Benchmark Agreement Testing Done Right: A Guide for LLM Benchmark Evaluation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent advancements in Language Models (LMs) have catalyzed the creation of multiple benchmarks. A crucial task, however, is assessing the validity of the benchmarks themselves. This is most commonly done via *Benchmark Agreement Testing* (BAT), where new benchmarks are validated against established ones using some agreement metric (e.g., Spearman correlation). Despite the crucial role of BAT for benchmark builders and consumers, there are no standardized procedures for such agreement testing, which can lead to invalid conclusions and mistrust. By analyzing over 40 prominent benchmarks, we show how some overlooked methodological choices can significantly influence BAT results. To address these inconsistencies, we propose a set of best practices and demonstrate their impact on robustness and validity. To foster adoption and facilitate future research, we introduce BenchBench (links in the App), a Py package and Leaderboard for BAT.

1 Introduction

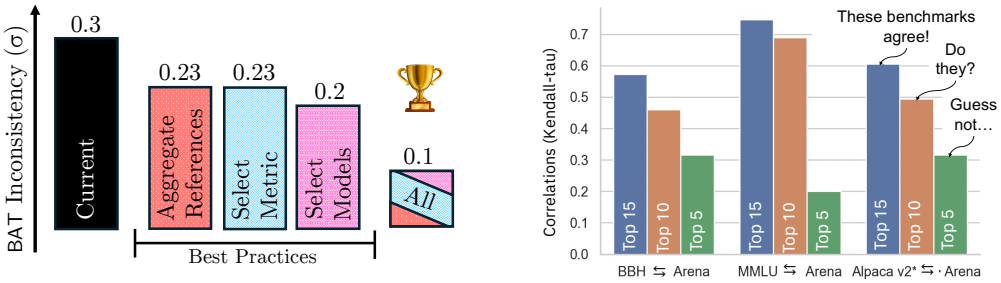
As Language Models (LMs) increasingly excel across a broad range of tasks, new benchmarks – often measuring similar abilities – are constantly proposed. This deluge of benchmarks underscores the importance of *Benchmark Agreement Testing* (BAT). BAT involves validating a new benchmark by comparing it against established and trusted benchmarks, using statistical agreement metrics. This comparison is based on the performance scores of models across the different benchmarks.

BAT is often used to validate that a new proposed benchmark measures what it was designed to measure. The expectations from this measurement depend on the benchmark's goal; demonstrating high agreement can serve to show that a new benchmark captures model abilities similar to those measured by established and well trusted benchmarks. [15, 33, 4, 17, 28]. High agreement can also validate that an efficient version of a benchmark (e.g., requiring less compute or labeling) measures the same thing as the original benchmark [26–28, 34]. In contrast, if a benchmark aims to test a unique trait – one that is not properly covered by existing benchmarks – BAT will be used to demonstrate the disagreement of such benchmarks with existing ones [35]. The above goals are relevant both for benchmark creators and for benchmark consumers. Creators will typically use BAT to validate the properties of their new benchmark; benchmark consumers might use it to choose which existing benchmark they want to use.

However, despite the wide application of BAT in recent years, there is a glaring absence of common methodology. Specifically, the significance of several methodological decisions in BAT is currently overlooked, undermining the validity of any conclusions made.

In this work, we aim to bring order and consistency into the practice of BAT. Analyzing more than 40 of the most common benchmarks (§2), spanning over 200 models, we show the critical impact of several methodological decisions in BAT, effectively altering the conclusions that researchers will draw from their analyses (§3).

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.



(a) **Running BAT using our best practices increases consistency by 3x.** The average standard deviation of BAT results over multiple instances is drastically decreased using our best practices, without incurring further computational costs. These best practices can be easily applied using our BenchBench package. For Further details, see Table 1 in the appendix.

(b) **BAT Conclusions change with models considered**. K-τ correlations between LMSys Arena and three other benchmarks. Bars represent correlation for different sets of top models, top 5, 10, and 15. The number of top models considered impacts agreement, highlighting that different selections of models can vary conclusions about benchmark agreement.

We focus on three such critical choices: selecting the reference benchmark (§3.1), the models included in the test (§3.2), as well as the correlation metrics and their interpretation (§3.3). For example, as seen in Fig. 1b, choosing a different subset of models produces substantially different correlation scores, leading to different conclusions about benchmark agreement. The figure demonstrates that two benchmarks can (and often do) show high agreement across a wide range of models, while agreement over a few top-ranked models remains low.

