Do These LLM Benchmarks Agree? Fixing Benchmark Agreement Evaluation with BenchBench

Anonymous ACL submission

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. This deficiency can lead to invalid conclusions, fostering mistrust in benchmarks and upending the ability to choose the appropriate benchmark. By analyzing over 50 prominent benchmarks, we demonstrate how some overlooked methodological choices can significantly influence BAT results, potentially undermining the validity of conclusions. To address these inconsistencies, we propose a set of best practices for BAT and demonstrate how utilizing these methodologies greatly improves BAT robustness and validity. To foster adoption and facilitate future research, we introduce BenchBench[1], a Python package for BAT, and release the BenchBench-leaderboard[2], a meta-benchmark designed to evaluate benchmarks using their peers.

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.

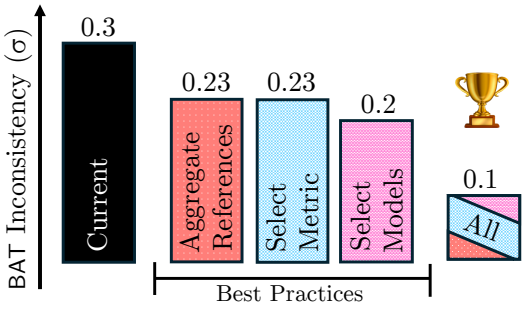

Figure 1: **Running BAT using our best practices increases consistency by 3x.** The average standard deviation of BAT results over multiple instances is drastically decreased using our best practices, without incurring further computational costs. These best practices can be easily applied using our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

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. (Lei et al., 2023; Viswanathan et al., 2023; Chang et al., 2023; Li et al., 2024b; Prabhu et al., 2024; He et al., 2024). 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 (Perlitz et al., 2023; Polo et al., 2024; Prabhu et al., 2024; Vivek et al., 2023). 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 (Yuan et al., 2024; Waldis et al., 2024). The above goals are relevant both for benchmark creators and for benchmark consumers. Creators will

[1] https://bit.ly/benchbench
[2] https://bit.ly/benchbench-leaderboard

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 50 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).

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 Figure 2, 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 upon our findings, we compile a set of best practices for BAT (§4) and demonstrate their impact (see Figure 1 and Table 1). To foster adoption and promote reproducibility, we have implemented these guidelines into *BenchBench*, a Python package for BAT (§5). BenchBench supplies users not only with a framework but also with the data needed to perform BAT, relieving users of the computational and time burden of gathering multiple benchmarks for comparison. Notably, when using BenchBench, applying our best practices for running BAT will not require further computational resources. Furthermore, BenchBench is built to continually evolve, allowing easy addition of new benchmarks.

Lastly (§5), we introduce the BenchBench-Leaderboard. Using BenchBench as its back-end, the BenchBench-Leaderboard is a dynamic leaderboard that provides easy access to BAT results for established benchmarks. By ranking benchmarks based on their agreement with the user's desired set of reference benchmarks, the BenchBench-Leaderboard facilitates making informed evaluation decisions.

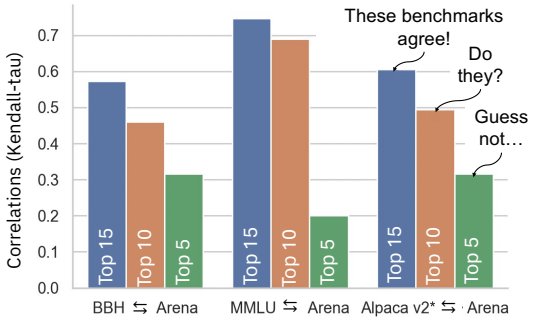

Figure 2: **BAT Conclusions depend on the models considered**. Kendall-tau correlations between the LMSys Arena benchmark and three other benchmarks: BBH, MMLU, and Alpaca v2. Each group of bars represents the correlation for different sets of top models, specifically the top 5, top 10, and top 15 (overlapping) models (according to the Arena). The results indicate that the degree of agreement between benchmarks varies with the number of top models considered, highlighting that different selections of models can lead to varying conclusions about benchmark agreement.

To sum up, our contributions are as follows:

1. We perform a large-scale analysis of benchmark agreement, highlighting the impact of several crucial methodological decisions (§3).

