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

Peiwen Yuan[1], **Shaoxiong Feng**[2], **Yiwei Li**[1], **Xinglin Wang**[1], **Yueqi Zhang**[1]
Jiayi Shi[1], **Chuyi Tan**[1], **Boyuan Pan**[2], **Yao Hu**[2], **Kan Li**[1]*
[1]School of Computer Science and Technology, Beijing Institute of Technology
[2]Xiaohongshu Inc
{peiwenyuan,liyiwei,wangxinglin,zhangyq,shijiayi,tanchuyi}@bit.edu.cn
{likan}@bit.edu.cn {shaoxiongfeng2023}@gmail.com
{panboyuan,xiahou}@xiaohongshu.com

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, generic and efficient benchmark generators that can construct high-quality benchmarks are widely needed. However, human annotators are constrained by inefficiency, and current LLM-based benchmark generators lack not only generalizability but also a comprehensive evaluation framework for validation and optimization. To fill this gap, we first establish an automated 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. On this basis, 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 comparable performance to human-annotated benchmarks on most metrics, highlighting its generalizability and validity. More importantly, it delivers highly consistent evaluation results across 21 LLMs (e.g., 0.969 Pearson correlation against MMLU-Pro on language understanding task), while incurring minimal overhead (e.g., $0.005 and 0.38 minutes per sample if using GPT-4o mini as generator). See our codes in https://github.com/ypw0102/BenchMaker.

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 pace of releasing available LLMs 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.

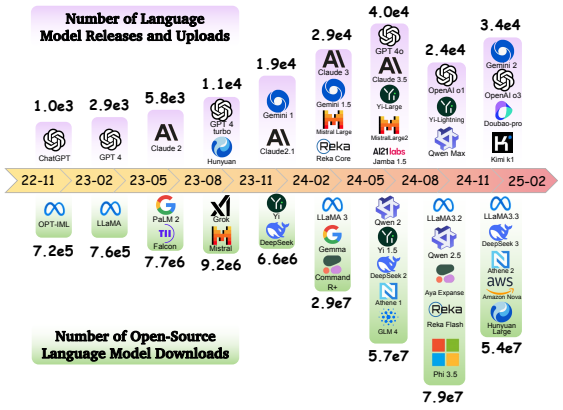


Figure 1: The trends of LLMs released and uploaded, and open-source LLMs downloads per season since the debut of ChatGPT. We obtain the data via the Huggingface API (Appendix E).

However, current benchmark construction processes largely rely on human-provided signals (Chang et al., 2024; Wang et al., 2024c), 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 generalizability across tasks. Meanwhile, the absence of a comprehensive evaluation framework hinders the assessment and optimization of benchmark generators, weakening confidence in LLM-generated benchmarks for real applications. Hence, a comprehensive evaluation framework and a generic LLM-based benchmark generator that can handle any assessment demands and efficiently generate high-quality samples are urgently needed.

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 35882–35903
July 2-7, 2026

To this end, we first establish an automatic evaluation framework with ten criteria for LLM benchmark generators. Notably, we identify the sources of bias in evaluating label correctness and task relevance using LLM-as-a-judge (Liu et al., 2023) through correlation analysis, and remove them using Multiple Regression. On this basis, we examine the strengths and weaknesses of naively prompting LLM as generic benchmark generator under this evaluation framework. The results reveal that the generated benchmarks exhibit limited lexical and semantic diversity, poor controllability over difficulty, and low label correctness, while showing advantages in high task relevance and behavior diversity. Bearing this in mind, we develop an LLM-based generic benchmark generation method **BENCHMAKER** by integrating existing techniques with newly designed approaches to address the identified issues. Specifically, BENCHMAKER: strengthens label correctness 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 from BENCHMAKER exhibit better task relevance and difficulty controllability, more challenging question difficulty, and comparable diversity, albeit with slightly lower label correctness. More importantly, they yield highly consistent evaluation results across 21 LLMs (e.g., 0.969 Pearson correlation with MMLU-Pro), while take minimal overhead (e.g., $0.005 and 0.38 minutes per sample when using GPT-4o mini as generator). We further perform detailed experiments to validate the generalizability 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 label correctness may be imperfect, further enhancing the practicality of BENCHMAKER in Appendix A.

