How Many Samples Do We Need? A Toolkit for Power-Aware Evaluation Design

Angelo Basile[1,2], **Areg Mikael Sarvazyan**[3], **José Ángel González**[4]
[1]Universitat Politècnica de València
[2] United Nations International Computing Centre (UNICC)
[3]Symanto Research,[4]TransPerfect
abasile@upv.edu.es

Abstract

If *datasets are the telescopes of our field*[1], then statistical power is their resolution, i.e., their ability to reveal a true difference in model performance when one exists. Many NLP evaluations are underpowered, leading to overstated claims of improvement. This paper introduces `sk-power`, an open-source Python library that helps researchers and practitioners design well-powered evaluations. Built with familiar `scikit-learn`-style abstractions, `sk-power` enables users to simulate evaluation scenarios, estimate minimum detectable effects, and assess the reliability of reported gains. We also illustrate — via a post-hoc analysis of recent LLM benchmarks — what can go wrong when power analysis is not carried out. Our goal is to position power analysis as a fundamental step in evaluation planning.

Keywords: evaluation, power, LLMs, simulation, Python

1. Introduction

In this work, we address a basic question in empirical NLP research: *how large should a test set be to know if a model truly outperforms another?* Previous work has shown that widely used benchmarks are often too small to support strong claims of improvement (Card et al., 2020a; Søgaard et al., 2014).

Large Language Models (LLMs) make this problem worse for several reasons. High-quality evaluation data are scarce and expensive to produce, making it tempting to test LLMs on only a limited number of examples. Consequently, benchmarks often saturate quickly, prompting the development of smaller yet more challenging datasets that demand substantial resources and expert effort to construct. At the same time, many LLM evaluations involve stochastic decoding or subjective human judgments, both of which increase performance variance, and therefore require larger sample sizes to reliably detect genuine differences. Although statistical safeguards can mitigate these issues, many practitioners still lack accessible tools or sufficient statistical training to determine whether observed differences are meaningful, leading to frequent overinterpretation of small or noisy improvements (Dror et al., 2017, 2018).

A natural way to estimate the appropriate dataset size for an evaluation is through *statistical power*, i.e., the probability that a model evaluation experiment will detect a real difference between two models. To make such analyses accessible and reproducible we introduce `sk-power`, a Python library that estimates the dataset size needed to detect meaningful performance differences by simulating model evaluations and computing the power offered by a benchmark.

Our contributions are as follows:

- We introduce `sk-power`, a modular Python library for conducting both pre-hoc and post-hoc power analyses. It includes reusable low-level components as well as task-specific interfaces designed for common NLP evaluation scenarios.
- We show how `sk-power` can be used to plan evaluation designs under resource constraints, helping researchers determine the dataset size needed to detect differences of interest.
- We also apply `sk-power` in a *post-hoc* analysis of over 2,000 model comparisons from the Open LLM Leaderboard (Beeching et al., 2023; Fourrier et al., 2024), finding that many claimed improvements in current practice are not statistically reliable.

Together, these contributions aim to make power analysis a standard part of model evaluation in NLP, and to support more robust and interpretable conclusions about model performance.

`sk-power` is available at https://github.com/jogonba2/power-analysis

[1]Attributed to Aravind Joshi, https://tinyurl.com/mr6trr4j

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 4507–4513
11–16 May 2026.

2. Power Analysis for Evaluation Design

When comparing two models on the same data, each is evaluated using a performance metric M (e.g., accuracy, win rate, or a human preference score). The central quantity of interest is the *difference in performance* between the models on that dataset. This observed difference reflects not only any true disparity in model performance, but also sampling variability, decoding randomness, or annotator noise. Consequently, the observed performance difference cannot be interpreted directly as evidence that one model is genuinely better than the other.

A statistical test provides a principled way to assess whether the observed gap is larger than what would be expected if the models would perform similarly on average. Based on the observed data and test results, we compute a *p-value*, the conditional probability of observing a gap at least as large as the one obtained, assuming there is no real difference. If this probability falls below a predefined significance level α (e.g., 0.05), the test rejects the null hypothesis and concludes that the difference is statistically significant. This procedure bounds the risk of a *Type I error*, i.e., incorrectly declaring a difference when none exists. The complementary risk is a *Type II error*, in which a true difference goes undetected; its probability is denoted β.

The *power* of a test, defined as $1 - \beta$, is the probability that it will correctly identify a real difference when one exists. Power depends on the size of the effect of interest, the sample size, the variability of the data, the employed test, and the chosen significance level. In practice, it serves as a key design quantity: by fixing the desired power (e.g., 0.8) and the smallest meaningful improvement, one can determine how many evaluation examples are needed to make the experiment sensitive enough to detect that improvement with high reliability.