2. We propose guidelines for reliable and standardized BAT (§4) and demonstrate their impact.

3. We release BenchBench, a Python package for BAT implementing the guidelines and incorporating them with the required benchmark data (§5).

4. We harness BenchBench as the back-end for a new meta-benchmark (§5).

2 Setup

For our analysis, we use over 40 benchmarks, with their results cutoff at Jan 2024. The benchmarks we used include: AGI Eval (Zhong et al., 2023), Alpaca (v2) (Li et al., 2023), and its length-adjusted version (Dubois et al., 2024), HuggingFace OpenLLM Leaderboard (Beeching et al., 2023), MMLU (Hendrycks et al., 2020), MAGI (Paech, 2024), Chatbot-Arena and MT-Bench (Zheng et al., 2023), Big Bench Hard (Suzgun et al., 2022). HumanEval (Chen et al., 2021)

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: Aggregate References	Recommendations: Select Metric	Recommendations: Select Models	BAT Variance: σ (↓)	BAT Variance: Reduction	Section Ref.
			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

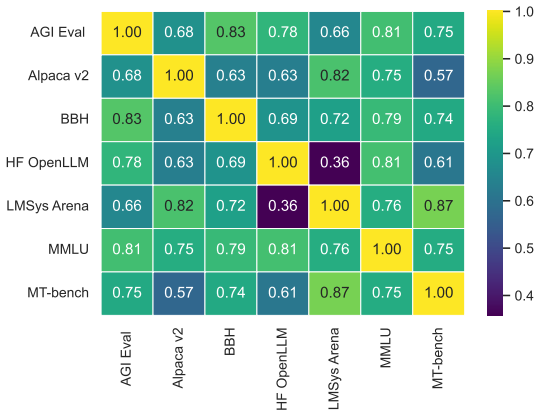

Figure 3: **Agreement scores significantly vary across different appropriate reference benchmarks**. Kendall-tau correlations between pairs of benchmarks that are seemingly valid for BAT. Each is taken over 20 models sampled at random.

ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), TruthfulQA (Lin et al., 2022), Winogrande (Sakaguchi et al., 2019), GSM8k (Cobbe et al., 2021a). EQ-Bench (v2) (Paech, 2023), ArenaHard (Li et al., 2024a) and OpenCompass (Contributors, 2023). For a wider survey of benchmarks used, see App. 9.1.

Our analysis focuses 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 (Kendall, 1938)) or scores (using Pearson (Pearson, 1895)).

We note that an inherent constraint in BAT is the number of intersecting models between the benchmarks (i.e., models appearing in both benchmarks). Benchmarks lacking a sufficiently large set of intersecting models (for this work, we chose ≥ 5), cannot be reliably used for BAT.

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. Next, we highlight how commonly reported BAT results can foster false expectations among benchmark consumers.

3.1 The Choice of Reference Benchmark Matters

Finding a reference benchmark for BAT is a non-trivial task. One needs to find a well-established benchmark, whose data is readily available, and which exhibits a large enough overlap with the models already evaluated in the target benchmark. Due to the above difficulty, BAT is commonly done against one or two reference benchmarks (Yuan et al., 2024). Benchmarks can be divided into groups according to their measured abilities – for example, holistic benchmarks that aim to measure some loosely-defined construct of *overall* model quality, such as BigBench (bench authors, 2023), benchmarks measuring coding abilities (Chen et al., 2021), math benchmarks (Cobbe et al., 2021a), etc. Thus, when selecting a reference benchmark, there is often a somewhat arbitrary choice between several possible benchmarks which are all seemingly appropriate.

Figure 3 illustrates the variability caused by such arbitrary choices: for each target benchmark, different reference benchmarks produce wildly varying agreement scores. For example, Alpaca V2 (second row from above) demonstrates a wide range of agreement levels with other benchmarks, spanning from a mediocre agreement of 0.57 with MT-bench to a high agreement of 0.82 with LMSys Arena, even though both of these reference benchmarks are considered to measure similar abilities. This variability calls into question the validity of conclusions based on applying BAT when relying on a single reference benchmark.

To address this issue, we advocate using an aggregated reference benchmark that consolidates results of multiple benchmarks based on the mean-win-rate; see more on this in §4.

3.2 The Choice of Models Matters

In performing BAT, one measures some agreement metric over the scores of a group of models overlapping between the target and reference benchmark. Typically, authors arbitrarily pick some small set of models for their analysis. However, as we detail below, both the quantity and the properties of the selected models should be taken into account when drawing conclusions from BAT.