2 Backgrounds

2.1 Data Synthesis

The growth of LLMs' abilities has led to widespread research on LLM-driven data synthesis, which demonstrates much better quality and controllability over traditional methods (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 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 framework for benchmark generators. As for implementation, current LLM-based benchmark generators (Perez et al., 2023; Wu et al., 2024; Zhu et al., 2024b; Lei et al., 2023; Li et al., 2025) are constrained by their dependence on seed benchmarks and task specific designs, preventing them from being generic. We construct a comprehensive evaluation framework and develop generic benchmark generation method BENCHMAKER to fill these gaps.

2.2 Potential Applicable Scenarios of BENCHMAKER

Given arbitrary assessment demands X (e.g., task description, sample type) as input, an automatic generic benchmark generator $\mathcal{G}$ 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, developing BENCHMAKER holds significant importance for both scientific research and practical applications within the NLP community.

3 Benchmarking Benchmark Generator

While the effectiveness of synthetic training data can be directly evaluated through the trained models, synthetic benchmarks demand a more thorough and multifaceted assessment to ensure their reliability. To this end, we consolidate insights from existing studies to establish a comprehensive ten-criteria evaluation framework for benchmark generators.

3.1 Validity

Two key criteria for ensuring the validity of a benchmark are **Label Correctness** and **Task Relevance**. Label correctness measures the accuracy of the generated samples' label (Wu et al., 2024), while task relevance assesses whether the samples effectively target the capability specified by the assessment demands X (Perez et al., 2023). For these criteria, previous studies rely on human evaluation (Zhu et al., 2024b) or LLM-as-a-judge (Zheng et al., 2023). However, the former lacks automation, and the latter may introduce bias (Thakur et al., 2024).

To this end, we seek to detect and mitigate possible biases of LLM-as-a-judge that may exist within the framework. We choose OpenAI o3-mini (OpenAI, 2025) as the judge for preliminary experiments with scoring range as $[0, 1]$ (See prompts for LLM-as-a-judge in Appendix M). Experiments are conducted on the high-quality MATH benchmark (Hendrycks et al., 2021b), where each sample is expected to achieve perfect label correctness and task relevance. Ideally, the scores assigned by the judge should not exhibit any consistency with specific factors. However, as shown in Figure 2-(a), both label correctness and task relevance 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 possible causal chain: *harder questions lead to longer rationales, requiring judges to conduct lengthier analyses. For label correctness, longer analyses increases the likelihood of judge errors, resulting in lower label correctness ratings. While for task relevance, longer analyses increases the probability of task-relevant words appearing and results in higher task relevance 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 the two criteria. 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 Qwen Plus (Yang et al., 2024) serve as the judge (Figure 7).

Based on the analysis above, the observed biases of the LLM judge are 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. When calculating label correctness (and similarly task relevance), we first obtain the mean score of $\mathcal{G}_i$ judged by the LLM and treat it as the dependent variable $f(i)$. We then set the generator indices as dummy variables β_i and include judge length as a covariate:

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

Here, β_{len} quantifies the effect of judge length bias. After fitting, β_i represents the average debiased score of $\mathcal{G}_i$, which serves as the metric for label correctness. ϵ is an offset that adjusts the human-annotated benchmark's β_i to 1. Thus, the score of $\mathcal{G}_i$ may exceed 1 if it outperforms the human-annotated counterpart.

3.2 Diversity

The diversity of the benchmark determines the extent to which evaluation results can reflect the true model capability across the assessed domain.

Lexical Diversity reflects vocabulary richness in benchmarks (Yu et al., 2023). Traditional metrics like vocabulary size and self-BLEU (Zhu et al., 2018) used in Wu et al. (2024); Yu et al. (2023) are biased by sample length (Guo and 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 (Chan et al., 2024). We calculate the average Euclidean distance between semantic embeddings of samples as the metric. Specifically, we use text-embedding-ada-002 (OpenAI, 2022) as embedding model for experiments.