Building on our findings, we propose a set of best practices for robust BAT (§4) and demonstrate their impact (Fig. 1a. To foster adoption, we implement these guidelines in *BenchBench*, a Python package and dynamic leaderboard for standardized benchmark evaluation (§7.2).

Notably, when using BenchBench, applying our best practices for running BAT will not require further computational or data resources. Furthermore, BenchBench is built to continually evolve, allowing easy addition of new benchmarks, allowing users to make more informed evaluation decisions.

To sum up, our contributions are as follows:

1. We perform a large-scale analysis of BAT, stressing the impact of methodological decisions (§3).
2. We propose guidelines for reliable and standardized BAT (§4) and demonstrate their impact.
3. We release BenchBench, a Python package and meta-benchmark for BAT implementing the guidelines and incorporating them with the required benchmark data (§7.2).

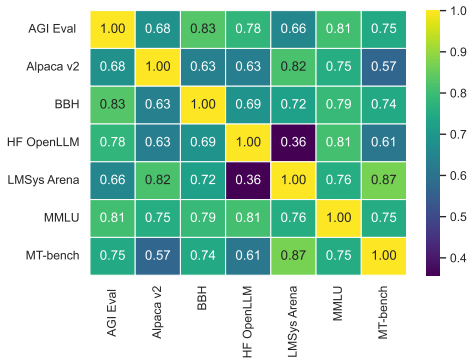
2 Setup

For our analysis, we use over 40 benchmarks [38, 18, 10, 2, 12, 23, 37, 32, 5, 7, 36, 20, 29, 9, 24, 16]. For a wider survey of benchmarks used, see App. 7.4. Performing BAT, we focus on evaluating agreement between two benchmarks – a *reference benchmark* (established and commonly acceptable) and a *target benchmark* (the one we assess, e.g., a new benchmark). Specifically, agreement is calculated as the correlation over the models ranks (using Kendall [14]) or scores (using Pearson [25]).

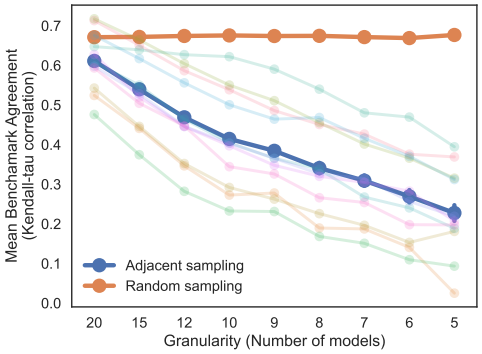
3 BAT Methodological Decisions: An Analysis

When conducting BAT, researchers face a multitude of decisions: which reference benchmarks to compare against, which models to select for comparison, which metrics to use, how to define "agreement" between benchmarks, and so on. In the absence of guidelines, benchmark creators often make arbitrary choices, without clear justification or consistency across different studies.

In this section, we demonstrate how such arbitrary choices hinder the validity of BAT conclusions and highlight how commonly reported results can foster false expectations among benchmark consumers.



(a) **Agreement scores significantly vary across different appropriate reference benchmarks**. Kendall-tau correlations between pairs of benchmarks that are seemingly valid for BAT. Each is taken over 20 models sampled at random.



(b) **Agreement is lower for closely ranked models.** Correlation (y) between each benchmark (lines) and the rest. Thick lines are averages of models, sampled randomly (orange) or adjacently (blue), shaded lines show individual benchmarks listed in App 7.6.

3.1 The Choice of Reference Benchmark Matters

Selecting a reference benchmark for BAT is a non-trivial task, as it requires a well-established benchmark with available data and significant model overlap. These strict requirements often lead researchers to use only one or two references. Furthermore, because benchmarks can be grouped by ability (e.g., holistic, coding, math), the selection of a specific reference from a pool of seemingly appropriate options is often an arbitrary choice.

Fig. 2a illustrates the variability caused by such arbitrary choices: for each target benchmark, different reference benchmarks produce wildly varying agreement scores. For example, Alpaca V2's agreement spans from a mediocre 0.57 with MT-bench to a high 0.82 with LMSys Arena, although both of these measure similar abilities. This variability calls into question the validity of conclusions based on applying BAT with a single reference benchmark and stressing the need for an aggregated reference benchmark, consolidating signals of multiple benchmarks (see more on this in §4).