The Number of Compared Models Matters Figure 5 illustrates the relationship between the number of models and the variability of BAT results. It shows that with a small amount of models, BAT results can get highly unreliable, with a standard deviation approaching 0.25. For instance, in our analysis we found that the Kendall-tau correlation between LMSysArena and MT-Bench can range from approximately 0.65 to 0.99, depending on the particular number of models chosen. Thus, we see that the common practice of using a small number of models for BAT may jeopardise the validity of conclusions.

Granularity Matters Performing BAT produces a score that indicates high or low agreement. However, the meaning of this score will differ depending on the models included in the analysis. For example, as seen in Figure 2, for a given pair of benchmarks, the agreement obtained over similarly strong models will generally be lower than over a set of models of varying qualities.

To quantify this phenomenon, we investigate benchmark agreement where the subset of models selected is not completely random, but is constrained to sets of models that are adjacent in rank (e.g., models 3-7)[3]. Adjacent models have more similar performance. Thus, their score differences and ranking may be less stable, resulting in lower correlation scores. In Figure 4, we show that indeed, for a given number of models, the correlation score when considering adjacent models is lower than that of randomly sampled models, with a stronger effect as the number of models in the subset decreases.

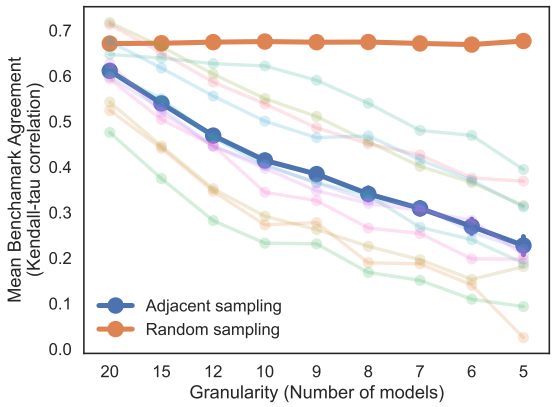

Figure 4: **Agreement is lower for closely ranked models.** Mean correlation (y) between each benchmark (lines) and the rest, given different numbers of models. The Blue and Orange lines are the average of all benchmark pair correlations with models sampled randomly (orange) or in contiguous sets (blue). The shaded lines represents adjacent sampling for the the set of benchmarks listed in App 9.3.

This discrepancy emphasizes the importance of reporting BAT scores at multiple levels of granularity. This would enable managing the expectations of benchmark consumers, who may expect and desire a specific level of granularity (e.g., getting the very best models right, or discriminating between strong and weak models).

3.3 The Choice of Correlation Metric (and Threshold) Matters

BAT is the process of measuring correlations of model scores (or ranks) between two benchmarks. Once a correlation score is obtained, this score is commonly interpreted based on how it compares to some threshold; surpassing the threshold means the agreement is considered "high", while falling below it means the agreement is "low".

Currently, there are no consistent standards for the types and thresholds of correlation metrics. For instance, Liu et al. (2021) utilized both rank and score correlations, setting a uniform threshold of 0.8 for both, whereas Sun et al. (2023) exclusively employed rank correlation and opted for a distinct threshold of 0.7.

To improve our understanding on the significance of these choices, we analyse the relationship between rank (Kendall-tau) and score (Pearson) correlation metrics. In Figure 6 we present correla-

[3]Note that the sets of adjacent models were not selected from a specific rank location (e.g., Top, Bottom, Middle) but were randomly selected from the full range. For an analysis of such location-dependent sets, see App 9.2.

tion scores between different pairs of benchmarks with varying model subsets. We observe a strong linear relationship ($r^2 = 0.85$) between the two correlation functions, indicating that they exhibit similar behavior in measuring agreement. However, the figure also shows a consistent score difference of approximately 0.2 between the two metrics, indicating a potential flaw in the current practice of applying the same threshold regardless of the metric chosen. This underscores the necessity for a data-driven approach – comparative in nature – to interpret correlation scores; see §4 for more details.