Behavior Diversity measures whether different samples lead to different model behaviors (Vivek et al., 2024). If a list of evaluated models (denoted as $\mathcal{M}_{1:|\mathcal{M}|}$, see Appendix I for detailed list) present the same correctness pattern across two samples (e.g., [✓, ✗, ✓, ✓, ✗] for both), the two samples are essentially interchangeable. We represent the behavior embedding of each sample by the correctness pattern vector across the evaluated models on it. Given the binary nature of this embedding ([0,1]), we quantify behavior diversity using the

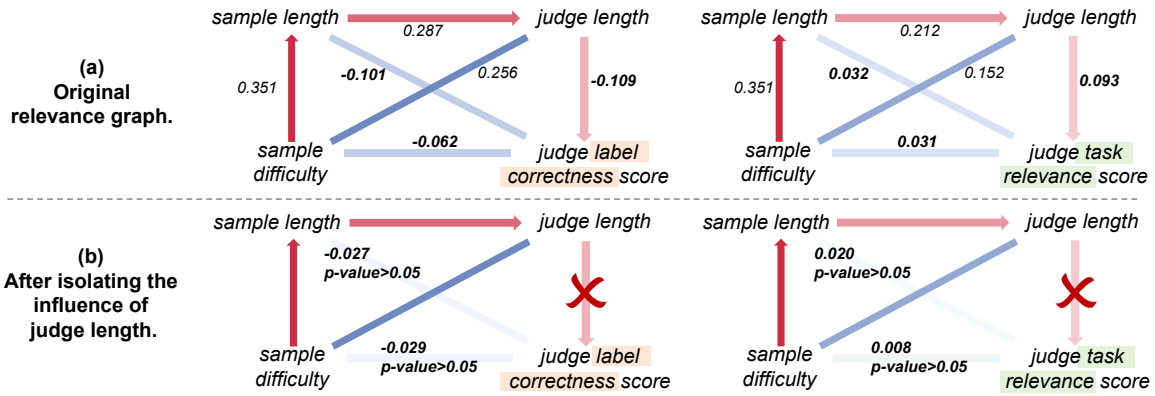


Figure 2: Pearson correlations among key factors of benchmark evaluation and LLM judge scores (label correctness and task relevance, OpenAI o3-mini as the judge). The most relevant path of each subject is highlighted in red.

average pairwise Hamming distance (Hamming, 1950) among samples. A higher value indicates greater sample irreplaceability.

3.3 Difficulty

The difficulty attribute of benchmarks is particularly significant in an era of rapid model evolution. **Difficulty Controllability** refers to assigning accurate 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 from the benchmark and the ground truth as the metric for difficulty controllability.

Difficulty Boundary denotes the difficulty of the hardest subset of a benchmark (Patel et al., 2025). With the growing strength of LLMs, the performance of the most advanced LLMs on simpler benchmarks has reached saturation (Hendrycks et al., 2021a), making it difficult to differentiate their capabilities. Consequently, more challenging benchmarks (Wang et al., 2024c) 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 certain benchmark to measure its difficulty boundary.

3.4 Benchmark-Level Metrics

Lastly, benchmark-level metrics are used to holistically assess the benchmark generation methods.

Effectiveness. While the earlier criteria assess benchmark from various aspects, a unified metric is required to measure benchmark effectiveness. Taking high-quality human-annotated benchmark as a reference, we examine if generated benchmark under identical assessment demands can deliver equivalent evaluation results. We calculate the accuracies of $\mathcal{M}_{1:|\mathcal{M}|}$ on both generated and human benchmarks and calculate their Pearson correlation as the effectiveness metric (Perlitz et al., 2024).

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 the original assessment demands and that rewritten by an LLM (we use GPT-4o (Hurst et al., 2024) for experiments), 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 compared. We further discuss the significance of evaluating benchmark quality from multiple dimensions in Appendix C.

4 Development of BenchMaker

In this section, we first analyze the pros and cons of directly prompting the LLM $\bar{\mathcal{M}}$ as generic benchmark generator (with assessment demands X as the sole input) in §4.1. On this basis, we refine its weaknesses in the following sections, leading to the development of BENCHMAKER.