3.2 The Choice of Models Matters

The selection of models for analysis is a critical yet often overlooked step in BAT. Here, we demonstrate that two key factors—the total number of models used and the specific granularity of the subset being analyzed—can dramatically alter the conclusions drawn from an agreement test.

First, the reliability of BAT is highly dependent on the number of models used for comparison. Our analysis finds that with an insufficient number of models, agreement scores become unstable, with a standard deviation approaching 0.25 for sets of six models (see Fig. 3a for details). For instance, the calculated correlation between two benchmarks like LMSys Arena and MT-Bench can vary dramatically—from 0.65 to 0.99—depending on the random sample. This demonstrates that conclusions drawn from a small model set are potentially unreliable.

Second, even when a sufficient number of models are used, a single overall agreement score can be misleading. To investigate this, we analyze agreement over subsets of adjacently-ranked models (e.g., models 3-7). Because these models have similar performance, their relative rankings are less stable, resulting in significantly lower correlation scores compared to randomly sampled models, an effect that is particularly strong for smaller subsets (Fig. 2b).

These findings emphasize the importance of reporting scores at multiple granularities. Doing so manages the expectations of consumers and provides a more complete picture of a benchmark's ability to distinguish not only between strong and weak models but also among top-tier competitors.

3.3 The Choice of Correlation Metric (and Threshold) Matters

In BAT, a correlation score is typically interpreted against a fixed threshold to determine if agreement is "high" or "low." However, there are no consistent standards for which metric or threshold to use.

To better understand these choices, we analyse the relationship between rank (Kendall-tau) and score (Pearson) correlation metrics. Our analysis finds that although rank (Kendall-tau) and score (Pearson) correlations are strongly related ($r^2 = 0.85$), they crucially have a consistent score difference of approximately 0.2. This systematic bias demonstrates a fundamental flaw in applying the same interpretation threshold to both. We therefore argue for a data-driven, comparative approach to interpreting correlation scores, as detailed in §4.

4 BAT Best Practices

Use an *Aggregate* Reference Benchmark The choice of reference benchmark can significantly affect the validity of BAT conclusions, as demonstrated by the variability in agreement scores when different single benchmarks are used as references (§3.1, Fig. 2a). To mitigate this variability, we propose combining the results from all benchmarks appropriate for the goal of the BAT (e.g., benchmarks measuring similar or dissimilar abilities) into an aggregate reference benchmark by averaging their model win-rates. This approach reduces the influence of outliers and provides a more stable and robust measure of agreement, leading to more reliable conclusions. By combining results from a group of benchmarks, achieving convergent validity [3], the aggregate benchmark provides both a more stable and robust (30% in Fig. 1a) basis for comparison.

Use a Data-driven Threshold Using predetermined thresholds to interpret correlation scores can be misleading, as the relative nature of “high” or “low” agreement varies depending on the context, such as model granularity (§3.3, Fig. 2b). A more accurate and context-aware assessment can be achieved by using a data-driven approach that compares the target benchmark’s agreement with a reference benchmark (preferably an aggregate) to the distribution of agreement scores from various other benchmarks against the same reference.

Use More (Randomly Sampled) Models and Report Multiple Granularities BAT based on a small set of models yields unreliable results with high variance, where the standard deviation can reach 0.25 with fewer models (§3.2). To enhance reliability and reduce this variability, we recommend using a diverse set of at least 10 models, preferably more, sampled randomly to minimize bias. A larger, more representative sample provides a more stable evaluation, a practice that, as shown in Table 1, decreases result variance by more than 30

Furthermore, even with a large model set, a single overall agreement score can be misleading. As demonstrated in §3.2, agreement is often high across a broad range of models but low among top-ranked competitors, which can mislead consumers seeking fine-grained distinctions. To address this, we recommend reporting agreement scores at multiple resolutions (e.g., for rolling subsets of 5, 10, or 20 adjacent models). This provides a more nuanced view and highlights critical distinctions that are otherwise missed, such as the frequent disagreement on the very top-ranked models.

Follow The Above Rules! Properly implementing these guidelines requires significant effort. Recognizing this difficulty, we developed BenchBench, a Leaderboard that incorporates our recommendations and allows easy addition of new benchmarks. As shown in Fig. 1a, the cumulative effect of our best practices reduces BAT variance by $\sim 67\%$.

