e-06	0.002
GPQA	C(num_demos)	0.061	0.805	2.924e-06	2.924e-06	0.015
GPQA	C(prompt_variation)	1.279	0.278	1.224e-04	1.224e-04	0.633
GPQA	C(template_used)	0.003	0.954	1.579e-07	1.579e-07	8.164e-04
GPQA	C(temprature)	0.583	0.445	2.785e-05	2.786e-05	0.144
IMDB	C(num_demos)	0.123	0.726	6.015e-06	6.016e-06	0.004
IMDB	C(prompt_variation)	0.892	0.410	8.747e-05	8.748e-05	0.058
IMDB	C(template_used)	0.128	0.721	6.275e-06	6.276e-06	0.004
IMDB	C(temprature)	0.947	0.331	4.641e-05	4.642e-05	0.031
MMLU	C(num_demos)	4.836	0.028	2.236e-04	2.239e-04	0.869
MMLU	C(prompt_variation)	15.438	1.996e-07	0.001	0.001	5.547
MMLU	C(template_used)	3.175	0.075	1.468e-04	1.470e-04	0.570
MMLU	C(temprature)	0.724	0.395	3.346e-05	3.352e-05	0.130
MMLU-Pro	C(num_demos)	5.030	0.025	2.328e-04	2.329e-04	1.197
MMLU-Pro	C(prompt_variation)	6.235	0.002	5.771e-04	5.773e-04	2.968
MMLU-Pro	C(template_used)	1.124	0.289	5.202e-05	5.206e-05	0.268
MMLU-Pro	C(temprature)	0.063	0.802	2.914e-06	2.916e-06	0.015
RewardBench	C(num_demos)	10.468	0.001	4.799e-04	4.824e-04	2.037
RewardBench	C(prompt_variation)	35.940	3.686e-23	0.005	0.005	20.980
RewardBench	C(template_used)	2.003	0.157	9.184e-05	9.235e-05	0.390
RewardBench	C(temprature)	1.439	0.230	6.595e-05	6.632e-05	0.280

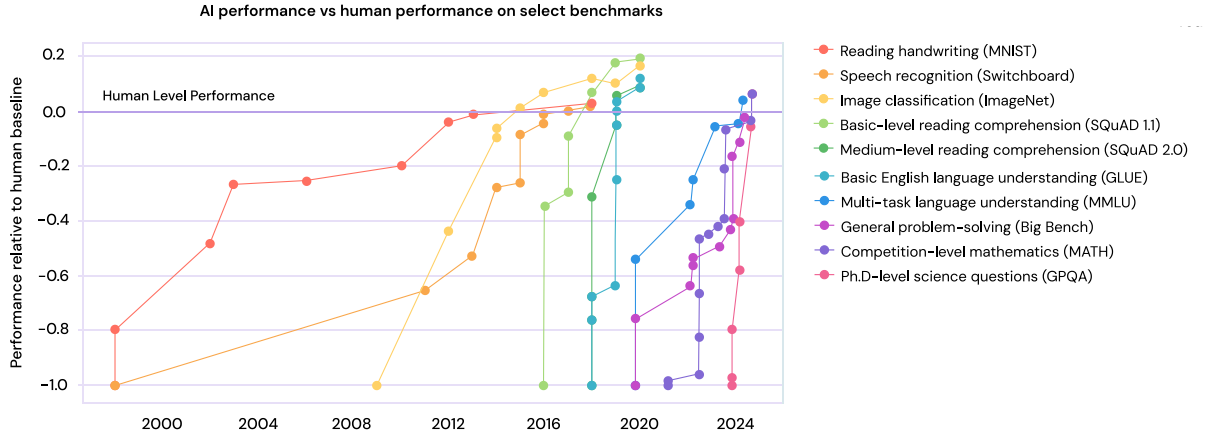
Table C.1: Summary across datasets — Type II ANOVA.