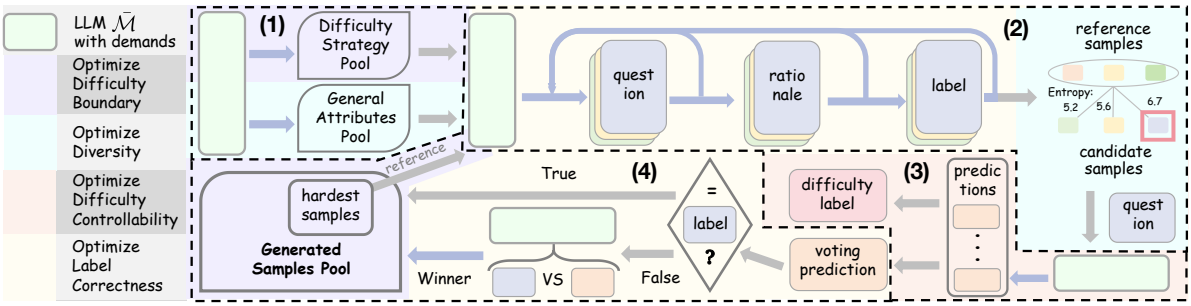


Figure 3: Workflow of BENCHMAKER: (1) Generator model $\bar{\mathcal{M}}$ takes the assessment demands X to generate an attribute pool and a difficulty strategy pool; (2) Given X, sampled attributes, difficulty strategy, and hardest samples, $\bar{\mathcal{M}}$ applies *Stepwise Self-correction*, *Difficulty Diffusion Mechanism*, *Difficulty Strategy Guidance*, and *Attribute-based Generation* to generate L candidate samples, from which the most diverse one is selected via *In-batch Diversity Boosting*; (3) The generated question is input into $\bar{\mathcal{M}}$ to produce T predictions, based on which γ is computed as the difficulty label; (4) The label is further refined using *Conflict Guided Contrastive Discrimination*.

4.1 Pros and Cons of Directly Prompting

We select three tasks for our experiments, each associated with a high-quality human-annotated benchmark for comparison: Math Reasoning (MATH (Hendrycks et al., 2021b)), Language Understanding (MMLU-Pro (Wang et al., 2024c)), and Commonsense Reasoning (HellaSwag (Zellers et al., 2019)). Following the sample format of human-annotated benchmarks, the first task adopts a free-form question-answering format, while the latter two take the form of multiple-choice questions. The evaluation capabilities and sample format are conveyed to generator LLM $\bar{\mathcal{M}}$ through the assessment demands X, as shown in Appendix O. We adopt the prompt_{base} in Appendix M to guide $\bar{\mathcal{M}}$ (GPT-4o mini by default) in generating credible and diverse samples $s_{1:|\mathcal{D}_{human}|}$:

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

where q_i, r_i, a_i denote question, rationale, and label, respectively. We proportionally adjust the difficulty level l in the prompt following the difficulty definition in Hendrycks et al. (2021a) (see descriptions in Appendix H), and select samples with the highest difficulty level to form the hardest subset. As shown in Table 1, compared to $\mathcal{D}_{human}$, directly prompting LLM as generic benchmark generator demonstrates poorer label correctness, lower lexical and semantic diversity, weaker difficulty controllability, and less challenging subset. Meanwhile, we also observe its advantages in better task relevance*, greater behavior diversity, and improved efficiency.

*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.

4.2 Label Correctness Optimization

To enhance label correctness, 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 label correctness.

Stepwise Self-correction. Since errors might occur at any step during the generation of $\{q_i, r_i, a_i\}$, we instruct $\bar{\mathcal{M}}$ to validate the content at each step. If an error is detected, $\bar{\mathcal{M}}$ 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 $\bar{\mathcal{M}}$ not only act as a judge but also as a test-taker to identify potential errors. Let $\bar{\mathcal{M}}$ predicts 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 $\bar{\mathcal{M}}$ conduct a contrastive discrimination between r_i and $\dot{r}_i$ to determine the final rationale and label for s_i.