2.1. Power Estimation through Simulation

Although closed-form expressions for power exist for several tests, NLP evaluations rarely satisfy their assumptions. Comparisons are typically *paired* (i.e., same items for both models), metrics are discrete or bounded (e.g., accuracy, win rate, preference rate, etc.), and sample sizes are modest. Non-parametric procedures (e.g., permutation or bootstrap) are often preferred. In `sk-power`, we estimate power by simulation, which offers a unified approach that applies across different evaluation designs without relying on fragile distributional assumptions.

Crucially, we simulate *evaluation outcomes*, not text. The simulator operates in the score space: it draws synthetic realizations of the metric inputs (e.g., correct/incorrect indicators, win/loss preferences) according to a simple data-generating process calibrated to the evaluation setting, rather than generating prompts or output text.

Given a candidate sample size N, a significance level α, the number of iterations S, the procedure is:

1. **Specify a data-generating process (DGP).** Define a stochastic model that mirrors the evaluation setup, including each model's marginal success rate, their dependence (e.g., a disagreement rate for paired accuracy), and variance components such as annotator disagreement, decoding stochasticity, prompt variants, or item difficulty.

2. **Simulate evaluations.** For each iteration $s = 1, \ldots, S$, draw one synthetic evaluation of size N from the DGP.

3. **Compute metric differences.** Apply the chosen metric (e.g., accuracy) to the synthetic predictions and compute the observed gap.

4. **Apply the statistical test.** Given a difference in performance, is it due to chance? Test and compute a p-value and record whether it rejects the null hypothesis at level α.

5. **Estimate power.** Compute the proportion of simulations that reject the null hypothesis: this is power.

6. **Explore design trade-offs.** Vary N or the assumed true effect to produce *power curves* (power vs. N) and *minimum detectable effect (MDE)* curves (smallest detectable improvement at the desired power).

3. Library Design

The goal of `sk-power` is to make statistical power analysis both accessible and reproducible for NLP evaluation. The library follows a clear division of responsibility. At the *low level*, it exposes modular statistical components for users with a statistical background who wish to define, validate, or extend the base workflow. At the *high level*, it provides task-specific, scikit-learn–style interfaces that let practitioners answer concrete design questions.

Low-level API. This layer exposes the core building blocks of simulation-based power analysis as presented in Section 2.1: (i) a *data-generating process* (DGP) that samples evaluation outcomes under specified parameters (e.g., paired predictions with a given disagreement rate), (ii) an *effect-size*

estimator that maps outcomes to a scalar difference (e.g., accuracy difference or Cohen's g), and (iii) a *statistical test* that maps outcomes to p-values (e.g., McNemar's test for paired accuracy, binomial or bootstrap tests for preference data). Components are retrieved from registries by name, independently tested, and can be swapped or extended without modifying the simulator.

High-level API. The high-level layer packages these components into *Tasks* representing common evaluation setups (e.g., binary classification on a shared test set, human preference win rates). Each task behaves like a scikit-learn estimator: users instantiate it with a few parameters and call `.compute_power()` to obtain power curves and MDEs. This design keeps the statistical machinery correct and reusable while surfacing a simple interface for day-to-day use.

3.1. Designing an Evaluation

A typical workflow begins with specifying the baseline and the expected range of new model performance, the candidate sample sizes, and the significance level. Listing 1 shows a minimal example using the `BinaryClassificationTask`. This code produces a *power vs. accuracy difference* curve (see Figure 1) that visualizes how power relates to effect sizes, across different number of test items.

```
from power.tasks import
    BinaryClassificationTask

task = BinaryClassificationTask(
    acc_a=0.70,          # base
    accuracy
    acc_b=np.linspace( # new
    results
        0.70,0.75, 10),
    dataset_size=[
    10, 20, 50, 100, 500],
)

report = task.compute_power(
    iterations=1000,
    alpha=0.05)
```

Listing 1: Estimating power for a binary classification setup.

The same interface can also be used to scan effect sizes (e.g., varying `acc_b`) or to request MDE curves at a fixed power target; both are produced by the same simulation engine.