dataset	factor	F	p-value	eta_sq	partial_eta_sq	sum_sq
BoolQ	C(num_demos, Sum)	0.245	0.620	9.265e-07	1.137e-05	0.018
BoolQ	C(prompt_variation, Sum)	12.338	4.412e-06	9.319e-05	0.001	1.848
BoolQ	C(template_used, Sum)	2.289	0.130	8.645e-06	1.060e-04	0.171
BoolQ	C(temprature, Sum)	0.022	0.881	8.410e-08	1.032e-06	0.002
BoolQ	Intercept	243192.586	0.000	0.918	0.918	18209.710
GPQA	C(num_demos, Sum)	0.061	0.805	1.615e-06	2.924e-06	0.015
GPQA	C(prompt_variation, Sum)	1.279	0.278	6.761e-05	1.224e-04	0.633
GPQA	C(template_used, Sum)	0.003	0.954	8.723e-08	1.579e-07	8.164e-04
GPQA	C(temprature, Sum)	0.583	0.445	1.539e-05	2.786e-05	0.144
GPQA	Intercept	16936.111	0.000	0.447	0.447	4187.914
IMDB	C(num_demos, Sum)	0.123	0.726	2.021e-07	6.016e-06	0.004
IMDB	C(prompt_variation, Sum)	0.892	0.410	2.939e-06	8.748e-05	0.058
IMDB	C(template_used, Sum)	0.128	0.721	2.108e-07	6.276e-06	0.004
IMDB	C(temprature, Sum)	0.947	0.331	1.559e-06	4.642e-05	0.031
IMDB	Intercept	586755.026	0.000	0.966	0.966	18989.883
MMLU	C(num_demos, Sum)	4.836	0.028	5.261e-05	2.239e-04	0.869
MMLU	C(prompt_variation, Sum)	15.438	1.996e-07	3.359e-04	0.001	5.547
MMLU	C(template_used, Sum)	3.175	0.075	3.454e-05	1.470e-04	0.570
MMLU	C(temprature, Sum)	0.724	0.395	7.873e-06	3.352e-05	0.130
MMLU	Intercept	70297.530	0.000	0.765	0.765	12630.152
MMLU-Pro	C(num_demos, Sum)	5.030	0.025	9.103e-05	2.329e-04	1.197
MMLU-Pro	C(prompt_variation, Sum)	6.235	0.002	2.257e-04	5.773e-04	2.968
MMLU-Pro	C(template_used, Sum)	1.124	0.289	2.034e-05	5.206e-05	0.268
MMLU-Pro	C(temprature, Sum)	0.063	0.802	1.140e-06	2.916e-06	0.015
MMLU-Pro	Intercept	33643.358	0.000	0.609	0.609	8007.085
RewardBench	C(num_demos, Sum)	10.468	0.001	4.018e-04	4.824e-04	2.037
RewardBench	C(prompt_variation, Sum)	35.940	3.686e-23	0.004	0.005	20.980
RewardBench	C(template_used, Sum)	2.003	0.157	7.689e-05	9.235e-05	0.390
RewardBench	C(temprature, Sum)	1.439	0.230	5.522e-05	6.632e-05	0.280
RewardBench	Intercept	4240.655	0.000	0.163	0.164	825.159

Table C.2: Summary across datasets — Type III ANOVA.

D Saturation Progress

Saturation in LLM evaluation is a well-known challenge, with numerous papers highlighting, analyzing, or accounting for it in their studies. Here, we provide evidence from existing research showing that saturation is becoming increasingly prevalent. This strengthens our findings: as models improve in capabilities and benchmarks lose relevance, their ability to generalize also becomes more critical.



A figure published as part of the International AI Safety Report (Bengio et al., 2025) demonstrating the rapid advance of AI model performance from 1998 to 2024. In recent benchmarks, models progressed quickly from poor performance to surpassing human experts.

Table 1: Descriptive statistics of reported results over time for specific benchmarks and AI tasks. A single task can be represented through several benchmarks.

	NLP	Computer vision	Total
Benchmarks with ≥ 1 reported result	1318	2447	3765
Benchmarks with ≥ 3 results at different time points (% of above)	661 (50%)	1274 (52%)	1935 (51%)
AI tasks with ≥ 1 reported result	346	601	947
AI tasks with ≥ 3 results at different time points (% of above)	197 (57%)	386 (64%)	583 (62%)

Figure D.1: A table taken from (Ott et al., 2022) that demonstrates how, over time, the number of benchmarks relevant for repeated reporting drops significantly. For example, only 50% of NLP benchmarks have results at three or more time points.

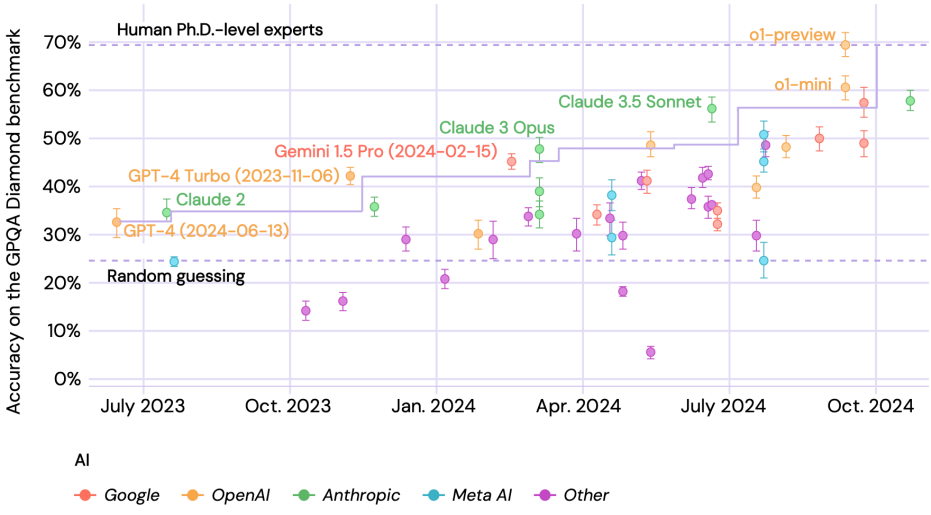


Figure D.2: A figure published as part of the International AI Safety Report (Bengio et al., 2025) demonstrating that AI models improved from near-random (33%) to expert-level (70%) accuracy on PhD-level science questions within 15 months,

E AI Assistance Usage

We used AI for paraphrasing and improving clarity of writing only.