LLM-Powered Benchmark Factory: Reliable, Generic, and Efficient

Anonymous Authors[1]

Abstract

The rapid advancement of large language models (LLMs) has led to a surge in both model supply and application demands. To facilitate effective matching between them, reliable, generic and efficient benchmark generators are widely needed. However, human annotators are constrained by inefficiency, and current LLM benchmark generators not only lack generalizability but also struggle with limited reliability, as they lack a comprehensive evaluation framework for validation and optimization. To fill this gap, we first propose an automated and unbiased evaluation framework, structured around four dimensions and ten criteria. Under this framework, we carefully analyze the advantages and weaknesses of directly prompting LLMs as generic benchmark generators. To enhance the reliability, we introduce a series of methods to address the identified weaknesses and integrate them as BENCHMAKER. Experiments across multiple LLMs and tasks confirm that BENCHMAKER achieves superior or comparable performance to human-annotated benchmarks on all metrics, highlighting its generalizability and reliability. More importantly, it delivers highly consistent evaluation results across 12 LLMs (0.967 Pearson correlation against MMLU-Pro), while taking only $0.005 and 0.38 minutes per sample.

1. Introduction

With the ongoing scaling up of large language models (LLMs) in multiple dimensions over the past few years, two key trends have emerged (Figure 1): (1) The LLM release process has accelerated and now exceeds 30k per season; (2) The growth in LLM capabilities has spurred application demand, reflected in over 50M downloads of open-source models per season. Serving as a bridge between massive LLM supply and various application needs, the demand for customized benchmarks is rapidly growing, helping downstream tasks identify the most suitable LLM.

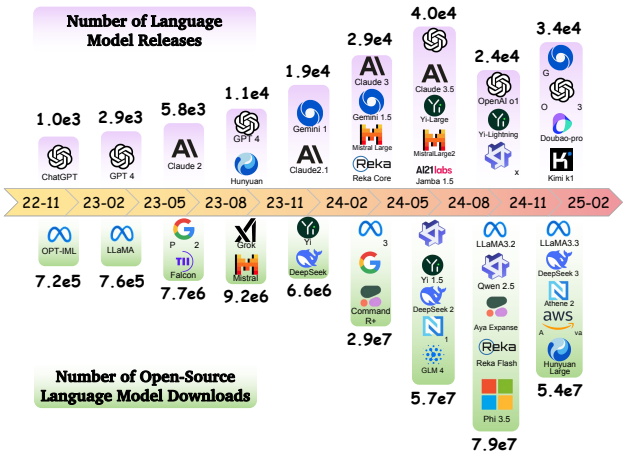

Figure 1. The trends of LLMs released and open-source LLMs downloads per season since the debut of ChatGPT. We obtain the data via the Huggingface API. See details in Appendix C.

However, current benchmark construction processes largely rely on human-provided signals (Chang et al., 2024; Wang et al., 2024b), leading to long cycles and high costs. To this end, efficient LLM-driven methods have recently been explored. Unfortunately, they generally rely on the existence of seed benchmarks for data augmentation (Zhu et al., 2024b; Wu et al., 2024; Li et al., 2024a; Maheshwari et al., 2024) and task specific designs (Zhu et al., 2024a; Lei et al., 2023), lacking generalization across tasks and domains. Meanwhile, the current absence of a comprehensive evaluation framework hinders the assessment and optimization of benchmark generators, weakening our confidence in their reliability for real applications. Hence, an automatic and comprehensive evaluation framework and a generic and reliable benchmark generator that can handle any assessment demands and efficiently generate high-quality samples are urgently needed.

To this end, we first construct an automatic evaluation framework with ten criteria for benchmark generators. Notably, we utilize causal learning (Kaddour et al., 2022) techniques to identify and remove biases of LLM-as-a-judge (Liu et al., 2023) across various criteria, ensuring the reliability of the

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review . Do not distribute.

framework. On this basis, we examine the strengths and weaknesses of directly prompting LLM as generic benchmark generator through this evaluation framework. The results reveal that the generated benchmark exhibits limited lexical and semantic diversity, poor controllability over difficulty, and low sample faithfulness, while showing advantages in high task alignment and knowledge diversity. Bearing this in mind, we develop a generic benchmark generator BENCHMAKER by integrating existing techniques with newly designed approaches to address the identified issues. Specifically, BENCHMAKER: strengthens sample faithfulness using stepwise self-correction generation and conflict guided contrastive discrimination; extends difficulty boundary with difficulty strategy guidance and difficulty diffusion mechanism; enhances diversity through AttrPrompt (Yu et al., 2023) and in-batch redundancy filtering. We also discuss some unsuccessful attempts in Appendix B to provide more insights for future research.

We conduct comprehensive experiments to validate BENCHMAKER under the proposed evaluation framework. Compared to high-quality human-annotated benchmarks, the benchmarks generated by BENCHMAKER exhibit superior task alignment, better difficulty controllability, more challenging question difficulties, and comparable sample faithfulness and diversity. More importantly, they yield highly consistent evaluation results across 12 LLMs (0.967 Pearson correlation with MMLU-Pro), with BENCHMAKER taking only \$0.005 and 0.38 minutes per sample. We further perform detailed experiments to validate the outstanding generalization and robustness across tasks and LLMs, and the effectiveness of each component of BENCHMAKER. Finally, we derive a formula for evaluating the confidence of benchmarking results under conditions where faithfulness cannot be fully satisfied, further enhancing the practicality and reliability of BENCHMAKER.

2. Backgrounds

In this section, we first review the latest developments in data synthesis §2.1 and then discuss the potential values of developing a generic benchmark generator §2.2.

2.1. Synthetic Data Generation

The growth of language model abilities has led to widespread research on LLM-driven data synthesis, which demonstrates much better quality and controllability over traditional approaches (Wang et al., 2024a; Long et al., 2024). Centering around the construction of data flywheel (LLM-driven evolution) (Luo et al., 2024a; Tao et al., 2024), training data synthesis has garnered much attention in fields like mathematics (Yu et al., 2024), science (Li et al., 2024b), and code (Luo et al., 2024b), continuously pushing LLMs' capability boundaries. Unlike the training data synthesis aimed at optimizing model performance, the goal of benchmark synthesis is to accurately evaluate models on specific task, presenting greater challenges in both measurement and implementation (Chang et al., 2024). In terms of measurement, recent studies (Zhu et al., 2024a; Maheshwari et al., 2024; Li et al., 2024a) generally focus on specific criteria, without establishing a comprehensive evaluation system for benchmark generators. In terms of implementation, current benchmark generators (Perez et al., 2023; Wu et al., 2024; Zhu et al., 2024b; Lei et al., 2023) are constrained by their dependence on existing benchmarks and task specific designs, preventing them from being generic. We construct a comprehensive evaluation framework and develop generic and reliable BENCHMAKER method to fill this gap.

2.2. Potential Applicable Scenarios of BENCHMAKER

Given arbitrary assessment demands X as the sole input, a generic benchmark generator (BENCHMAKER) $\mathcal{G}$ is expected to generate a well-aligned high-quality benchmark $\mathcal{D}$. On this basis, we summarize its applicable scenarios as follows: (1) Complementing existing benchmarks for tailored assessment demands; (2) Acting as a dynamic benchmark generator to alleviate data contamination issues (Balloccu et al., 2024); (3) Serving as a difficulty controllable benchmark generator to mitigate the benchmark saturation problem (Glazer et al., 2024); (4) Functioning as a versatile training data generator. Therefore, building BENCHMAKER holds significant importance for both scientific research and practical applications within the NLP community.

3. Benchmarking Benchmark Generator

While training data synthesis focuses on faithfulness, diversity and the final performance of the trained models (Yu et al., 2023; Long et al., 2024), the evaluation of synthetic benchmark should be more comprehensive to ensure the reliability of its benchmarking results. Thus, we carefully establish an evaluation framework for benchmark generator with ten criteria, as illustrated in Table 1.

3.1. Credibility

Two key criteria for ensuring the credibility of a benchmark are **faithfulness** and **alignment**. Faithfulness indicates that the generated sample be free of ambiguity with a correct answer. Alignment requires the generated samples to strictly adhere to the specified assessment demands X, especially in abilities to be assessed. For these criteria, previous approaches rely on human evaluation (Wu et al., 2024; Zhu et al., 2024b) or LLM-as-a-judge (Zheng et al., 2023). However, the former lacks automation, and the latter is susceptible to biases (Thakur et al., 2024).

To this end, we seek to detect and mitigate any biases of

Table 1. Criteria taxonomy and definition of the proposed evaluation framework for the benchmark generator. Criteria marked with * indicate optimization objectives that are distinctive to benchmark synthesis compared to training data synthesis.

Taxonomy	Criterion	Definition
Credibility	faithfulness	The sample is well-defined, with the ground truth answer being correct.
	alignment *	The abilities evaluated by the sample align well with the given assessment demands.
Diversity	lexical	The samples exhibit sufficient lexical diversity.
	semantic	The samples exhibit sufficient semantic richness.
	knowledge *	The knowledge and skills assessed by different samples should not be redundant.
Difficulty	controllability *	The samples have correct difficulty labels to form subsets with varying difficulties.
	boundary *	The hardest subset is difficult enough to explore the boundaries of advanced models.
Benchmark-Level	effectiveness *	The benchmarking results align with human benchmark under the same assessment demands.
	robustness *	The benchmarking results of generated benchmarks under similar assessment demands align.
	efficiency	The time and cost of generating a benchmark are low enough.