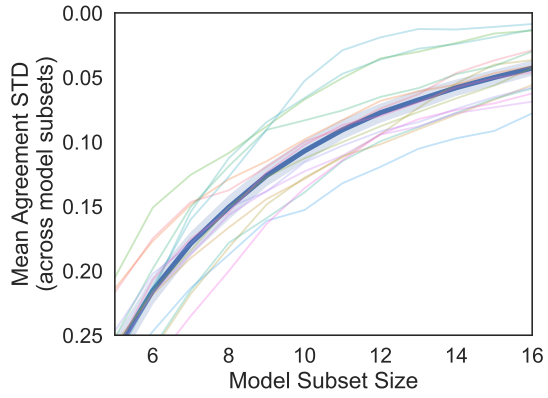

Figure 5: **Agreement variance is inversely related to model subset size.** The mean standard deviation of the Kendall-tau correlations arising from performing BAT using different randomly sampled model subsets. The blue line represents the benchmark mean while the other ones are for the benchmarks listed in App 9.3.

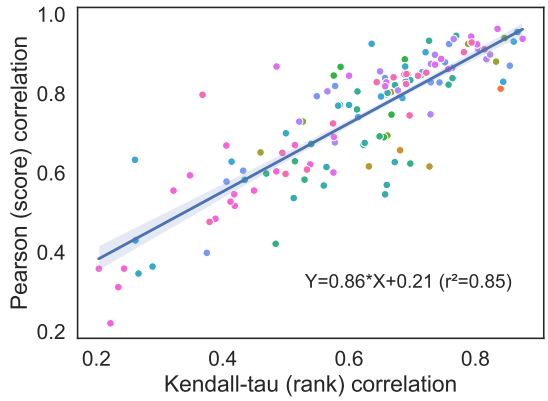

Figure 6: **Agreement measures are linearly depended but biased.** The Kendall-tau and Pearson correlation of all benchmark pairs show a strong linear dependence, and a bias factor of 0.21. Colors represent the different benchmarks listed in App 9.3.

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, Figure 3). 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. For example, when using BAT to validate some efficient holistic benchmark, the reference benchmark should be the aggregate of all available holistic benchmarks. By combining results from a group of benchmarks, the aggregate benchmark provides both a more stable and robust basis for comparison. Notably, since the aggregate benchmark captures the distribution of relevant results, it constitutes a better measure of the underlying construct represented by the group, called in the literature convergent validity (Carlson and Herdman, 2012).

Measuring the effect of such methodology, in Table 1, we compare the standard deviation of BAT correlation results when using arbitrary reference benchmarks (first line) to that when using the aggregate, it shows that the standard deviation of the correlation drops with our recommendation by more that 30%.

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, Figure 4). 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. The steps of this approach are as follows:

1. **Compile a Distribution:** Begin by compiling a distribution of agreement scores from various benchmarks relative to the chosen reference benchmark.

2. **Calculate the Target Benchmark's Z-Score:**

Next, compare the target benchmark's correlation score to this distribution by calculating its Z-score. Indicating how the target benchmark's agreement compares to that of other benchmarks.

3. **Interpret the Z-Score:** Benchmarks with a Z-score above -1σ are considered to be in agreement with the reference; those below this threshold are not.

By incorporating the natural distribution of benchmark agreement scores, this method ensures that the assessment of agreement is both context-sensitive and adaptive to changes in the benchmark landscape. Furthermore, as more benchmarks are added, the distribution is updated, making the test increasingly reflective of the current landscape of benchmarks measuring the desired trait.

Use More Models and Sample Them Randomly BAT based on a small set of models tends to have high variance, as shown in Figure 5, where the standard deviation of results can reach 0.25 with fewer models (§3.2). To reduce this variability and enhance reliability, we recommend using at least 10 models, preferably more. A larger and more diverse sample provides a more representative evaluation, minimizing bias and improving result stability. While increasing the number of models does raise computational costs, our recommendation remains practical, given that most model benchmarks already evaluate a larger number of models. These models should represent the entire spectrum of available models, including diverse sizes, architectures, and training methods. Aiming for a random selection ensures equal representation and minimizes bias. Table 1 shows that using this methodology to select models decreases BAT variance by more than 30%.