5 Discussion and Conclusions

In this work, we highlighted the lack of standardized methodology in BAT, showing how arbitrary choices regarding models, reference benchmarks, and metrics can significantly alter conclusions. To address this, we proposed a set of best practices and released the BenchBench Python package and leaderboard to facilitate consistent, reliable evaluations. These contributions foster a more standardized approach to benchmark validation, enabling more accurate comparisons across the field. While our work standardizes the ’how’ of BAT, open questions regarding ’when’ and ’how’ to interpret the results remain, as discussed in App.6.3.

References

[1] Elron Bandel, Yotam Perlitz, Elad Venezian, Roni Friedman-Melamed, Ofir Arviv, Matan Orbach, Shachar Don-Yehyia, Dafna Sheinwald, Ariel Gera, Leshem Choshen, Michal Shmueli-Scheuer, and Yoav Katz. Unitxt: Flexible, shareable and reusable data preparation and evaluation for generative ai, 2024. URL https://arxiv.org/abs/2401.14019.

[2] Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard, 2023.

[3] Kevin D Carlson and Andrew O Herdman. Understanding the impact of convergent validity on research results. *Organizational Research Methods*, 15(1):17–32, 2012.

[4] Ting-Yun Chang, Jesse Thomason, and Robin Jia. Do localization methods actually localize memorized data in llms? *ArXiv*, abs/2311.09060, 2023. URL https://api.semanticscholar.org/CorpusID:265213092.

[5] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, Suchir Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374, 2021. URL https://api.semanticscholar.org/CorpusID:235755472.

[6] Leshem Choshen, Ariel Gera, Yotam Perlitz, Michal Shmueli-Scheuer, and Gabriel Stanovsky. Navigating the modern evaluation landscape: Considerations in benchmarks and frameworks for large language models (LLMs). In Roman Klinger, Naozaki Okazaki, Nicoletta Calzolari, and Min-Yen Kan, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024): Tutorial Summaries*, pages 19–25, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-tutorials.4.

[7] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[10] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

[11] Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, September 2021. URL https://doi.org/10.5281/zenodo.5371628.

[12] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300, 2020. URL https://api.semanticscholar.org/CorpusID:221516475.

[13] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021.

[14] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–93, 1938. URL https://api.semanticscholar.org/CorpusID:120478295.

[15] Fangyu Lei, Qian Liu, Yiming Huang, Shizhu He, Jun Zhao, and Kang Liu. S3eval: A synthetic, scalable, systematic evaluation suite for large language models. *ArXiv*, abs/2310.15147, 2023. URL https://api.semanticscholar.org/CorpusID:264436382.

[16] Tianle Li, Wei-Lin Chiang, Evan Frick, Dunlap Lisa, Zhu Banghua, Gonzalez Joseph E., and Ion Stoica. From live data to high-quality benchmarks: The arena-hard pipeline, April 2024. URL https://lmsys.org/blog/2024-04-19-arena-hard/.

[17] Xiang Li, Yunshi Lan, and Chao Yang. Treeeval: Benchmark-free evaluation of large language models through tree planning. *ArXiv*, abs/2402.13125, 2024. URL https://api.semanticscholar.org/CorpusID:267760188.

[18] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023.

[19] 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, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models, 2023.

[20] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022.

[21] Nelson F. Liu, Tony Lee, Robin Jia, and Percy Liang. Do question answering modeling improvements hold across benchmarks? In *Annual Meeting of the Association for Computational Linguistics*, 2021. URL https://api.semanticscholar.org/CorpusID:252846670.

[22] Moran Mizrahi, Guy Kaplan, Daniel Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt llm evaluation. *ArXiv*, abs/2401.00595, 2023. URL https://api.semanticscholar.org/CorpusID:266693922.

[23] Sam Paech. Magi benchmark. https://sampaech.substack.com/p/creating-magi-a-hard-subset-of-mmlu, 2024. Accessed: 2024-04-20.

[24] Samuel J. Paech. Eq-bench: An emotional intelligence benchmark for large language models, 2023.

[25] Karl Pearson. Vii. note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240 – 242, 1895. URL https://api.semanticscholar.org/CorpusID:121644161.

[26] Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. Efficient benchmarking (of language models). *ArXiv*, abs/2308.11696, 2023. URL https://api.semanticscholar.org/CorpusID:261076362.