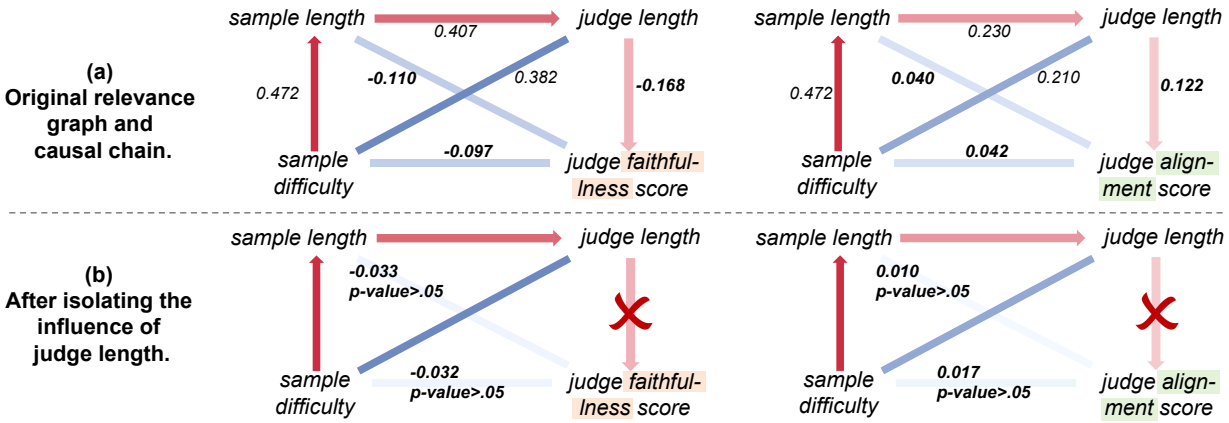

Figure 2. Pearson correlations among key factors of benchmark evaluation and LLM (Qwen-Plus) judge scores (faithfulness and alignment). The most relevant path of each subject is highlighted in red to show the possible causal chain.

LLM-as-a-judge that may exist within the framework. We choose Qwen-Plus (Yang et al., 2024) as the judge with scoring range as $[0, 1]$ (See prompt in Appendix I). Experiments are conducted on the high-quality MATH benchmark (Hendrycks et al., 2021b), for which we assign score 1 to both faithfulness and alignment for every sample. Ideally, the scores assigned by the judge should not exhibit any consistency with specific factors. However, as shown in Figure 2-(a), both faithfulness and alignment are significantly correlated (p-value < 0.05) with sample difficulty, sample length, and the length of the judge's rationale. For each factor, we highlight its weightiest path in red, revealing a clear causal chain: harder questions lead to longer samples, requiring judges to conduct lengthier analyses. For faithfulness, longer analyses increases the likelihood of judge errors, resulting in lower faithfulness ratings. While for alignment, longer analyses increases the probability of task-relevant words appearing and results in higher alignment ratings. To validate the above hypothesis, we control the judge length and respectively calculate the partial correlations (Vallat, 2018) of sample difficulty and sample length with faithfulness and alignment. As shown in Figure 2-(b), after isolating the influence of judge length, the effects of other factors are no longer significant (p-value > 0.05). Similar conclusions also hold true when GPT-4o mini (Hurst et al., 2024) serves as the judge (Figure 7 in Appendix).

Based on the analysis above, the potential biases of the LLM judge in this scenario are all mediated by judgment length. Therefore, for benchmark generators $\mathcal{G}_{1:|\mathcal{G}|}$ under evaluation, we derive their unbiased judge results with a Multiple Regression model. Specifically, we set the judge score as the dependent variable, the generator categories as dummy variables, and judge length as the covariate:

$$f(i) = \beta_i + \beta_{len} \cdot \text{judge\_length} + \epsilon \tag{1}$$

where $f(i)$ denotes the average judge score of $\mathcal{G}_i$ and β_i reflects the debiased score of $\mathcal{G}_i$, which we select as our metrics for faithfulness and alignment.

3.2. Diversity

With credibility ensured, the diversity of the benchmark determines the extent to which evaluation results can reflect the true model capability across the assessed domain. Apart

from the widely tested lexical and semantic diversity, our framework also examines the knowledge diversity to make the evaluation more comprehensive.

Lexical Diversity reflects vocabulary richness in benchmarks. Traditional metrics like vocabulary size and self-BLEU (Zhu et al., 2018) used in Wu et al. (2024) and (Yu et al., 2023) are biased by sample length (Guo & Vosoughi, 2023). We use unbiased word frequency entropy (Montahaei et al., 2019) as the metric to evaluate lexical diversity.

Semantic Diversity quantifies a benchmark's semantic comprehensiveness. We calculate the average Euclidean distance between semantic embeddings of samples as the metric. Specifically, we use powerful text-embedding-ada-002 (OpenAI) as the embedding model.

Knowledge Diversity evaluates whether the samples evaluate different sub-abilities within the assessment demands. When samples test the same sub-ability, the model is likely to exhibit similar correctness patterns. Therefore, we use the correctness of a set of models (denoted as $\mathcal{M}_{1:|\mathcal{M}|}$, see Appendix E for detailed list) to represent the knowledge embedding for each sample. If the embeddings of two samples are highly similar, it reflects a strong alignment in the sub-abilities they assess. The average pairwise Hamming distance (Hamming, 1950), suitable for discrete embeddings, is employed as the metric for this criterion.

3.3. Difficulty

When diversity meets requirements, we should further consider the difficulty attribute, which is particularly significant in an era of increasingly divergent model capabilities.

Difficulty Controllability refers to assigning differentiated difficulty labels to the samples (e.g., MATH (Hendrycks et al., 2021b)). These labels enable the benchmark to be divided into subsets for more targeted evaluation of models with varying capabilities. For each sample, we use the average error rate of $\mathcal{M}_{1:|\mathcal{M}|}$ as the ground truth for difficulty label. Based on this, we compute the Spearman correlation between the difficulty labels provided by the benchmark and the ground truth as the metric.

Difficulty Boundary denotes the difficulty of the hardest subset of a benchmark. With the growing strength of LLMs, their performance on simpler benchmarks has reached saturation (Hendrycks et al., 2021a), making it difficult to differentiate their capabilities. Consequently, more challenging benchmarks (Wang et al., 2024b) are continuously introduced to evaluate the latest LLMs. Thus, we propose assessing the average error rate of $\mathcal{M}_{1:|\mathcal{M}|}$ on the hardest subset of benchmark to measure its difficulty boundary.

3.4. Benchmark-Level

Lastly, we introduce high-level metrics for assessing benchmark generators.

Effectiveness. While the earlier criteria assess benchmark quality from various aspects, a unified metric is required to measure benchmark effectiveness. Taking high-quality human-annotated benchmark as the ground truth, we examine whether generated benchmark under identical assessment demands can deliver equivalent evaluation results. To this end, we calculate the accuracy of $\mathcal{M}_{1:|\mathcal{M}|}$ on both generated and human benchmarks and use the Pearson correlation between them as the effectiveness metric.

Robustness. Under similar inputs, a robust system should produce comparable outputs. Similarly, we expect a robust benchmark generator to produce benchmarks with equivalent evaluation efficacy for similar assessment demands. Thus, we calculate the accuracy of $\mathcal{M}_{1:|\mathcal{M}|}$ on benchmarks generated under similar assessment demands (the original and that rewritten by GPT-4o) and calculate the Pearson correlation between them as the robustness metric.

Efficiency. High-quality human-annotated benchmarks are constrained due to inefficiencies in their construction. We evaluate the efficiency of a benchmark generator by measuring the time and monetary costs associated with generating benchmarks of a certain size.

By establishing this comprehensive evaluation framework, the strengths and weaknesses of benchmark generators can be thoroughly assessed, and the reliability of the proposed method can be validated.

4. Development of BenchMaker

In this section, we first discuss the primary sample format we studied in §4.1. Afterwards, since previous studies have yet to realize generic benchmark generators (with assessment demands X as the sole input), we analyze the pros and cons of directly prompting the LLM as such generator in §4.2. Building on the experimental results, we refine its weaknesses in the following sections, leading to the development of BENCHMAKER.

4.1. Sample Format Selection

Following previous studies (Li et al., 2024a; Zhu et al., 2024b), we have chosen multiple-choice questions (MCQs) as the primary sample format for benchmark generation based on the following reasons: (1) Versatility: MCQ serves as a universal format for evaluating most capabilities; (2) Accuracy: Misjudgment can be effectively prevented caused by variations in output formats (Tam et al., 2024); (3) Effi-

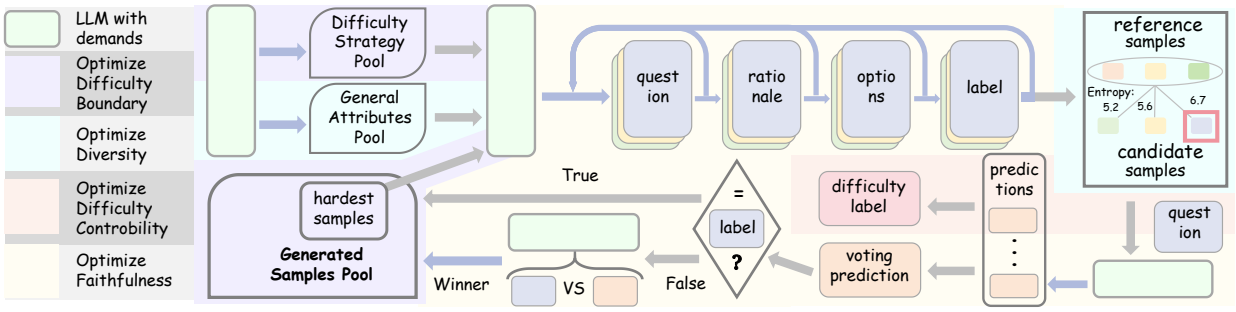

Figure 3. Overview of BENCHMAKER.

ciency: MCQs do not depend on external modules such as LLM-as-a-Judge, ensuring a streamlined evaluation process; (4) Transformability: Each generated sample includes a rationale, enabling easy conversion into other formats, such as the text generation format presented in the Appendix G.