4.3 Difficulty Optimization

Difficulty Controllability. From §4.1, we know that the LLM's ability to control the difficulty of generated samples is limited. In particular, for the language understanding task, the Spearman correlation between the actual and estimated difficulty of the samples is near-zero. To further explore this,

we examine LLM's difficulty perception by asking $\bar{\mathcal{M}}$ 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:

$$\gamma_i = \frac{1}{T}\sum_{j=1}^{T} \mathbf{1}_{\bar{a}_i^j \neq a_i} \tag{3}$$

By taking the inconsistency between $\bar{\mathcal{M}}$'s predictions $\bar{a}_i^{1:T}$ and label a_i as difficulty label, the correlation increases to 0.415, suggesting that γ is more reliable for estimating difficulty label.

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

Difficulty Strategy Guidance. We further consider providing $\bar{\mathcal{M}}$ with task-specific difficulty-control strategies. Specifically, we first instruct $\bar{\mathcal{M}}$ to give varying strategies for generating samples of specific difficulty levels based on assessment demands X (see examples in Appendix N). For example, difficult samples generally require more reasoning steps. With Difficulty Diffusion Mechanism, we progressively introduce more difficult difficulty-control strategies to $\bar{\mathcal{M}}$ to further extend the difficulty boundary.

4.4 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 **Attribute-based Generation Technique** introduced in AttrPrompt (Yu et al., 2023) for BENCHMAKER. This method 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 $\bar{\mathcal{M}}$ 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 run extensive experiments to evaluate BENCHMAKER under the proposed evaluation framework.

Settings. We select the widely used human-annotated MATH (Hendrycks et al., 2021b) (mathematical reasoning), MMLU-Pro (multi-task language understanding) (Wang et al., 2024c) 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[†] and HellaSwag, we write simple assessment demands X respectively (see details in Appendix O) as inputs for the generator model $\bar{M}$. For each demand, we generate N (default as 500) samples and randomly downsample the human-annotated benchmark to match the number of generated samples for fair comparison (when calculating effectiveness, we avoid downsampling to ensure more accurate results). Each experiment is repeated three times, and the average results are reported. We experiment with GPT-4o mini (Hurst et al., 2024) as the default $\bar{M}$ and also explore the performance of GPT-4o and Claude 3.5 Haiku (Anthropic, 2024). The decoding temperature is set to 1. To mitigate the self-enhancement bias (Zheng et al., 2023) of LLM-as-a-judge, we substitute the generators with OpenAI o3-mini (OpenAI, 2025) as the judge. Following the guidelines of Perlitz et al. (2024), we use 21 LLMs listed in Appendix I as models under evaluation. We choose the direct prompt method introduced in §4.1 and the widely used AttrPrompt (Yu et al., 2023) method as baselines.

5.1 Comparison with Human-annotated Benchmark

Overall (Table 1), while benchmarks from BENCHMAKER exhibit slightly reduced label correctness, they perform on par with human-annotated benchmarks in lexical and semantic diversity, and surpass them in task relevance, behavior diversity, difficulty controllability, and efficiency.

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 0.955 linear correlation (Pearson) and a

[†]Excluding the type 'other'.