[27] Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinybenchmarks: evaluating llms with fewer examples. *ArXiv*, abs/2402.14992, 2024. URL https://api.semanticscholar.org/CorpusID:267897919.

[28] Ameya Prabhu, Vishaal Udandarao, Philip H.S. Torr, Matthias Bethge, Adel Bibi, and Samuel Albanie. Lifelong benchmarks: Efficient model evaluation in an era of rapid progress. *ArXiv*, abs/2402.19472, 2024. URL https://api.semanticscholar.org/CorpusID:268091214.

[29] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019.

[30] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[31] Kaiser Sun, Adina Williams, and Dieuwke Hupkes. The validity of evaluation results: Assessing concurrence across compositionality benchmarks. *arXiv preprint arXiv:2310.17514*, 2023.

[32] 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. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

[33] Vijay Viswanathan, Chenyang Zhao, Amanda Bertsch, Tongshuang Sherry Wu, and Graham Neubig. Prompt2model: Generating deployable models from natural language instructions. *ArXiv*, abs/2308.12261, 2023. URL https://api.semanticscholar.org/CorpusID:261075905.

[34] Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. *ArXiv*, abs/2309.08638, 2023. URL https://api.semanticscholar.org/CorpusID:262045288.

[35] Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych. Holmes: Benchmark the linguistic competence of language models, 2024.

[36] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019.

[37] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Haotong Zhang, Joseph Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685, 2023. URL https://api.semanticscholar.org/CorpusID:259129398.

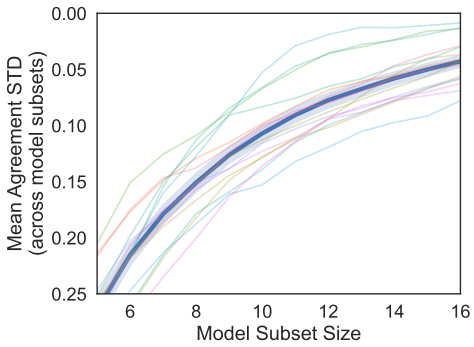
[38] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied Sanosi Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *ArXiv*, abs/2304.06364, 2023. URL https://api.semanticscholar.org/CorpusID:258108259.

6 Appendices

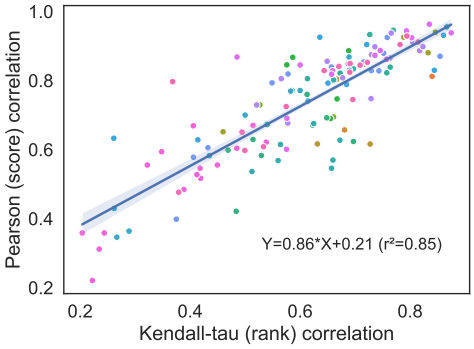
6.1 Links

Leaderboard: `https://huggingface.co/spaces/anonymous-org-123/benchbench` Pacakge: `https://anonymous.4open.science/r/benchbench-ARR`

6.2 Number of models and Metric matters



(a) **Agreement variance is inversely related to model subset size.** The mean standard deviation of the Kendall-tau correlations arising from performing `BAT` using different randomly sampled model subsets. The blue line represents the benchmark mean while the other ones are for the benchmarks listed in App 7.6.



(b) **Agreement measures are linearly depended but biased.** The Kendall-tau and Pearson correlation of all benchmark pairs show a strong linear dependence, and a bias factor of 0.21. Colors represent the different benchmarks listed in App 7.6.

6.3 Open Questions

What do we make of high agreement? It is not trivial how one should treat two benchmarks that are in high agreement with each other. If one is more convenient to run (e.g., doesn't require costly metrics), then from a practical perspective, a user can simply choose it over the more expensive one. However, practitioners and researchers must not confuse high agreement with the notion that the benchmarks actually measure the exact same qualities. Among other things, this could lead to the erroneous conclusion that new benchmarks are no longer needed, impeding new benchmark development. The community must also discriminate between correlations of model abilities (strong models are strong at many tasks) and correlations of the benchmarks themselves (the benchmarks actually measure the same qualities).

What do we make of low agreement? Reliability concerns the consistency of benchmark results. In this paper, we accept the benchmark scores as presented and focus on their benchmark validity, which assesses whether benchmarks accurately measure what they purport to evaluate. However, this ignores the *reliability* issues within the benchmarks, which place an upper bound on the level of benchmark agreement. If, for instance, a benchmark cannot reliably differentiate between its top-3 models, then naturally we do not expect to see agreement over the top-3 models with other benchmarks. Looking forward, methodological improvements in `BAT` must include incorporating reliability measures, allowing to decouple disagreements from low reliability.