4.2. Pros and Cons of Directly Prompting

We choose MATH (Hendrycks et al., 2021b), MMLU-Pro (Wang et al., 2024b) and HellaSwag (Zellers et al., 2019) as high-quality benchmarks $\mathcal{D}_{human}$ for comparison. We adopt the prompt in Appendix I to guide $\mathcal{M}$: GPT-4o mini in generating credible and diverse samples $s_{1:|\mathcal{D}_{human}|}$:

$$s_i = \{q_i, r_i, o_i, a_i\} = \mathcal{M}(\text{prompt}_{base}, l, X) \quad (2)$$

where q_i, r_i, o_i, a_i denote question, rationale, options, and label, respectively. We proportionally adjust the difficulty level l from 1 to 10 in the prompt (see descriptions in Appendix D), and select samples with top 20% difficulty level to form the hardest subset. The assessment demands X are shown in Appendix K. As shown in Table 2, compared to $\mathcal{D}_{human}$ (red line), directly prompting LLM as generic benchmark generator (yellow line) demonstrates poorer faithfulness, lower lexical and semantic diversity, weaker difficulty controllability, and less challenging subset. Meanwhile, we also observe its advantages in better alignment[1], greater knowledge diversity, and improved efficiency.

4.3. Faithfulness Optimization

To enhance faithfulness, previous studies have explored methods such as self-correction (Wang et al., 2023b; Ji et al., 2023) and the use of external tools (Li et al., 2024c; Lewis et al., 2020). As self-correction offers greater versatility, we propose the following two BenchMaker-compatible techniques to optimize faithfulness.

[1]We set the debiased LLM-as-a-judge score of the human benchmark to 1, adjusting scores of generated benchmarks accordingly, which may result in scores exceeding 1.

Stepwise Self-correction. Since errors might occur at any step during the generation of $\{q_i, r_i, o_i, a_i\}$, we instruct the model to validate the content at each step. If an error is detected, the model will return to the beginning. Compared to full-sample self-checking, step-wise critique boosts error detection with less decoding cost (See Appendix B).

Conflict Guided Contrastive Discrimination. Huang et al. (2024) finds that LLMs struggle to correctly judge their prior answers on challenging questions. Therefore, we extend Stepwise Self-correction by having the LLM not only act as a judge but also as a test-taker to identify potential errors. Let the LLM answers q_i T times to attain $\bar{a}_i^{1:T}$, we get the self-consistency (Wang et al., 2023a) result $\dot{a}_i$ through majority voting. If $\dot{a}_i \neq a_i$, the conflict suggests differing r_i and $\dot{r}_i$. As Zheng et al. (2023) finds that comparison-based judges are more accurate than item-wise judges, we have the LLM conduct a contrastive discrimination between r_i and $\dot{r}_i$ to determine the final rationale and label for s_i.

4.4. Difficulty Optimization

Difficulty Controllability. From §4.2, we know that the LLM's ability to control the difficulty of generated samples is limited. In particular, for the language understanding task (MMLU-Pro), the Spearman correlation between the actual and expected difficulty of the samples is only 0.021. To further explore this, we examine LLM's difficulty perception by asking it to score the difficulty label of the generated samples. However, the correlation only increases to 0.089, suggesting that while LLM has some capacity to perceive difficulty, it is still weak. We then switch the role of LLM and assess the difficulty from the perspective of test-taker:

$$\beta_i = \frac{1}{T}\sum_{j=1}^{T} \mathbf{1}_{\bar{a}_i^j \neq a_i} \quad (3)$$

By taking the inconsistency between $\bar{a}_i^{1:T}$ and a_i as difficulty label, the correlation increases to 0.415, suggesting that β is a reliable metric for difficulty controllability.

Difficulty Diffusion Mechanism. Given that the LLM has a certain level of difficulty perception, we iteratively select the more challenging samples according to β from the generated ones as difficulty references, and instruct the LLM to generate a more difficult sample. This allows the sample difficulty to rise continuously through diffusion. The detailed algorithm is described in Appendix F.

Difficulty Strategy Guidance. We further consider providing the LLM with task-specific difficulty-control strategies. Specifically, we first require the LLM to give varying strategies for generating samples of specific difficulty levels based on the given X (see examples in Appendix J). For example, difficult samples those assessing reasoning ability generally require more reasoning steps. With the Difficulty Diffusion Mechanism, we progressively introduce more difficult sample generation strategies to the LLM to further extend the difficulty boundary.

4.5. Diversity Optimization

The optimization of synthetic data diversity has been widely studied (Wang et al., 2024a). We conduct extensive tests and select the most generic and effective **AttrPrompt** (Yu et al., 2023) technique for BENCHMAKER. AttrPrompt explicitly enhances the lexical and semantic diversity of benchmarks by randomly assigning pre-generated (attribute, value) pairs as part of the input for each sample. Furthermore, we notice that the introduction of treating the generated samples as difficulty references might cause sample homogeneity. To mitigate this, we propose an **In-batch Diversity Boosting** method, where LLM generates L (We set L as 5 for our default setting) candidate samples and selects the one with the greatest word frequency entropy difference from the input reference samples.

5. Experiments and Analyses

We conduct comprehensive experiments to validate BENCHMAKER under the proposed framework in this section.

Settings. We select the widely used human-annotated MATH[2] (Hendrycks et al., 2021b) (mathematical reasoning), MMLU-Pro (multi-task language understanding) (Wang et al., 2024b) and HellaSwag (commonsense reasoning) (Zellers et al., 2019) as high-quality baseline benchmarks. For the 7 subsets of MATH, the 13 subsets of MMLU-Pro[3] and HellaSwag, we write simple assessment demands respectively (see details in Appendix K) as inputs for the benchmark generator. For each demand, we generate 500 samples and randomly downsample the human-annotated benchmark to match the number of generated samples for fair comparison. Each experiment is repeated three times, and the average results are reported. We use GPT-4o mini (Hurst et al., 2024) as the default generator and also explore the performance of GPT-4o and Claude 3.5 Haiku (Anthropic). The decoding temperature is set to 1. To mitigate the self-enhancement bias (Zheng et al., 2023) associated with LLM-as-a-judge, we substitute the generators with Qwen-Plus (Yang et al., 2024) as the judge.

[2]Converted into a MCQ format, see details in Appendix G.

[3]Excluding the type 'other'.

5.1. Comparison with Human-annotated Benchmark

As shown in Table 2, overall, BENCHMAKER achieves comparable performance to human-annotated benchmarks in terms of faithfulness and lexical&semantic diversity. Meanwhile, BENCHMAKER outperforms them in all other metrics, especially in alignment, knowledge diversity, difficulty controllability and efficiency. The exceptional results achieved in these metrics comprehensively validate the reliability of the generated samples by BENCHMAKER.

Effectiveness. The primary goal of benchmarking is to assign accurate scores to models under evaluation, facilitating capability differentiation. The benchmarking results of BENCHMAKER align closely with human-annotated benchmarks, with an average of 0.953 linear correlation (Pearson) and a remarkable 0.966 for rank-order correlation (Spearman), highlighting its outstanding effectiveness.

Robustness. Under evaluation demands where semantic equivalence is maintained but linguistic styles vary, the benchmarks exhibit nearly identical assessment efficacy, with an average Pearson correlation of 0.984. This demonstrates the robustness of BENCHMAKER to diverse inputs and ensures that users with different linguistic preferences can obtain consistent evaluation results.

Efficiency. The primary limitation of human-annotated benchmarks lies in their low construction efficiency. However, BENCHMAKER can generate a sample at an average cost of $0.005 within 0.40 minutes. Furthermore, its efficiency is expected to continuously improve with the development of technology and hardware.

Generalizability. Experimental results demonstrate that BENCHMAKER exhibits strong generalization across different task types and generators. Notably, a more powerful model does not necessarily yield superior performance across all metrics. Compared to GPT-4o, GPT-4o mini proves to be a more cost-effective benchmark generator.

5.2. Ablation Studies

We validate the effectiveness of different techniques by sequentially integrating them to the Direct Prompt baseline on the MATH benchmark, as shown in Table 2.

Table 2. Overall experimental results under the proposed evaluation framework. For each setting, we run three times and report the average results. We take GPT-4o mini as default generator. Values in bold denote the best results between BENCHMAKER and Human Benchmark.

Methods	Faithful	Alignment	Lexical	Semantic	Knowledge	Control	Boundary	Effective	Robust	Efficiency
	Unbias Score↑	Unbias Score↑	Entropy↑	Euclidean Distance↑	Hamming Distance↑	Spea-rman↑	Error Rate↑	Pear-son↑	Pear-son↑	$/item, min/item↓
MATH (Hendrycks et al., 2021b)										
Human Benchmark	**1.000**	1.000	8.054	0.665	0.349	0.143	0.752	-	-	high
Direct Prompt	0.665	1.166	7.091	0.618	0.365	0.109	0.635	0.687	0.991	0.002, 0.17
+AttrPrompt	0.611	1.138	8.265	0.675	0.360	0.124	0.659	0.759	0.983	0.002, 0.19
+InBatchDivBoost	0.623	1.142	8.652	0.677	0.366	0.115	0.628	0.778	0.985	0.003, 0.20
+StepSelfCorrect	0.924	1.152	8.674	0.675	0.369	0.162	0.557	0.803	0.979	0.003, 0.23
+ConflictConDisc	1.019	1.151	8.668	0.678	0.357	0.175	0.515	0.838	0.992	0.004, 0.35
+DiffControl	1.019	1.151	8.668	0.678	0.357	0.403	0.515	0.838	0.992	0.004, 0.35
+DiffDiffusion	0.994	1.166	8.705	0.680	0.387	0.451	0.683	0.882	0.990	0.005, 0.39
BenchMaker	0.930	1.200	**8.976**	**0.681**	0.403	**0.434**	0.768	0.935	0.986	**0.005, 0.42**
BenchMaker$_{4o}$	0.918	**1.223**	8.835	0.675	0.385	0.432	**0.779**	**0.941**	**0.988**	0.084, 1.12
BenchMaker$_{haiku}$	0.902	1.116	8.878	0.676	**0.410**	0.401	0.775	0.912	0.979	0.026, 0.57
MMLU-Pro (Wang et al., 2024b)										
Human Benchmark	1.000	1.000	**10.404**	**0.731**	0.307	0.000	0.751	-	-	high
Direct Prompt	0.894	1.218	9.608	0.726	0.391	0.021	0.587	0.850	0.989	0.002, 0.16
BenchMaker	**1.020**	**1.245**	10.166	0.728	**0.395**	**0.477**	**0.759**	0.967	0.982	**0.005, 0.38**
HellaSwag (Zellers et al., 2019)										
Human Benchmark	1.000	1.000	**9.167**	0.655	0.384	0.000	0.569	-	-	high
Direct Prompt	0.862	1.107	8.165	0.660	0.396	0.047	0.626	0.821	0.979	0.002, 0.17
BenchMaker	**1.032**	**1.130**	9.052	**0.663**	**0.421**	**0.439**	**0.708**	0.958	0.984	**0.005, 0.40**