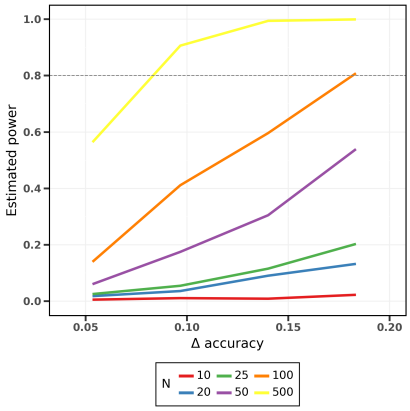

Figure 1: Power curves across different sample sizes (N). The dashed horizontal line shows the 80% power threshold, commonly taken as the minimum power required to conduct an experiment.

4. Post-hoc power analysis of recent LLM Benchmarks

In recent years, leaderboard-driven evaluation has become central to NLP research. As new LLMs are released at an increasingly rapid pace, the ability to assess whether leaderboard movements reflect real improvements or are simply the result of statistical noise, has become essential.

While `sk-power` was primarily designed to support pre-hoc evaluation design, its simulation-based architecture makes it equally suitable for post-hoc power analysis.

As an example, we examine a subset of models from the now-retired Open LLM Leaderboard (Beeching et al., 2023; Fourrier et al., 2024). Our analysis focuses exclusively on models that are publicly available on the Hugging Face Hub[2]. We group models into three categories based on their parameter count: small (fewer than 1 billion parameters), medium (between 1 and 10 billion), and large (more than 10 billion). The benchmark includes five datasets: `IFEval` (Zhou et al., 2023), `BBH` (Suzgun et al., 2022), `MATH Lvl 5` (Hendrycks et al., 2021), `GPQA` (Rein et al., 2023), and `MMLU-PRO` (Wang et al., 2024). Table 1 lists the number of test examples in each dataset. Datasets with more examples provide higher average statistical power in experiments.

Within each model size group, we compare each model against the group baseline, defined as the median performer. Figure 2 presents the results of these pairwise comparisons. As expected, smaller datasets such as `IFEval` tend to exhibit lower power on average, making it harder to detect meaningful differences. Additionally, smaller models, which often perform close to the baseline, struggle

[2] https://huggingface.co/docs/hub/index

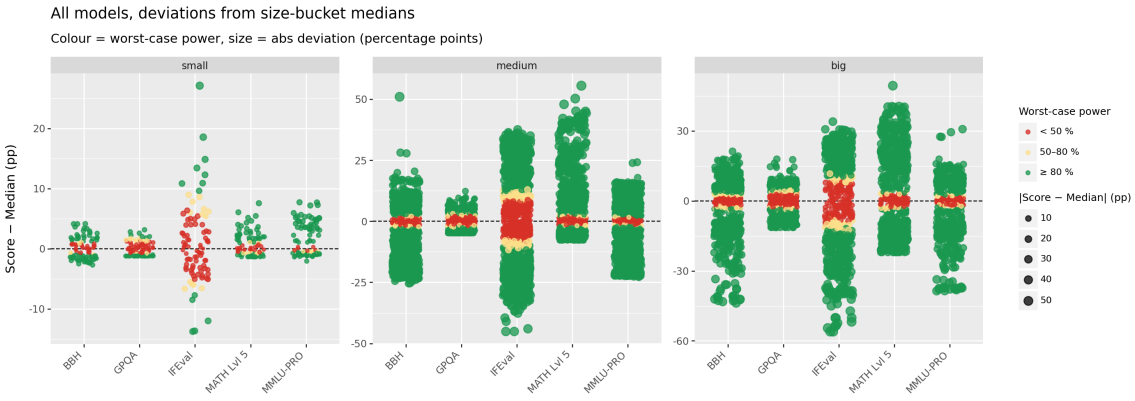

Figure 2: Each point represents a two-way test between a model and its corresponding median baseline, with the vertical axis showing the difference in score relative to the baseline. The horizontal dotted line denotes the performance of the baseline. Colors show the statistical power of the comparison.