How do we use `BAT` to retire benchmarks? Another point concerns the role of `BAT` for benchmark retirement, i.e., at what point do we decide that an old benchmark is no longer relevant and should be discarded. Currently the issue of retirement is viewed mainly from the perspective of saturation, where the community stops using benchmarks on which all new models succeed. However, another reason to retire benchmarks may be that the mixture of abilities models are expected to possess has shifted over time. In this scenario, `BAT` can reveal that a certain benchmark is no longer viable.

7 Limitations

We note that finding low agreement may indicate one of two issues, both of which have negative implications. These issues should be addressed or interpreted differently. One option is that the benchmark measures something different from what it is supposed to and is hence not valid. That is the more common interpretation and calls for changes. Another option might be that the benchmark is just not reliable, intuitively its ranking is unstable and did not converge. In such cases, even the same benchmark may not agree with itself given small changes (subsets, seeds etc.), this usually calls for evaluating on more examples [6] or configuration [1]. There is a positive note to the same story, if a benchmark already shows a strong BAT in fine-grained evaluation (e.g., 5 models close to each other), it also means that it is quite reliable.

Sometimes BAT is not needed. BAT gives a way to validate a benchmark by an external source of authority. However, other methods or other sources for authority (e.g., being masterfully crafted by experts) might give stronger signals. Especially in the case of new and unique signals that can mostly show they are different, but not that they are valid for their own unique purpose.

In general, BAT needs a reference benchmark, or ideally multiple benchmarks that provide diverse measurements of the same construct. Still, choosing the right reference benchmarks might be tricky, and the results might be sensitive to this choice.

7.1 BAT uses in Related Work

While some examples were given in the text, we elaborate on a handful of works employing BAT.

Some works survey and analyze a field by utilizing BAT techniques. Liu et al. [21] check agreement across many QA datasets and conclude that since agreement is high, there is no need for more QA datasets. Sun et al. [31] use correlations to show that Compositionality Benchmarks do not agree amongst themselves. They used Kendall-Tau and set 0.7 as the high agreement threshold. Other works performed general efficient evaluation research and utilized BAT [28, 26, 27, 33]. All of these works performed a thoughtful evaluation and large (reliable) rank correlation over all the models in the benchmarks. However, they did not consider the high correlations achieved in such settings (§3.2).

Other work relies on BAT to compare to a specific benchmark. Lei et al. [15] and Viswanathan et al. [33] both propose a synthetic benchmark as a proxy and show good agreement with the original benchmark, although they differ in their methodology. Chang et al. [4] propose two benchmarks and use agreement to show that they capture the same phenomenon, and Mizrahi et al. [22] test agreement within the same benchmarks using different prompts. Li et al. [17] validate a new benchmark with 6 models of 3 sizes 7B,13B,33B with agreement alpaca(v2) [18]. [35] show divergent validity by comparing their benchmark to established ones, showing low BAT scores. Lastly, [26] compared efficient versions of the HELM benchmark to the full one.

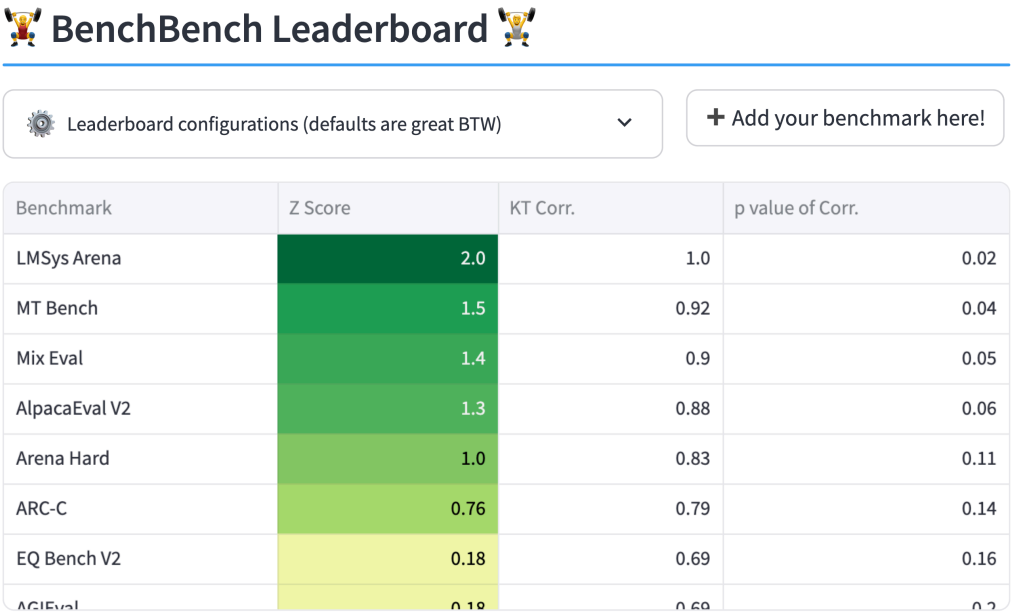
7.2 BenchBench - a Package and Leaderboard