Diversity. Compared to Direct Prompt, both AttrPrompt and In-batch Diversity Boosting effectively enhance lexical and semantic diversity. Noticeably, we observe that knowledge diversity remains unchanged, indicating that surface-level diversification does not necessarily equate to a broader assessment of knowledge and skills. Meanwhile, the diversity improvement leads to a slight drop in faithfulness, possibly because of the attributes constraints.

Faithfulness. After applying Stepwise Self-correction and Conflict Guided Contrastive Discrimination, we observe a sustained and significant improvement in faithfulness. At the same time, we notice a reduction in the difficulty of the hardest subset, with the error rate decreasing from 0.659 to 0.557. We hypothesize that this may be due to the high error rate in labels when faithfulness is not ensured, which leads to an underestimation of model performance. Consequently, once the labels are corrected, the accuracy can better reflect the actual difficulty of the benchmark.

Difficulty Controllability. By treating the generator as the test-taker and using its error rate as the difficulty label, we achieve more precise control over sample difficulty (Spearman correlation of 0.403). Considering the previously observed weak difficulty perception of LLMs, we hypothesize that this improvement stems from the role shift, which requires the model to engage in explicit reasoning, along with the adoption of prediction-label inconsistency as an objective metric.

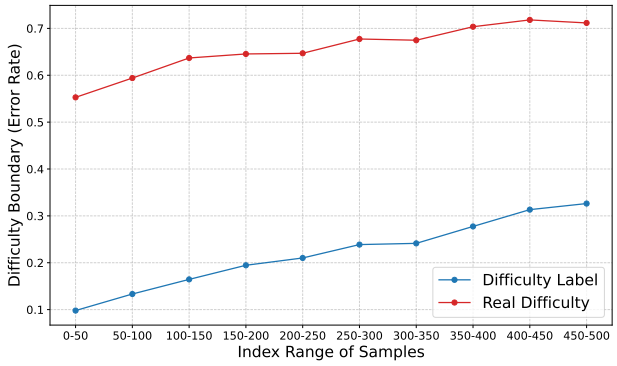

Figure 4. Trends of real and labeled difficulty over the index.

Difficulty Boundary. With our proposed Difficulty Diffusion Mechanism and Difficulty Strategy Guidance, the difficulty boundary is significantly extended, as evidenced by an increase in error rate from 0.515 to 0.768, validating their effectiveness. Additionally, we analyze how the actual difficulty and difficulty labels evolve with the order of generated samples. As illustrated in Figure 4, both the difficulty label and actual difficulty exhibit a continuous

upward trend. This not only confirms that Difficulty Diffusion Mechanism operates as intended but also visually demonstrates the strong consistency between difficulty label and actual difficulty.

(a) MMLU-Pro (b) BENCHMAKER

Figure 5. Word cloud of MMLU-Pro and the benchmark generated by BENCHMAKER under similar assessment demands.

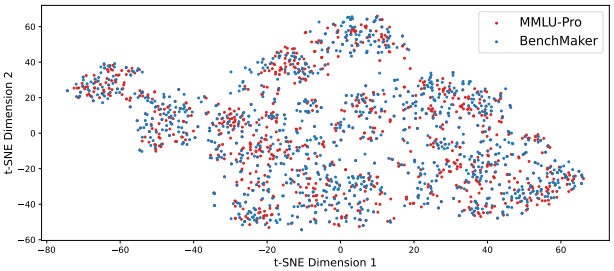

Figure 6. T-SNE results on the text embeddings of benchmarks.

5.3. A Closer Look at the Generated Benchmark

After metric analysis, we perform a more thorough examination of BENCHMAKER. Some of the generated samples are shown in Appendix H.

Lexical and Semantic. First, despite the obvious differences in word distribution between the generated benchmark and MMLU-Pro (Figure 5), it remains closely aligned with the domains covered by MMLU-Pro, demonstrating strong task alignment. Meanwhile, The semantic alignment between the two is more pronounced (Figure 6). Notably, the input demands (Appendix K) do not mention any information related to MMLU-Pro, effectively preventing the model from achieving a high degree of alignment by memorizing and replicating samples from MMLU-Pro.

Actual Error Rate. Although LLM-as-a-judge has provided an unbiased estimation of the benchmark's faithfulness, we additionally conduct a manual check on 80 randomly selected samples. Our findings indicate that 3 samples have incorrect labels, 3 samples lack a correct candidate, resulting in an overall error rate of 7.5%. Meanwhile, LLM-as-a-judge identifies 5 problematic samples, with 3 overlapping with human judgment. These results suggest that: (1) BENCHMAKER still has room for improvement in faithfulness; (2) LLM-as-a-judge can serve as a partial proxy for human evaluation.

5.4. Reliability Estimation.

Since faithfulness of the generated benchmark cannot be totally ensured, we are curious about the effects of incorrect samples: Let $\bar{a}$ and $\bar{b}$ be the observed accuracies of two models A and B on the generated benchmark of size N, where a fraction K of the samples are incorrect (which can be estimated by the LLM-as-a-judge). Suppose that $\bar{a} > \bar{b}$, we aim to estimate the probability that the observed ability rank is correct (the true accuracies satisfy $E[a] > E[b]$). See detailed derivation in Appendix A. Suppose A and B have the same accuracy p on incorrect samples, we get:

$$E[a] = \frac{\bar{a} - K \cdot p}{1 - K}, \quad E[b] = \frac{\bar{b} - K \cdot p}{1 - K} \tag{4}$$

and

$$E[a] - E[b] = \frac{\bar{a} - \bar{b}}{1 - K} \tag{5}$$

Next, we perform hypothesis testing to assess the probability of $E[a] > E[b]$. We assume that $\bar{a} - \bar{b}$ follows a normal distribution. The z-score for the difference is:

$$\begin{aligned} z &= \frac{(\bar{a} - \bar{b})/(1 - K)}{\sqrt{(\bar{a}(1 - \bar{a}) + \bar{b}(1 - \bar{b}))/(N(1 - K)^2)}} \\ &= \frac{(\bar{a} - \bar{b})\sqrt{N}}{\sqrt{\bar{a}(1 - \bar{a}) + \bar{b}(1 - \bar{b})}} \end{aligned} \tag{6}$$

where $\Phi(z)$ is the cumulative distribution function of the standard normal distribution. The probability $P(E[a] > E[b])$ is given by the right-tail probability of the normal distribution:

$$P(E[a] > E[b]) = 1 - \Phi(z) \tag{7}$$

where $\Phi(z)$ is the cumulative distribution function of the standard normal distribution. We can assess the reliability of BENCHMAKER evaluation results using (7). Also, we notice that K has the same scaling effect on both the numerator and denominator of the test statistic, thus does not alter the z-score. Consequently, as long as there is no bias, a certain proportion of noise in the benchmark will not affect the statistical significance of ability ranking.

Conclusions

The rapid advancement of large language models has driven an urgent demand for a generic benchmark generator. To this end, we first propose a comprehensive, automated, and unbiased evaluation framework to validate and optimize the reliability of benchmark generators. Based on this, we develop the BENCHMAKER method for reliable, generic, and efficient benchmark generation. Comprehensive experiments across multiple tasks and LLMs demonstrate that BENCHMAKER achieves human-aligned benchmark quality, with superior efficiency and generalization.

Impact Statement

This paper presents BENCHMAKER, an LLM-driven reliable , generic and efficient benchmark generator. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Ethics Statement. All of the datasets used in this study were publicly available. Multiple authors jointly conducted the manual check for the Actual Error Rate section, and no extra annotators were employed for our data collection. We confirm that the datasets we used did not contain any harmful content and was consistent with their intended use (research). We have cited the datasets and relevant works used in this study.

References

Anthropic. Claude 3.5. https://www.anthropic.com/news/3-5-models-and-computer-use.

Balloccu, S., Schmidtová, P., Lango, M., and Dusek, O. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pp. 67–93. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.eacl-long.5.

Chan, X., Wang, X., Yu, D., Mi, H., and Yu, D. Scaling synthetic data creation with 1,000,000,000 personas. *CoRR*, abs/2406.20094, 2024. doi: 10.48550/ARXIV.2406.20094. URL https://doi.org/10.48550/arXiv.2406.20094.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., and Xie, X. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3):39:1–39:45, 2024. doi: 10.1145/3641289. URL https://doi.org/10.1145/3641289.