Report Multiple Granularities Benchmark agreement varies significantly with the range of model qualities considered, as demonstrated in Figure 2 (§3.2). For instance, agreement can be high across a broad range of models but low among top-ranked models, which can mislead benchmark consumers who seek fine-grained distinctions. To address this, we recommend reporting agreement scores at multiple resolutions (e.g., 5/10/20 contiguous models, averaging across groups when more models were sampled). This practice provides a more nuanced and complete picture, allowing users to make informed decisions based on their specific needs. This approach provides a more nuanced view of benchmark agreement, highlighting critical distinctions that might otherwise be missed (e.g. the top 3 models are almost never in agreement across benchmarks).

Follow The Above Rules! Properly performing BAT using the above guidelines is not a trivial task. These methodologies require complex statistical tools, reproducible analysis and mostly, access to a large amount of up-to-date benchmarks data. Recognizing this difficulty, we have implemented our recommended workflow into BenchBench, a Python package for BAT, described below.

Making the case for our above recommendations, Table 1 demonstrates the significant gains obtained when using our methodological choices to perform BAT. It shows not only that the different recommendations each have an impact on variance, but also that their effect can be combined to achieve a substantially lower variance point – reducing the standard deviation by $\sim 67\%$, and thereby delivering far more robust BAT results.

5 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 varity of reference benchmark choices. The python package is available in GitHub at: https://bit.ly/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,

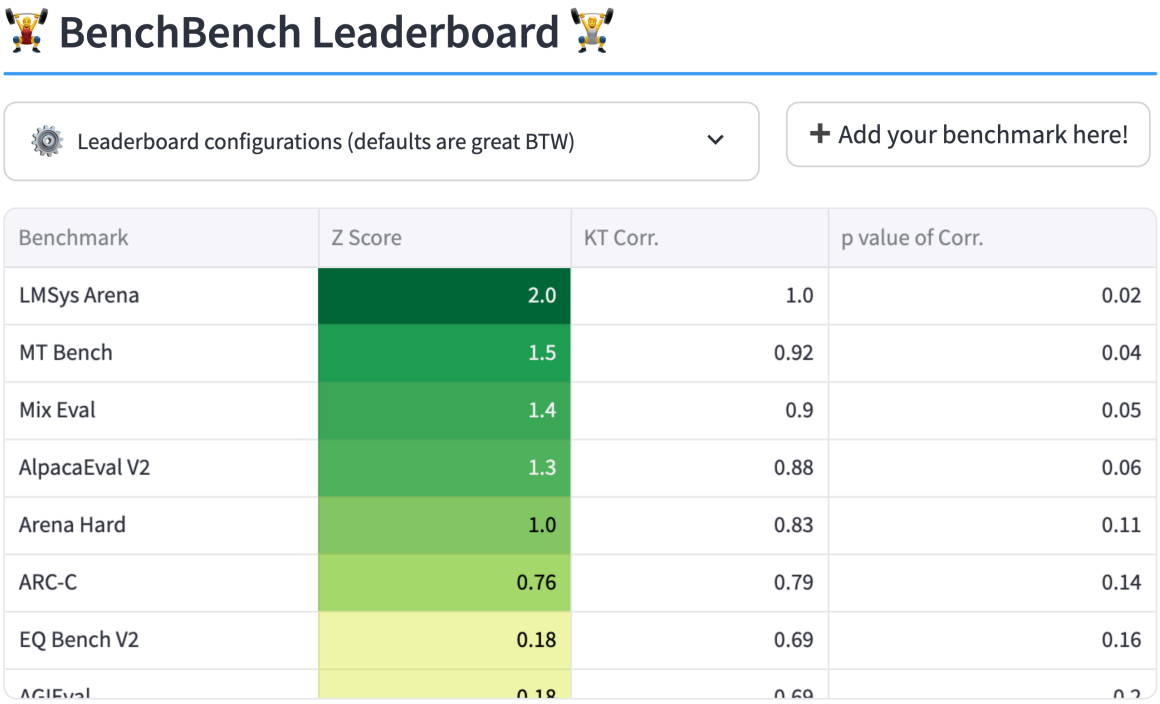

Figure 7: **The BenchBench-leaderboard - A meta-benchmark for BAT.** The following leaderboard is obtained with the default configurations, using the aggregate of all holistic reference benchmarks as the reference benchmarks and comparing subsets of 20 models that were sampled randomly. As more benchmarks are added to Holistic set, results may be different upon view.

BenchBench's BAT report includes several granularities of models. BenchBench standardizes arbitrary decisions that hinder reproducibility, following the best 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 7). 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.