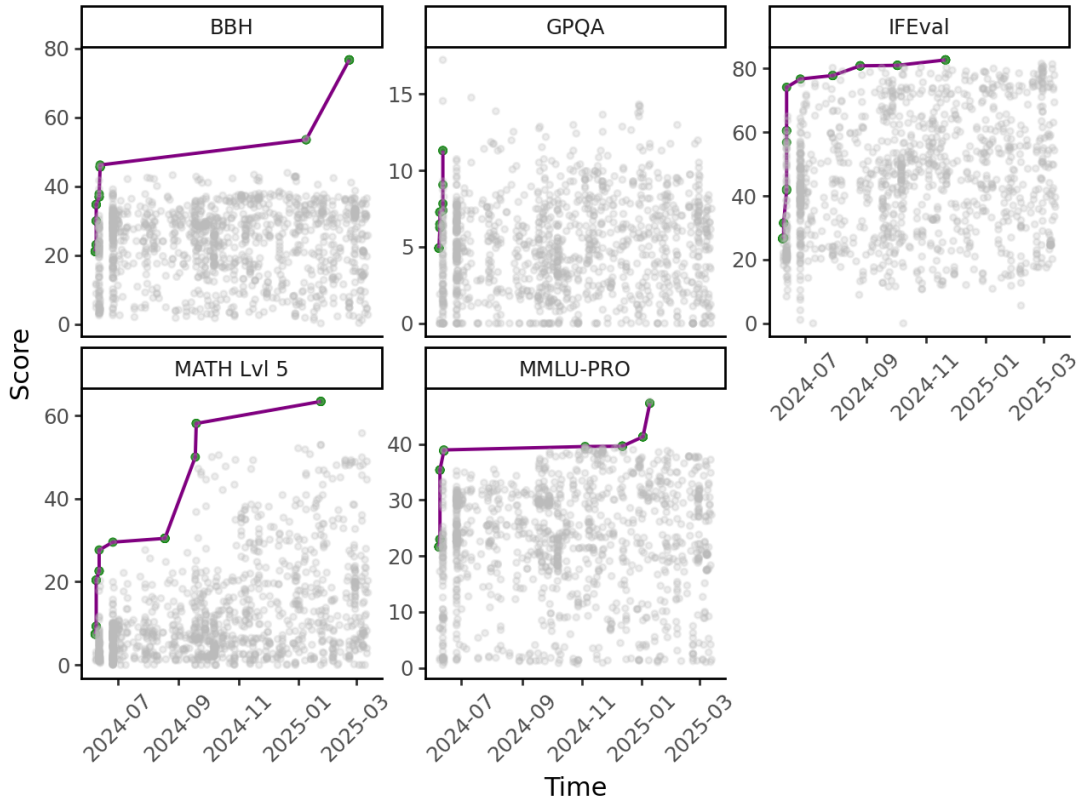

Figure 3: Scores for medium-sized models on the Open LLM Leaderboard (June 6, 2024–March 13, 2025). The purple line connects submissions that outperform their predecessor with at least 80% statistical power.

to reach statistically significant improvements.

For an alternative view of the same data, we examine how model performance evolves over time. We analyze the statistical power of Open LLM submissions from June 2024 to March 2025, focusing on medium-sized models for simplicity. Figure 3 shows the model scores over time across the datasets that compose Open LLM. The purple trajectories highlight the few instances where leaderboard progress is statistically reliable. Each connected point marks a submission that achieves a new top score with more than 80% power relative to the previous best, indicating that the improvement is unlikely to be due to chance. In contrast, the dense gray clouds show that most reported gains fall within the range of statistical uncertainty. For smaller datasets such as IFEval, limited sample size appears to be a key constraint: even when

	SIZE
IFEval	541
BBH	6511
Math Lvl 5	2648
MMLU-PRO	9497
GPQA	1252

Table 1: Overview of the datasets (instances).

models achieve modest improvements, the dataset does not provide enough power to confirm these differences as statistically reliable. Larger benchmarks like BBH and MATH Lvl 5, by contrast, exhibit more frequent power-supported gains, reflecting their greater sensitivity to small but genuine performance differences.

Overall, these patterns suggest that while leaderboard scores fluctuate continually, only a minority of updates correspond to reliably detectable progress, underscoring the importance of integrating power analysis into evaluation design.

5. Related Work

A clear introduction to statistical power and its limitations is given by Gelman and Carlin (2014). Building on simulation-based estimation (Efron and Tibshirani, 1994), our approach follows prior work in NLP (Morgan, 2006; Yeh, 2000; Søgaard et al., 2014) that advocates nonparametric, simulation-based inference under realistic assumptions.

Several studies have argued that the approach to statistical significance in NLP evaluation lacks rigor and requires methodological improvement (Dror et al., 2017, 2018). The closest sources to our work are Card et al. (2020b) and Zhu et al. (2020); the first one proposes a general algorithm and working code to estimate statistical power in NLP with a generative process, while the latter inspired the creation of an easy-to-use tool for power computation in NLP. We extend this work to account for the recent advancements brought by LLMs.