Glazer, E., Erdil, E., Besiroglu, T., Chicharro, D., Chen, E., Gunning, A., Olsson, C. F., Denain, J., Ho, A., de Oliveira Santos, E., Järviniemi, O., Barnett, M., Sandler, R., Vrzala, M., Sevilla, J., Ren, Q., Pratt, E., Levine, L., Barkley, G., Stewart, N., Grechuk, B., Grechuk, T., Enugandla, S. V., and Wildon, M. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in AI. *CoRR*, abs/2411.04872, 2024. doi: 10.48550/ARXIV.2411.04872. URL https://doi.org/10.48550/arXiv.2411.04872.

Guo, X. and Vosoughi, S. Length does matter: Summary length can bias summarization metrics. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 15869–15879. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.984. URL https://doi.org/10.18653/v1/2023.emnlp-main.984.

Hamming, R. W. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Huang, J., Chen, X., Mishra, S., Zheng, H. S., Yu, A. W., Song, X., and Zhou, D. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=IkmD3fKBPQ.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Ji, Z., Yu, T., Xu, Y., Lee, N., Ishii, E., and Fung, P. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 1827–1843. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.123. URL https://doi.org/10.18653/v1/2023.findings-emnlp.123.

Kaddour, J., Lynch, A., Liu, Q., Kusner, M. J., and Silva, R. Causal machine learning: A survey and open problems. *CoRR*, abs/2206.15475, 2022. doi: 10.48550/ARXIV.

2206.15475. URL https://doi.org/10.48550/arXiv.2206.15475.

Lei, F., Liu, Q., Huang, Y., He, S., Zhao, J., and Liu, K. S3eval: A synthetic, scalable, systematic evaluation suite for large language models. *CoRR*, abs/2310.15147, 2023. doi: 10.48550/ARXIV.2310.15147. URL https://doi.org/10.48550/arXiv.2310.15147.

Lewis, P. S. H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Li, J., Hu, R., Huang, K., Zhuang, Y., Liu, Q., Zhu, M., Shi, X., and Lin, W. Perteval: Unveiling real knowledge capacity of llms with knowledge-invariant perturbations. *CoRR*, abs/2405.19740, 2024a. doi: 10.48550/ARXIV.2405.19740. URL https://doi.org/10.48550/arXiv.2405.19740.

Li, S., Huang, J., Zhuang, J., Shi, Y., Cai, X., Xu, M., Wang, X., Zhang, L., Ke, G., and Cai, H. Scilitllm: How to adapt llms for scientific literature understanding. *CoRR*, abs/2408.15545, 2024b. doi: 10.48550/ARXIV.2408.15545. URL https://doi.org/10.48550/arXiv.2408.15545.

Li, Z., Zhou, Z., Yao, Y., Li, Y., Cao, C., Yang, F., Zhang, X., and Ma, X. Neuro-symbolic data generation for math reasoning. *CoRR*, abs/2412.04857, 2024c. doi: 10.48550/ARXIV.2412.04857. URL https://doi.org/10.48550/arXiv.2412.04857.

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 2511–2522. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.153.

Long, L., Wang, R., Xiao, R., Zhao, J., Ding, X., Chen, G., and Wang, H. On llms-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 11065–11082. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.658. URL https://doi.org/10.18653/v1/2024.findings-acl.658.

Luo, H., Sun, Q., Xu, C., Zhao, P., Lin, Q., Lou, J., Chen, S., Tang, Y., and Chen, W. Arena learning: Build data flywheel for llms post-training via simulated chatbot arena. *CoRR*, abs/2407.10627, 2024a. doi: 10.48550/ARXIV.2407.10627. URL https://doi.org/10.48550/arXiv.2407.10627.

Luo, Z., Xu, C., Zhao, P., Sun, Q., Geng, X., Hu, W., Tao, C., Ma, J., Lin, Q., and Jiang, D. Wizardcoder: Empowering code large language models with evol-instruct. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=UnUwSIgK5W.

Maheshwari, G., Ivanov, D., and Haddad, K. E. Efficacy of synthetic data as a benchmark. *CoRR*, abs/2409.11968, 2024. doi: 10.48550/ARXIV.2409.11968. URL https://doi.org/10.48550/arXiv.2409.11968.

Montahaei, E., Alihosseini, D., and Baghshah, M. S. Jointly measuring diversity and quality in text generation models. *CoRR*, abs/1904.03971, 2019. URL http://arxiv.org/abs/1904.03971.

OpenAI. text-embedding-ada-002. https://platform.openai.com/docs/guides/embeddings.

Perez, E., Ringer, S., Lukosiute, K., Nguyen, K., Chen, E., Heiner, S., Pettit, C., Olsson, C., Kundu, S., Kadavath, S., Jones, A., Chen, A., Mann, B., Israel, B., Seethor, B., McKinnon, C., Olah, C., Yan, D., Amodei, D., Amodei, D., Drain, D., Li, D., Tran-Johnson, E., Khundadze, G., Kernion, J., Landis, J., Kerr, J., Mueller, J., Hyun, J., Landau, J., Ndousse, K., Goldberg, L., Lovitt, L., Lucas, M., Sellitto, M., Zhang, M., Kingsland, N., Elhage, N., Joseph, N., Mercado, N., DasSarma, N., Rausch, O., Larson, R., McCandlish, S., Johnston, S., Kravec, S., Showk, S. E., Lanham, T., Telleen-Lawton, T., Brown, T., Henighan, T., Hume, T., Bai, Y., Hatfield-Dodds, Z., Clark, J., Bowman, S. R., Askell, A., Grosse, R., Hernandez, D., Ganguli, D., Hubinger, E., Schiefer, N., and Kaplan, J. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 13387–13434. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.847. URL https://doi.org/10.18653/v1/2023.findings-acl.847.

Tam, Z. R., Wu, C., Tsai, Y., Lin, C., Lee, H., and Chen, Y. Let me speak freely? A study on the impact of format

restrictions on large language model performance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - Industry Track, Miami, Florida, USA, November 12-16, 2024*, pp. 1218–1236. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.emnlp-industry.91.

Tao, Z., Lin, T., Chen, X., Li, H., Wu, Y., Li, Y., Jin, Z., Huang, F., Tao, D., and Zhou, J. A survey on self-evolution of large language models. *CoRR*, abs/2404.14387, 2024. doi: 10.48550/ARXIV.2404.14387. URL https://doi.org/10.48550/arXiv.2404.14387.

Thakur, A. S., Choudhary, K., Ramayapally, V. S., Vaidyanathan, S., and Hupkes, D. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges. *CoRR*, abs/2406.12624, 2024. doi: 10.48550/ARXIV.2406.12624. URL https://doi.org/10.48550/arXiv.2406.12624.

Vallat, R. Pingouin: statistics in python. *J. Open Source Softw.*, 3(31):1026, 2018. doi: 10.21105/JOSS.01026. URL https://doi.org/10.21105/joss.01026.

Wang, K., Zhu, J., Ren, M., Liu, Z., Li, S., Zhang, Z., Zhang, C., Wu, X., Zhan, Q., Liu, Q., and Wang, Y. A survey on data synthesis and augmentation for large language models. *CoRR*, abs/2410.12896, 2024a. doi: 10.48550/ARXIV.2410.12896. URL https://doi.org/10.48550/arXiv.2410.12896.

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023a. URL https://openreview.net/forum?id=1PL1NIMMrw.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 13484–13508. Association for Computational Linguistics, 2023b. doi: 10.18653/V1/2023.ACL-LONG.754. URL https://doi.org/10.18653/v1/2023.acl-long.754.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., Li, T., Ku, M., Wang, K., Zhuang, A., Fan, R., Yue, X., and Chen, W. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *CoRR*, abs/2406.01574, 2024b. doi: 10.48550/ARXIV.2406.01574. URL https://doi.org/10.48550/arXiv.2406.01574.

Wu, S., Huang, Y., Gao, C., Chen, D., Zhang, Q., Wan, Y., Zhou, T., Zhang, X., Gao, J., Xiao, C., and Sun, L. Unigen: A unified framework for textual dataset generation using large language models. *CoRR*, abs/2406.18966, 2024. doi: 10.48550/ARXIV.2406.18966. URL https://doi.org/10.48550/arXiv.2406.18966.

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Yang, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Liu, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Guo, Z., and Fan, Z. Qwen2 technical report. *CoRR*, abs/2407.10671, 2024. doi: 10.48550/ARXIV.2407.10671. URL https://doi.org/10.48550/arXiv.2407.10671.

Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J. T., Li, Z., Weller, A., and Liu, W. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=N8N0hgNDRt.

Yu, Y., Zhuang, Y., Zhang, J., Meng, Y., Ratner, A. J., Krishna, R., Shen, J., and Zhang, C. Large language model as attributed training data generator: A tale of diversity and bias. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/ae9500c4f5607caf2eff033c67daa9d7-Abstract-Datasets_and_Benchmarks.html.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

Zheng, L., Chiang, W., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P.,

Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Zhu, K., Chen, J., Wang, J., Gong, N. Z., Yang, D., and Xie, X. Dyval: Dynamic evaluation of large language models for reasoning tasks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=gjfOL9z5Xr.

Zhu, K., Wang, J., Zhao, Q., Xu, R., and Xie, X. Dyval 2: Dynamic evaluation of large language models by meta probing agents. *CoRR*, abs/2402.14865, 2024b. doi: 10.48550/ARXIV.2402.14865. URL https://doi.org/10.48550/arXiv.2402.14865.

Zhu, Y., Lu, S., Zheng, L., Guo, J., Zhang, W., Wang, J., and Yu, Y. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pp. 1097–1100. ACM, 2018. doi: 10.1145/3209978.3210080. URL https://doi.org/10.1145/3209978.3210080.