6 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. (2021) check agreement across many QA datasets and conclude that since agreement is high, there is no need for more QA datasets. Sun et al. (2023) 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 (Prabhu et al., 2024; Perlitz et al., 2023; Polo et al., 2024; Viswanathan et al., 2023). 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. Feng et al. (2024) automatically sample a small set of instructions as an efficient LLM benchmark, reducing human labor significantly. They show this still agrees with existing benchmarks. Similarly, Lei et al. (2023) and Viswanathan et al. (2023) 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. (2023) propose two benchmarks

and use agreement to show that they capture the same phenomenon, and Mizrahi et al. (2023) test agreement within the same benchmarks using different prompts. Li et al. (2024b) validate a new benchmark with 6 models of 3 sizes 7B,13B,33B with agreement alpaca(v2) (Li et al., 2023). Yuan et al. (2024) and (Waldis et al., 2024) show divergent validity by comparing their benchmark to established ones, showing low BAT scores. Lastly, (Perlitz et al., 2023) compared efficient versions of the HELM benchmark to the full one.

7 Discussion and Conclusions

In this work, we shine a light on the lack of consistent BAT methodology. We analyze several BAT choices on a broad spectrum of benchmarks and assess their effect. Our analysis shows that different choices of (1) Models (2) Reference Benchmark(s), and (3) Thresholding scheme, can significantly alter BAT conclusions. Therefore, we advise a set of best practices and provide a Python package that aims to facilitate a consistent BAT process in the community. We also release the BenchBench-leaderboard, a benchmark that quantifies the agreement of a benchmark with an aggregate of existing benchmarks.

In this paper, our focus was on the methodological issues when performing BAT. We did not deal with questions regarding when BAT should be used, and how conclusions from BAT should be interpreted. Next, we describe several such 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.

In conclusion, our study enhances the precision and reliability of Benchmark Agreement Testing by establishing best practices and introducing the BenchBench Python package and leaderboard. These contributions foster standardized evaluations, enabling more accurate comparisons across benchmarks and setting a new direction for computational linguistics research.

8 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 (Choshen et al., 2024) or configuration (Bandel et al., 2024). 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.

References

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. 2024. Unitxt: Flexible, shareable and reusable data preparation and evaluation for generative ai.

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

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

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

Ting-Yun Chang, Jesse Thomason, and Robin Jia. 2023. Do localization methods actually localize memorized data in llms? *ArXiv*, abs/2311.09060.

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. 2021. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374.

Leshem Choshen, Ariel Gera, Yotam Perlitz, Michal Shmueli-Scheuer, and Gabriel Stanovsky. 2024. Navigating the modern evaluation landscape: Considerations in benchmarks and frameworks for large language models (LLMs). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024): Tutorial Summaries*, pages 19–25, Torino, Italia. ELRA and ICCL.

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

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. 2021a. Training verifiers to solve math word problems.

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. 2021b. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

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

Kehua Feng, Keyan Ding, Kede Ma, Zhihua Wang, Qiang Zhang, and Huajun Chen. 2024. Sample-efficient human evaluation of large language models via maximum discrepancy competition.

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. 2021. A framework for few-shot language model evaluation.

Chaoqun He, Renjie Luo, Shengding Hu, Yuanqian Zhao, Jie Zhou, Hanghao Wu, Jiajie Zhang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Ultraeval: A lightweight platform for flexible and comprehensive evaluation for llms.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

M. G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30:81–93.

Fangyu Lei, Qian Liu, Yiming Huang, Shizhu He, Jun Zhao, and Kang Liu. 2023. S3eval: A synthetic, scalable, systematic evaluation suite for large language models. *ArXiv*, abs/2310.15147.

Tianle Li, Wei-Lin Chiang, Evan Frick, Dunlap Lisa, Zhu Banghua, Gonzalez Joseph E., and Ion Stoica. 2024a. From live data to high-quality benchmarks: The arena-hard pipeline.

Xiang Li, Yunshi Lan, and Chao Yang. 2024b. Treeeval: Benchmark-free evaluation of large language models through tree planning. *ArXiv*, abs/2402.13125.

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

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. 2023. Holistic evaluation of language models.

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

Nelson F. Liu, Tony Lee, Robin Jia, and Percy Liang. 2021. Do question answering modeling improvements hold across benchmarks? In *Annual Meeting of the Association for Computational Linguistics*.