Modern LLM benchmarks are often smaller than those of the pre-LLM era, when evaluations could be broad and large-scale. This creates a growing tension: evaluation is becoming more costly and selective, yet also more susceptible to statistical noise. Many benchmarks still omit basic features such as error bars, which are vital for conveying uncertainty in performance estimates (Miller, 2024). Even when uncertainty is reported, it typically relies on asymptotic approximations like the central limit theorem, which are unreliable with small samples or skewed predictions (Bowyer et al., 2025). Practitioners also frequently fine-tune models on narrow, domain-specific datasets and report gains without assessing their statistical significance. Our work addresses these issues by providing a general-purpose framework for power analysis that enables rigorous evaluation in small-data settings.

While general-purpose tools like G*Power (Faul et al., 2007, 2009) and scipy's statistical functions (Virtanen et al., 2020) are widely used for power analysis across the social, behavioral, and biomedical sciences, `sk-power` is designed specifically for NLP evaluation. G*Power provides comprehensive coverage of classical statistical tests through a standalone desktop application, while scipy offers low-level statistical primitives for Python users. In contrast, `sk-power` addresses the distinct challenges of model evaluation in NLP: paired comparisons on shared test sets, discrete and bounded metrics (accuracy, F1, win rates), and variance from multiple sources including stochastic decoding, annotator disagreement, and prompt selection. By providing task-specific interfaces with familiar scikit-learn-style abstractions, `sk-power` makes it straightforward for NLP researchers to answer practical questions without requiring deep statistical expertise or manual parameter specification.

6. Conclusion

We introduced `sk-power`, an open-source Python library. This work was driven by a simple question: when designing an evaluation, how many test examples are needed to detect a meaningful difference between models? The answer depends on the task, effect size, data variability, and statistical test. `sk-power` provides a practical way to explore this space and design evaluations that are adequately powered. Applied to the Open LLM Leaderboard, `sk-power` shows that many reported improvements fall below standard power thresholds, especially on smaller benchmarks and among closely matched models. These results highlight the need to integrate power analysis into evaluation design.

Limitations

While `sk-power` offers a principled and simulation-based framework for evaluating statistical power in NLP settings, certain limitations remain. First, although power analysis provides valuable insights into whether observed differences are likely detectable given an evaluation design, its frequentist formulation does not directly quantify the probability that one model truly outperforms another. Furthermore, `sk-power` is still an early-stage library. Although its core components are robust and extensively tested, the current implementation emphasizes a targeted set of evaluation settings and statistical tests.

Acknowledgments

We are grateful to Desi Ivanova for her support and insights during the development of this work. We also thank the anonymous reviewers for their thoughtful comments and suggestions, which helped us improve both the clarity and quality of the paper.

The views and opinions expressed in this publication are solely those of the authors and do not necessarily reflect the official position or views of the United Nations International Computing Centre (UNICC).

7. Bibliographical References

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard (2023-2024). https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard.

Sam Bowyer, Laurence Aitchison, and Desi R Ivanova. 2025. Position: Don't use the clt in llm evals with fewer than a few hundred datapoints. *arXiv preprint arXiv:2503.01747*.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020a. With little power comes great responsibility. *arXiv preprint arXiv:2010.06595*.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020b. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274, Online. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. 2017. Replicability analysis for natural language processing: Testing significance with multiple datasets. *Transactions of the Association for Computational Linguistics*, 5:471–486.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Bradley Efron and Robert J. Tibshirani. 1994. *An Introduction to the Bootstrap*. CRC Press, New York.

Franz Faul, Edgar Erdfelder, Axel Buchner, and Albert-Georg Lang. 2009. Statistical power analyses using g* power 3.1: Tests for correlation and regression analyses. *Behavior research methods*, 41(4):1149–1160.

Franz Faul, Edgar Erdfelder, Albert-Georg Lang, and Axel Buchner. 2007. G* power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior research methods*, 39(2):175–191.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

Andrew Gelman and John Carlin. 2014. Beyond power calculations: Assessing type s (sign) and type m (magnitude) errors. *Perspectives on psychological science*, 9(6):641–651.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset.

Evan Miller. 2024. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*.

William Morgan. 2006. Statistical hypothesis tests for nlp. https://cs.stanford.edu/people/wmorgan/sigtest.pdf. Accessed: 2025-07-22.

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.

Anders Søgaard, Anders Johannsen, Barbara Plank, Dirk Hovy, and Hector Martínez Alonso. 2014. What's in a p-value in NLP? In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 1–10, Ann Arbor, Michigan. Association for Computational Linguistics.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.

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.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models.

Haotian Zhu, Denise Mak, Jesse Gioannini, and Fei Xia. 2020. NLPStatTest: A toolkit for comparing NLP system performance. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 40–46, Suzhou, China. Association for Computational Linguistics.