A. Derivation of the Probability for $E[a] > E[b]$

In this appendix, we present a detailed derivation of the probability that the expected accuracy $E[a]$ of model A is greater than the expected accuracy $E[b]$ of model B in a noiseless benchmark, given the noisy benchmark observations. We will use the following notation throughout:

- N: The total number of samples in the benchmark.
- K: The proportion of samples with incorrect labels, i.e., the noise ratio, where $K \in [0, 1]$.
- $\bar{a}$: The observed accuracy of model A on the noisy benchmark.
- $\bar{b}$: The observed accuracy of model B on the noisy benchmark.
- p: The probability that both models predict the incorrect label correctly on a noisy sample. This probability is assumed to be identical for both models on incorrect labels.
- $E[a]$: The expected accuracy of model A on the noiseless benchmark.
- $E[b]$: The expected accuracy of model B on the noiseless benchmark.

Given these notations, our goal is to determine the probability that $E[a] > E[b]$ based on the noisy observed accuracies $\bar{a}$ and $\bar{b}$.

Step 1: Relating $\bar{a}$ and $\bar{b}$ to $E[a]$ and $E[b]$

From the given setup, we know that the observed accuracies $\bar{a}$ and $\bar{b}$ can be written as a weighted average of the expected accuracies $E[a]$ and $E[b]$ on the correct samples, and the probability p on the incorrect samples. Specifically, the formulas for $\bar{a}$ and $\bar{b}$ are:

$$\bar{a} = (1 - K) \cdot E[a] + K \cdot p$$
$$\bar{b} = (1 - K) \cdot E[b] + K \cdot p$$

These equations express the observed accuracy of each model as the weighted average of the correct label samples and the noisy (incorrect label) samples. The weight $(1 - K)$ represents the proportion of correct labels, and K represents the proportion of incorrect labels.

Step 2: Solving for $E[a]$ and $E[b]$

To isolate $E[a]$ and $E[b]$, we rearrange the above equations:

$$E[a] = \frac{\bar{a} - K \cdot p}{1 - K}$$
$$E[b] = \frac{\bar{b} - K \cdot p}{1 - K}$$

Thus, $E[a]$ and $E[b]$ are directly related to the observed accuracies $\bar{a}$ and $\bar{b}$, and the noise ratio K.

Step 3: Comparing $E[a]$ and $E[b]$

To determine the probability that $E[a] > E[b]$, we first compute the difference between $E[a]$ and $E[b]$:

$$E[a] - E[b] = \frac{\bar{a} - K \cdot p}{1 - K} - \frac{\bar{b} - K \cdot p}{1 - K}$$

Simplifying this expression:

$$E[a] - E[b] = \frac{\bar{a} - \bar{b}}{1 - K}$$

Thus, $E[a] > E[b]$ if and only if $\bar{a} - \bar{b} > 0$, which indicates that the observed accuracy of model A must be greater than that of model B in the noisy benchmark for $E[a]$ to exceed $E[b]$ in the noiseless benchmark.

Step 4: Statistical Hypothesis Testing

To quantify the probability of $E[a] > E[b]$, we perform a hypothesis test on $\bar{a} - \bar{b}$, assuming that both $\bar{a}$ and $\bar{b}$ are derived from binomial distributions (since they represent the correct classification probabilities on the noisy benchmark). We assume that the difference $\bar{a} - \bar{b}$ follows a normal distribution under certain conditions (via the Central Limit Theorem). Thus, the expected value of $\bar{a} - \bar{b}$ is:

$$\mathbb{E}[\bar{a} - \bar{b}] = \frac{\bar{a} - \bar{b}}{1 - K}$$

The variance of $\bar{a} - \bar{b}$, assuming independent samples, is given by:

$$\begin{aligned} \text{Var}[\bar{a} - \bar{b}] =& \frac{\text{Var}[\bar{a}] + \text{Var}[\bar{b}]}{(1 - K)^2} \\ =& \frac{\bar{a}(1 - \bar{a})/N + \bar{b}(1 - \bar{b})/N}{(1 - K)^2} \end{aligned} \tag{8}$$

where $\text{Var}[\bar{a}]$ and $\text{Var}[\bar{b}]$ are the variances of the observed accuracies of models A and B, respectively. These variances can be computed from the binomial distributions underlying $\bar{a}$ and $\bar{b}$.

Now, we define the z-score for the observed difference $\bar{a} - \bar{b}$ as follows:

$$\begin{aligned} z =& \frac{\mathbb{E}[\bar{a} - \bar{b}]}{\sqrt{\text{Var}[\bar{a} - \bar{b}]}} \\ =& \frac{(\bar{a} - \bar{b})/(1 - K)}{\sqrt{(\bar{a}(1 - \bar{a}) + \bar{b}(1 - \bar{b}))/(N(1 - K)^2)}} \\ =& \frac{(\bar{a} - \bar{b})\sqrt{N}}{\sqrt{\bar{a}(1 - \bar{a}) + \bar{b}(1 - \bar{b})}} \end{aligned} \tag{9}$$

This z-score follows a standard normal distribution. The probability that $E[a] > E[b]$ is the probability that $\bar{a} - \bar{b} > 0$, which is equivalent to:

$$P(\bar{a} - \bar{b} > 0) = P(z > 0) = 1 - \Phi(z)$$

where $\Phi(z)$ is the cumulative distribution function (CDF) of the standard normal distribution. Thus, the p-value is given by:

$$p\text{-value} = 1 - \Phi(z)$$

Step 5: Conclusion

To summarize, the probability that $E[a] > E[b]$ in the noiseless benchmark, given the noisy benchmark observations, is determined by the observed accuracy difference $\bar{a} - \bar{b}$ and the noise ratio K. The probability is computed using a hypothesis test on $\bar{a} - \bar{b}$, assuming it follows a normal distribution. The final formula for this probability is:

$$P(E[a] > E[b]) = 1 - \Phi(z)$$

where $\Phi(z)$ is the CDF of the standard normal distribution and z is the computed z-score. This result allows us to assess the likelihood that model A has a higher expected accuracy than model B in the noiseless benchmark based on noisy observations.

B. Unsuccessful Attempts for Optimizing Benchmark Generator

B.1. Faithfulness

We explored the widely studied self-correction strategy to improve the faithfulness of benchmarks. Specifically, for each generated sample, the model first acts as a judge and then refines samples it deems insufficiently faithful. However, our preliminary results indicate that while this approach yields minor improvements in mathematical tasks, it provides little benefit for tasks such as MMLU-Pro and instead introduces additional computational overhead.

B.2. Difficulty Controllability

As previously mentioned, we attempted to have the model generate samples with specified difficulty levels, but the resulting samples exhibited low difficulty differentiation. To address this, we further explored having the model assess the difficulty of its generated samples. However, this strategy yielded promising results only on the MATH task.

B.3. Difficulty Diffusion Mechanism

Previous studies (Wang et al., 2024b) have attempted to increase question difficulty by expanding the number of answer choices. However, our experiments show that scaling up the number of candidates quickly reaches a saturation point. We hypothesize that this is due to the model's difficulty in generating a large number of sufficiently deceptive distractors.

B.4. Diversity

To enhance sample diversity, in addition to AttrPrompt, we experimented with assigning different personas (Chan et al., 2024) to the model and instructing it to generate characteristic samples based on its assigned persona. However, we found that this approach was not particularly effective for the MATH task, especially in semantic diversity.

C. Data from Huggingface

We obtained information on open-source model releases and download counts from the Hugging Face API (`from huggingface_hub import HfApi`). Since the number of open-source model releases far exceeds that of closed-source models, we use the former to represent the "Number of Language Model Releases." Additionally, as Hugging Face does not provide monthly download counts for each model, we use the historical total downloads of models released within a given statistical period as the total downloads for that period. The corresponding code is shown below.

D. Difficulty Levels

- **Level 1**: The simplest, equivalent to lower-grade elementary school
- **Level 2**: Relatively simple, equivalent to upper-grade elementary school
- **Level 3**: Simple, equivalent to middle school
- **Level 4**: Average, equivalent to high school
- **Level 5**: Slightly difficult, equivalent to university student
- **Level 6**: Difficult, equivalent to Master's
- **Level 7**: Quite difficult, equivalent to PhD student

- **Level 8**: Very difficult, equivalent to professor
- **Level 9**: Extremely difficult, equivalent to field expert
- **Level 10**: Most difficult, equivalent to top human level or beyond human level

E. Benchmarking Model List

- **phoenix-inst-chat-7b**: https://huggingface.co/FreedomIntelligence/phoenix-inst-chat-7b
- **vicuna-7b-v1.3**: https://huggingface.co/lmsys/vicuna-7b-v1.3
- **Qwen2.5-3B**: https://huggingface.co/Qwen/Qwen2.5-3B
- **phi-2**: https://huggingface.co/microsoft/phi-2
- **Phi-3.5-mini-instruct**: https://huggingface.co/microsoft/Phi-3.5-mini-instruct
- **Yi-1.5-6B-Chat**: https://huggingface.co/01-ai/Yi-1.5-6B-Chat
- **Qwen2.5-7B**: https://huggingface.co/Qwen/Qwen2.5-7B
- **vicuna-7b-v1.5**: https://huggingface.co/lmsys/vicuna-7b-v1.5
- **Qwen2-1.5B-Instruct**: https://huggingface.co/Qwen/Qwen2-1.5B-Instruct
- **phoenix-inst-chat-7b-v1.1**: https://huggingface.co/FreedomIntelligence/phoenix-inst-chat-7b-v1.1
- **Qwen-Plus**: https://huggingface.co/Qwen
- **GPT-3.5 turbo**: https://openai.com/index/gpt-3-5-turbo-fine-tuning-and-api-updates/

F. Details of Difficulty Diffusion Mechanism

Given that the LLM has a certain level of difficulty perception, we iteratively select the more challenging samples according to β from the generated ones as difficulty references, and instruct the LLM to generate a more difficult sample. Specifically, To prevent reference samples from becoming overly fixed, which may lead to homogenization in generated samples, we adopt the following strategy:

1. We track the number of times each sample x_i has been used as a reference sample, denoted as t_i, and compute a calibrated difficulty label:

$$\text{Calibrate\_Difficulty} = \text{Difficulty\_Label} \times 0.9^{t_i/\text{Reference\_Number}} \tag{10}$$

   The samples are then sorted based on this adjusted difficulty.