We introduce **BenchBench**, a package implementing the above guidelines - standardizing the practice of BAT – and holding results of multiple benchmarks for a wide variety of reference benchmark choices. The python package is available in GitHub at: `github.com/IBM/benchbench`.

The workflow of using the package is as follows:

1. A user enters their BAT configuration, including the desired group of reference benchmarks.
2. BenchBench recommends a set of models for evaluation on the target benchmark.
3. The user inputs their benchmark results for the recommended models.
4. BenchBench produces a full BAT report.

In the default functionality, BenchBench expects a list of model scores over the target benchmark, as well as a desired group of reference benchmarks to compare to. It also offers the functionality of proposing a minimal set of models for evaluation, ensuring fair and unbiased comparisons. While offering flexibility to change the defaults, BenchBench's BAT report includes several granularities of models. BenchBench standardizes arbitrary decisions that hinder reproducibility, following the best



Benchmark	Z Score	KT Corr.	p value of Corr.
LMSys Arena	2.0	1.0	0.02
MT Bench	1.5	0.92	0.04
Mix Eval	1.4	0.9	0.05
AlpacaEval V2	1.3	0.88	0.06
Arena Hard	1.0	0.83	0.11
ARC-C	0.76	0.79	0.14
EQ Bench V2	0.18	0.69	0.16
AGIEval	0.18	0.69	0.2

Figure 4: **The `BenchBench`-leaderboard - A meta-benchmark for `BAT`.** The following leaderboard is obtained with the default configurations, using the aggregate of all holistic reference benchmarks as the reference benchmarks and comparing subsets of 20 models that were sampled randomly. As more benchmarks are added to Holistic set, results may be different upon view.

Table 1: **Our recommendations substantially reduce the variance of `BAT`.** Ablation analysis for each `BAT` recommendation separately and their combination. It shows great gains in using our methodologies when running `BAT` both separately and combined.

Recommendations			BAT Variance		Section Ref.
Aggregate References	Select Metric	Select Models	σ (↓)	Reduction	
			0.31	-	-
X			0.23	-30%	§3.1
	X		0.23	-30%	§3.3
		X	0.20	-35%	§3.2
X	X	X	0.10	-67%	§4

practices proposed here. Lastly, `BenchBench` offers the user to upload their benchmark results to the `BenchBench` database, enriching the reference benchmark distribution for future efforts, thereby enhancing `BAT` reliability without additional computational costs. due to running additional reference benchmarks.

We propose the **`BenchBench`-leaderboard**, a new leaderboard designed to rank benchmarks according to their agreement to a desired group of reference benchmarks (see Figure 4). To do so `BenchBench` ranks all submitted benchmarks by comparable standards.

Since the `BenchBench`-leaderboard is build on top of the `BenchBench` package, new benchmarks uploaded to the package will be added to the leaderboard as well. Thus, the benchmark will improve with time, taking into account novel benchmarks and measured model traits.