Moran Mizrahi, Guy Kaplan, Daniel Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2023. State of what art? a call for multi-prompt llm evaluation. *ArXiv*, abs/2401.00595.

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

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

Karl Pearson. 1895. Vii. note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240 – 242.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. 2023. Efficient benchmarking (of language models). *ArXiv*, abs/2308.11696.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. tinybenchmarks: evaluating llms with fewer examples. *ArXiv*, abs/2402.14992.

Ameya Prabhu, Vishaal Udandarao, Philip H.S. Torr, Matthias Bethge, Adel Bibi, and Samuel Albanie. 2024. Lifelong benchmarks: Efficient model evaluation in an era of rapid progress. *ArXiv*, abs/2402.19472.

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

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

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

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. *arXiv preprint arXiv:2210.09261*.

Vijay Viswanathan, Chenyang Zhao, Amanda Bertsch, Tongshuang Sherry Wu, and Graham Neubig. 2023. Prompt2model: Generating deployable models from natural language instructions. *ArXiv*, abs/2308.12261.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. 2023. Anchor points: Benchmarking models with much fewer examples. *ArXiv*, abs/2309.08638.

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

Moy Yuan, Chenxi Whitehouse, Eric Chamoun, Rami Aly, and Andreas Vlachos. 2024. Probelm: Plausibility ranking evaluation for language models.

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

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. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied Sanosi Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *ArXiv*, abs/2304.06364.

9 Appendices

9.1 Benchmarks used

The **AGI Eval** (Zhong et al., 2023) benchmark assesses models on human-level cognition and problem-solving tasks, which tests the real-world applicability of model outputs. Similarly, **Alpaca (v2)** (Li et al., 2023) and its **length-adjusted version** (Dubois et al., 2024) focus on a model's ability to follow complex instructions with the latter specifically addressing biases associated with output length.

HumanEval (Chen et al., 2021) presents code generation challenges, evaluating the syntactic correctness and logical soundness of model-generated code. Alongside, the **HuggingFace OpenLLM Leaderboard** (Beeching et al., 2023) employs the Eleuther AI Evaluation Harness (Gao et al., 2021) to test models on several key benchmarks such as ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), TruthfulQA (Lin et al., 2022), Winogrande (Sakaguchi et al., 2021), and GSM8k (Cobbe et al., 2021b). **EQ-Bench (v2)** (Paech, 2023), measures the emotional intelligence of models, essential for applications that involve nuanced human interactions.

The **MAGI** (Paech, 2024) 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** (Hendrycks et al., 2020) assesses both general and specialized knowledge across various domains, providing a broad evaluation spectrum.

Further, benchmarks like **Chatbot-Arena and MTBench** (Zheng et al., 2023) focus on multi-turn conversation abilities, crucial for applications in customer service and virtual assistance. Lastly, **Big Bench Hard** (Suzgun et al., 2022) 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 (Liang et al., 2023) was excluded from our analysis because there were few overlapping models with the other benchmarks.

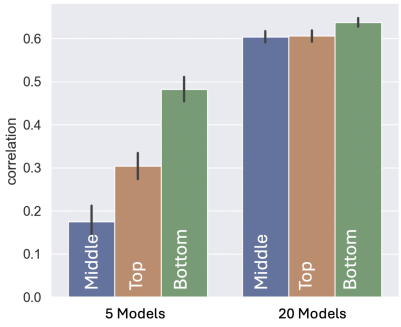

Figure 8: **Correlation as a function of model subset size:** Correlations substantially decline as the models considered are closer to the top, error bars are the SEMs across the different pairs of benchmarks

9.2 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 8, 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 (Perlitz et al., 2023). Figure 8 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.

9.3 Benchmark used for visualizations

The benchmarks we used include: AGI Eval (Zhong et al., 2023), Alpaca (v2) (Li et al., 2023), and its length-adjusted version (Dubois et al., 2024), HuggingFace OpenLLM Leaderboard (Beeching et al., 2023), MMLU (Hendrycks et al., 2020), Chatbot-Arena and MTBench (Zheng et al., 2023), Big Bench Hard (Suzgun et al., 2022). ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), TruthfulQA (Lin et al., 2022), Winogrande (Sakaguchi et al., 2019), EQ-Bench (v2) (Paech, 2023). All benchmarks have a permissive license that allows academic use.