2. Each time, we select $2 \times$ Reference_Number samples with the highest Calibrate_Difficulty as candidates. From this pool, we randomly sample Reference_Number as reference samples and shuffle their order.

Our preliminary experiments indicate a positive correlation between problem difficulty and Reference_Number. In our experiments, we set Reference_Number to 8. This allows the sample difficulty to rise continuously through diffusion.

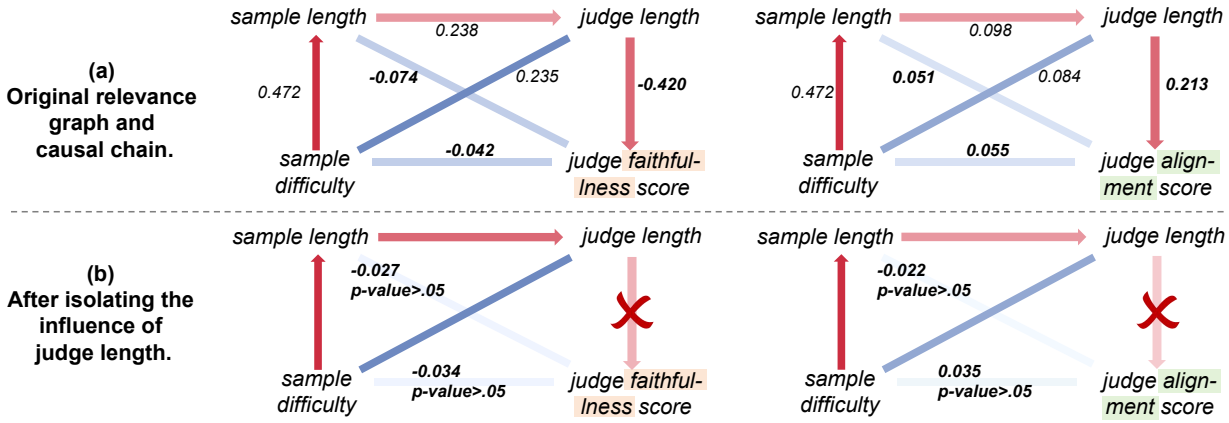

Figure 7. Pearson correlations among key factors of benchmark evaluation and LLM (GPT-4o mini) judge scores (faithfulness and alignment). The most relevant path of each subject is highlighted in red to show the possible causal chain.

G. Converting Benchmark Sample Format

MCQ to OTG Format. By removing the options from the samples and using the solution and answer corresponding to the correct option as the ground truth, we can easily transform the MCQ-style benchmark generated based on MATH assessment demands into an open-ended text generation (OTG) benchmark. Comparing these two benchmark formats, we find that the OTG format makes the questions more challenging (error rate: $0.865 > 0.768$) and results in lower knowledge diversity (hamming distance: $0.365 < 0.403$). We attribute this to the model's inability to rely on option cues to answer certain questions, which leads to a large portion of the knowledge vector being zero, thereby reducing knowledge diversity. Additionally, we observe a decline in the benchmark's effectiveness (pearson: $0.915 < 0.935$), which we hypothesize is indirectly caused by the drop in knowledge diversity.

OTG to MCQ Format. To convert the MATH benchmark into a MCQ format, we employed GPT-4o, GPT-4o mini, GPT-3.5 turbo, and Claude-3.5 Haiku to answer MATH problems, sampling 10 responses per model with a temperature of 1. We identified the three most frequently occurring incorrect answers as distractors while retaining the correct answer's rationale. If the number of incorrect answers was insufficient, we supplemented them using GPT-4o mini. Through this process, we successfully transformed the MATH benchmark from an OTG format to an MCQ format.

H. Examples of the Generated Samples

MATH:

```
Example 1:
A researcher is studying the distribution of three specific proteins in a cell. There are
    4 locations within the cell where each protein can be present. However, due to
    experimental conditions, at least one protein must be present in each location. In how
     many different ways can the proteins be distributed in the cell, considering overlap
    in presence is allowed?
A. 2187
B. 2401
C. 4096
D. 2048

Label:B


Example 2:
Find the smallest positive integer \\( n \\) such that \\( n \\) is divisible by 6, 10,
    and 15, and \\( n \\equiv 2 \\pmod{4} \\).
A. 120
B. 20
C. 90
D. 30
```

```
Label:D
```

MMLU-Pro:

```
Example 1:
A 45-year-old woman with type 2 diabetes decides to improve her health by adopting a low-
    carbohydrate, high-protein diet, starting a daily 30-minute brisk walk routine, and
    taking a new medication that increases insulin sensitivity. She also begins consuming
    a herbal supplement believed to enhance energy levels. After two months, she notices
    an increase in fatigue, frequent headaches, and unexplained weight gain. What is the
    most likely reason for her symptoms?
A. Low-carbohydrate diet leading to nutritional deficiencies
B. Brisk walk routine causing excessive physical exertion
C. Medication side effects causing insulin fluctuations
D. Herbal supplement causing hormonal imbalance
E. Increased protein intake causing kidney strain
F. Inadequate hydration from dietary changes
G. Overconsumption of high-protein foods leading to weight gain
H. Lack of fiber intake affecting metabolism
I. Decrease in carbohydrate intake causing energy depletion
J. Stress from lifestyle changes impacting health

Label:C


Example 2:
An architect is designing a complex apartment building, which features a series of
    irregularly shaped balconies. The layout of one of the building's wings is depicted in
     the accompanying diagram. Each balcony's area is defined by the function \( f(x) = 3x
    ^2 + 2x + 1 \) over the interval [0, 2] meters, representing a horizontal cross-
    section. The total length of the wing is 10 meters, and each balcony occurs at every
    meter along this length, aligned perpendicularly. To meet safety regulations, the
    architect needs to ensure that the probability of a randomly selected balcony having
    an area greater than 8 square meters is at least 0.5. Calculate the probability that a
     randomly selected balcony from this wing has an area greater than 8 square meters,
    using integration to determine the areas and probabilities involved. Consider
    potential pitfalls like incorrect integral setup or probability interpretation.
A. 0.1
B. 0.2
C. 0.3
D. 0.4
E. 0.5
F. 0.6
G. 0.7
H. 0.8
I. 0.9
J. 1.0

Label:J
```

HellaSwag:

```
During a family reunion, Mark is honored with the 'Outstanding Contributor' award for his
    recent volunteer efforts in the community. As he stands in front of his relatives, he
    expresses heartfelt gratitude towards everyone who supported him but fails to mention
    his younger sister, Lily, who organized the charity event that helped him earn this
    recognition. After the ceremony, Lily watches Mark celebrate with others, her face a
    mix of pride and disappointment. When Mark approaches her, excitedly asking, 'Did you
    see me win? I couldn't have done it without your help!' Given Lily's conflicting
    feelings about being overlooked, how is she most likely to respond?
A. 'Congratulations, Mark! I'm really proud of you! But I can't help feeling a bit
    overshadowed since I organized the event.'
```

```
B. 'Wow, Mark! You totally deserve this! Yet, it's tough for me to celebrate when my
    efforts went unnoticed.'
C. 'That was an amazing award, Mark! I'm happy for you! However, it stings that my
    contribution was overlooked.'
D. 'I'm so thrilled for you, Mark! Your achievement is incredible! But it feels a little
    unfair that I didn't get a shoutout.'

Label:C
```

I. Prompt List

LLM as Faithfulness Judge:

```
You are an expert who excels at analyzing whether a given response correctly answers a
    provided question.

**Question:**
{{question}}

**Response to be Checked:**
{{response}}

Please note that the given question may be unsolvable, have a unique solution, multiple
    solutions, etc.
Therefore, you should carefully analyze the correctness of the response to be checked
    based on the given question.

Here are the rules to strictly follow when analyzing the correctness of a response:
1. **Step-by-Step Analysis**: Analyze the response step by step, reviewing the reasoning
    and correctness of each step. For every step, first **restate and summarize** the
    reasoning logic and conclusion presented in the response, then analyze the correctness
     of that specific step.
2. **Focus on Evaluation**: Remember that your primary mission is to determine whether the
     reasoning process is correct. Avoid attempting to solve the problem yourself. Instead
    , focus strictly on analyzing the correctness of the response's reasoning process, one
     step at a time.
3. **Avoid Premature Judgments**: Do not rush to make judgments (such as claiming the
    response is flawed or completely correct) at the beginning. Ensure your evaluation is
    based on a thorough step-by-step analysis before arriving at a conclusion.
4. **Reverse Validation**: After completing the step-by-step analysis, substitute the
    answer back into the original problem and perform reverse validation of the parameters
     to cross-verify the correctness of the response.
After completing your analysis, please provide your judgment on the correctness of the
    response, as well as your confidence level in that judgment.

Your output should follow the template and example below:
Analyses:{Your detailed analyses}
Judgement:{0: You think both the final answer of the response is wrong; 0.5: You think the
     reasoning path has some mistakes, but the final answer of the response is correct; 1:
     You agree with the reasoning path and the final answer of the response}

##Example##
Analyses:{Your detailed analyses}
Judgement:1
##Example End##

Now begin with "Analyses:"
```