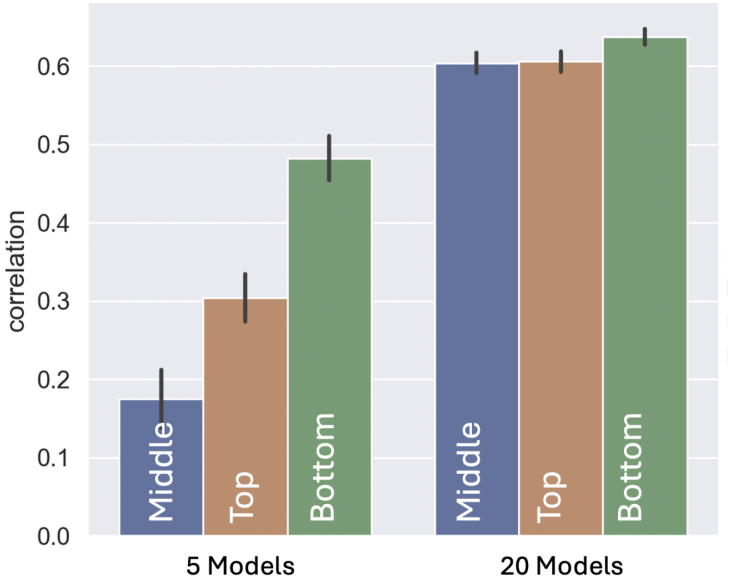


Figure 5: **Correlation as a function of model subset size:** Correlations substantially decline as the models considered are closer to the top, error bars are the SEMs across the different pairs of benchmarks

7.3 Ablations Table

7.4 Benchmarks used

The **AGI Eval** [38] benchmark assesses models on human-level cognition and problem-solving tasks, which tests the real-world applicability of model outputs. Similarly, **Alpaca (v2)** [18] and its **length-adjusted version** [10] focus on a model's ability to follow complex instructions with the latter specifically addressing biases associated with output length.

HumanEval [5] presents code generation challenges, evaluating the syntactic correctness and logical soundness of model-generated code. Alongside, the **HuggingFace OpenLLM Leaderboard** [2] employs the Eleuther AI Evaluation Harness [11] to test models on several key benchmarks such as ARC [7], HellaSwag [36], MMLU [13], TruthfulQA [20], Winogrande [30], and GSM8k [8]. **EQ-Bench (v2)** [24], measures the emotional intelligence of models, essential for applications that involve nuanced human interactions.

The **MAGI** [23] benchmark integrates challenging elements from MMLU and AGIEval to test complex reasoning and problem-solving capabilities of models. It is particularly effective in highlighting subtle performance differences among top-tier models. **MMLU** [12] assesses both general and specialized knowledge across various domains, providing a broad evaluation spectrum.

Further, benchmarks like **Chatbot-Arena and MTBench** [37] focus on multi-turn conversation abilities, crucial for applications in customer service and virtual assistance. Lastly, **Big Bench Hard** [32] challenges models with complex text understanding and generation, pushing the limits of what natural language processing technologies can achieve. It is worth noting, that the HELM benchmark [19] was excluded from our analysis because there were few overlapping models with the other benchmarks.

7.5 Model Tier

Building on the importance of model proximity, another crucial factor in benchmark agreement is the tier of models being assessed. Current BAT practices often treat benchmarks as a uniform slab, disregarding the variations across different tiers of model performance. However, agreement might

not be uniform across these tiers, and understanding this variance can provide deeper insights into benchmark reliability and model performance.

In Figure 5, we show that model tier significantly impacts benchmark agreement. Bottom-tier models exhibit higher agreement among themselves, with Kendall correlation coefficients just below 0.5. In contrast, middle-tier models show low agreement (coefficients below 0.2), and top-tier models demonstrate low to medium agreement (around 0.3).

One potential explanation for this phenomenon is the (lack of) reliability of the benchmark, as discussed in the introduction and literature [26]. Figure 5 highlights that the standard deviation of scores bottom-ranked models is significantly higher than the rest. This might mean that there is some effect the goes beyond granularity or density, with older models being easier to differentiate (and gaining higher correlations to the models). However middle and top ranked models do not show such a trend (even when taking into account that middle granularity is higher as top models are still joining the game), which means that no strong conclusion should be made excluding older models, switching benchmarks frequently or similar actions, at most, old models may be left out of BAT, but other effects seem more pressing.

7.6 Benchmark used for visualizations

The benchmarks we used include: AGI Eval [38], Alpaca (v2) [18], and its length-adjusted version [10], HuggingFace OpenLLM Leaderboard [2], MMLU [12], Chatbot-Arena and MTBench [37], Big Bench Hard [32]. ARC [7], HellaSwag [36], TruthfulQA [20], Winogrande [29], EQ-Bench (v2) [24]. All benchmarks have a permissive license that allows academic use.