LLM as Comparison-based Faithfulness Judge:

```
You are a knowledgeable expert with the task of analyzing the quality of a given question
    and its candidate answers.
```

```
###Question
{{question}}

###Candidate 1:
{{can1}}

###Candidate 2:
{{can2}}

###Your task: Correctness Analysis
1. Analyze whether the question is correct, reasonable, and clearly stated.
2. For the given question, analyze whether the provided ###Candidate 1 and ###Candidate 2
    are correct step by step sequentially.
(Do not favor a candidate just because it is long; evaluate candidates strictly based on
    correctness.)
3. Based on the above analysis, output your judgment of the question quality according to
    the following scale:
    0 point indicate an incorrect question with ambiguities and no uniquely suitable
        answer among the options.
    0.5 point indicates a minor error in the question, but there is still a uniquely
        suitable answer among the options.
    1 point indicate no errors in the question, with one uniquely correct answer among the
         options.
4. Please also output your chosen correct option
You should follow the template below to output:
"##Faithfulness:{{score}}##, ##Label:{{}}##" (e.g., ##Faithfulness:2##,##Label:B##).
Please note that if you believe there is no correct option or there are multiple correct
    options, output ##Faithfulness:0##, ##Label:None##.

You should begin your response with "Correctness Analysis".
```

LLM as Relevance Judge:

```
You are an expert who excels at analyzing whether a given question can be used to assess a
     specific ability.

**Question:**
{{question}}

**Ability:**
{{ability}}

You should first carefully analyze what abilities the given question can be used to test.
Based on this analysis, compare it with the given abilities.
After completing your analysis, please provide your judgment on whether the given question
     can be used to test the given ability, as well as your confidence in that judgment.

Your output should follow the template below:
Analyses:{Your detailed analyses}
Judgement:
{output 0 if: You believe the given question is completely unable to test the given
    ability;
output 0.5 if: You believe the given question is primarily meant to test other abilities,
    but can also test the given ability to some extent;
output 1 if: You believe the given question primarily tests the given ability.}

Now begin with "Analyses:"
```

Directly Prompting LLM as Generic Benchmark Generator: Notably, before allowing the LLM to formally generate the benchmark, we first require it to produce descriptions for each part of the sample based on the assessment demands, including Task Description, Query Description, and Option Description. This helps the model better understand and align with the assessment demands, ensuring higher-quality and more consistent benchmark generation.

```
You are a knowledgeable benchmark creator.
Your task is to generate a creative questions based on the provided Task Description,
    Query Description, Option Description, Generation Guidelines, and Output Description
    to help build a benchmark that assesses the given task.

### Task Description:
{{task define}}

### Query Description:
{{query define}}

### Option Description:
{{option define}}

### Generation Guidelines:
1. Analyze the given task and think step-by-step about the content needed to construct the
     question, begin with "Analyses:".
2. Generate the question content, begin with "Question:".
3. Generate 10 candidates, with only one as the right option. Begin with "Candidates:".
4. Generate the index of the right option, begin with "Right Option:".

### Output Description:
Strictly follow the template below to generate your sample.
**Template**
##Analyses:## {{You analyze the provided attributes and outline the process for
    constructing the question to be generated.}}
##Question:## {{Your generated question content}}
##Candidates:##
{{Your generated Candidates}}
##Right Option:##{{Index of the right option, e.g., B}}
**Template End**

Attention: You need to **strictly follow the template** and don't generate any other
    contents. Begin your response with "##Analyses:## "
```

J. Examples of the Generated Difficulty Strategies

MATH:

```
Strategy 1:
Complexity of Biological Concept is Basic
Complexity of Biological Concept is Intermediate
Complexity of Biological Concept is Advanced

Strategy 2:
Required Reasoning Steps set as Single-step
Required Reasoning Steps set as Multi-step (2-3 steps)
Required Reasoning Steps set as Multi-step (4-6 steps)
Required Reasoning Steps set as More than 6 steps

Strategy 3:
Familiarity with the Topic is Common
Familiarity with the Topic is Uncommon
Familiarity with the Topic is Rare

Strategy 4:
Type of Biological Data Analysis is Qualitative
Type of Biological Data Analysis is Quantitative
Type of Biological Data Analysis is Advanced Data Interpretation

Strategy 5:
Application of Concepts is Direct
Application of Concepts is Modified
```

```
Application of Concepts is Novel

Strategy 6:
Integration Across Biological Disciplines is Single-discipline
Integration Across Biological Disciplines is Cross-disciplinary
Integration Across Biological Disciplines is Interdisciplinary

Strategy 7:
Depth of Required Knowledge is Surface-level
Depth of Required Knowledge is In-depth
Depth of Required Knowledge is Comprehensive
```

Prompt of BENCHMAKER:

```
You are a knowledgeable benchmark creator.
Your task is to generate a creative question based on the provided Task Description, Query
     Description, Option Description, General Attributes Descriptions, Difficulty
    Strategies Description, Generation Guidelines, and Output Description to help build a
    benchmark that assesses the given task.

### Overall Task Description:
{{original task}}


### Detailed Task Description:
{{task define}}


### Query Description:
{{query define}}


### Option Description:
{{option define}}


### General Attributes Description:
You can refer to the following attributes and their corresponding values to construct
    questions, which means the questions you generate should ideally align with some of
    these attributes.
Please note, if you find any conflicting or confusing parts among the attributes listed,
    you may disregard them.
{{attribute define}}


### Difficulty Strategies Description:
Your generated questions should meet the following difficulty attribute requirements. If
    you find conflicts among these requirements, you may choose to selectively ignore them
    .
{{difficulty attribute define}}


### Difficulty Description:
The following are some samples (0 or several).
Please ensure that the difficulty level of the samples you generate is harder than these
    examples.
The samples you generate should aim to assess different knowledge and skills compared to
    the given samples.
The format of given samples are not what you should follow.
**Please ensure that the sample you create differ substantially from the following samples
    , so as to maintain diversity in the resulting benchmark.**
{{demonstrations}}


### Generation Guidelines:
```

```
**Stage 1: Analyze**
In this stage, you should analyze following the steps below and begin with "##Analyses:##". **You need to clearly articulate the analysis content for each step**, which means after completing Stage 1, you should have already produced a question that meets the requirements along with a correct and unique answer.
1-1. Analyze the general attributes, difficulty attributes and difficulty description, and think step-by-step about the content needed to construct the question. **Please use your imagination and avoid any obvious overlap with the given samples, either in the specific knowledge points being tested or in the format.**
1-2. Start by drafting your question. If you discover any issues with the question or any overlapping parts between the generated question and the given samples during this process, feel free to revise it.
1-3. Think through what the correct answer should be. If you discover any issues during this process, repeat the entire Stage 1 process from the beginning.
1-4. Identify the plausible and potentially misleading incorrect options that could serve as distractors (at least nine). If you discover any issues during this process, repeat the entire Stage 1 process from the beginning.
1-5. Reevaluate your proposed question, answer and options to ensure that: the question meet the given attributes and Difficulty Description (you should compare the generated samples and given samples to verify this); the answer is both correct and unique. If it does not meet these criteria or you are not sure about this, repeat the entire Stage 1 process from the beginning.

**Stage 2: Generate Sample**
In this stage, you should give your generated sample in the right template based on the analyses above.
2-1. Generate the question content, begin with "##Question:##".
2-2. Generate a step-by-step reasoning process and the corresponding correct answer. Begin with "##Reasoning Path:##". If you find an issue with the question, return to Step 2-1 to regenerate the question.
2-3. Generate {{OptionNum}} candidates, with only one as the right option. Begin with "##Candidates:##".
2-4. Generate the index of the right option, begin with "##Right Option:##".

### Output Description:
Strictly follow the template below to generate your sample.
**Template**
##Analyses:## {{You analyze the provided attributes and outline the process for constructing the question to be generated.}}
##Question:## {{Your generated question content}}
##Reasoning Path:## {{Your step-by-step reasoning process}}
##Candidates:##
{{CandidatesDemo}}
##Right Option:##{{Index of the right option, e.g., B}}
**Template End**


Attention: You need to **strictly follow the template** and don't generate any other contents. Begin your response with "##Analyses:##\n1-1. "
```

K. Assessment Demands List

MATH:

```
Subset Name: Prealgebra
Assessment Demands:Prealgebra

Subset Name: Algebra
Assessment Demands:Algebra

Subset Name: Number Theory
Assessment Demands:Number Theory
```

```
Subset Name: Counting & Probability
Assessment Demands:Counting & Probability

Subset Name: Geometry
Assessment Demands:Geometry

Subset Name: Intermediate Algebra
Assessment Demands:Intermediate Algebra

Subset Name: Precalculus
Assessment Demands:Precalculus
```

MMLU-Pro:

```
Subset Name: psychology
Assessment Demands:This benchmark is designed to assess **psychology** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format

Subset Name: philosophy
Assessment Demands:This benchmark is designed to assess **philosophy** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format

Subset Name: health
Assessment Demands:This benchmark is designed to assess **health** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format

Subset Name: history
Assessment Demands:This benchmark is designed to assess **history** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format

Subset Name: business
Assessment Demands:This benchmark is designed to assess **business** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format

Subset Name: physics
Assessment Demands:This benchmark is designed to assess **physics** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format

Subset Name: engineering
Assessment Demands:This benchmark is designed to assess **engineering** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format

Subset Name: chemistry
Assessment Demands:This benchmark is designed to assess **chemistry** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format

Subset Name: math
Assessment Demands:This benchmark is designed to assess **math** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format

Subset Name: computer science
Assessment Demands:This benchmark is designed to assess **computer science** abilities
    while simultaneously evaluating knowledge understanding and complex reasoning skills,
    using **ten multiple-choice questions** as the evaluation format

Subset Name: biology
```

```
Assessment Demands:This benchmark is designed to assess **biology** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format

Subset Name: economics
Assessment Demands:This benchmark is designed to assess **economics** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format

Subset Name: law
Assessment Demands:This benchmark is designed to assess **law** abilities while
    simultaneously evaluating knowledge understanding and complex reasoning skills, using
    **ten multiple-choice questions** as the evaluation format
```

HellaSwag:

```
Subset Name: NLI
Assessment Demands:The task is to evaluate the model's commonsense natural language
    inference ability. Specifically, each question should present a concrete scenario, and
     the model should select the most likely event from the options based on a series of
    inferences.
```