Methods	Label Correct	TaskRelevance	LexicalDiv	SemanticDiv	BehaviorDiv	DifControl	DifBoundary	Effectiveness	Robustness	Efficiency
	Unbias Score↑	Unbias Score↑	Entropy↑	EuclideanDis↑	HammingDis↑	Spearman↑	Error Rate↑	Pearson↑	Pearson↑	$/item, min/item↓
Math Reasoning										
Human Benchmark	**1.000**	1.000	8.032	0.668	0.363	0.178	0.652	-	-	high
AttrPrompt	0.638	1.145	8.292	0.672	0.364	0.141	0.599	0.747	0.981	0.002, 0.19
Direct Prompt	0.692	1.120	7.105	0.621	0.366	0.115	0.570	0.654	0.989	0.002, 0.17
+InBatchDivBoost	0.660	1.097	8.627	0.676	0.365	0.172	0.548	0.780	0.982	0.003, 0.20
+StepSelfCorrect	0.928	1.098	8.640	0.675	0.377	0.170	0.481	0.792	0.984	0.003, 0.23
+ConflictConDisc	0.987	1.134	8.671	0.678	0.372	0.202	0.434	0.851	0.987	0.004, 0.36
+DiffControl	0.987	1.134	8.671	0.678	0.372	0.424	0.434	0.851	0.987	0.004, 0.36
+DiffDiffusion	0.971	1.159	8.694	0.678	0.390	0.468	0.601	0.895	0.990	0.005, 0.39
BenchMaker	0.922	1.151	**8.930**	**0.680**	0.407	**0.450**	0.650	0.931	**0.988**	**0.005, 0.42**
BenchMaker$_{4o}$	0.931	**1.174**	8.866	0.675	0.389	0.444	0.661	**0.933**	0.983	0.082, 1.10
BenchMaker$_{haiku}$	0.875	1.052	8.895	0.677	**0.412**	0.386	**0.684**	0.884	0.971	0.026, 0.56
Language Understanding										
Human Benchmark	1.000	1.000	**10.404**	**0.731**	0.311	0.000	**0.654**	-	-	high
AttrPrompt	0.867	1.185	9.916	0.735	0.390	0.079	0.561	0.890	0.988	0.002, 0.17
Direct Prompt	0.882	1.174	9.608	0.726	0.394	0.037	0.532	0.867	0.989	0.002, 0.16
BenchMaker	**1.032**	**1.211**	10.166	0.728	**0.397**	**0.461**	0.642	**0.969**	0.982	**0.005, 0.38**
Commonsense Reasoning										
Human Benchmark	**1.000**	1.000	**9.167**	0.655	0.371	0.000	0.481	-	-	high
AttrPrompt	0.884	1.064	8.763	0.658	0.392	0.121	0.580	0.858	0.974	0.002, 0.19
Direct Prompt	0.878	1.057	8.165	0.660	0.384	0.062	0.536	0.844	0.979	0.002, 0.17
BenchMaker	0.987	**1.084**	9.052	**0.663**	**0.404**	**0.445**	**0.634**	**0.946**	**0.984**	**0.005, 0.40**

Table 1: Overall results under the proposed evaluation framework. For each setting, We run 3 times and report the average results. GPT-4o mini serves as the default generator. Values in bold denote the best results between Human Benchmark and BENCHMAKER. The blue lines are results of baselines and the yellow lines are ablation studies.

remarkable 0.967 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 0.987. 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.

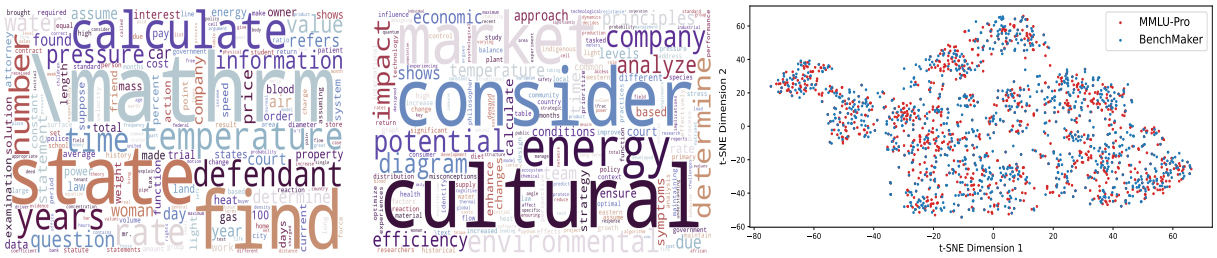
Generalizability. Experimental results demonstrate that BENCHMAKER exhibits strong generalizability across task types and generators, achieving an effectiveness of no less than 0.884 (Claude 3.5 Haiku). Compared to GPT-4o, GPT-4o mini proves to be a more cost-effective benchmark generator.

5.2 Ablation Studies

We validate the effectiveness of different techniques by sequentially integrating them to the Direct Prompt baseline. Since the number of ablation settings is large and the computational cost per setting is substantial, we choose the Math Reasoning task for ablation studies, as shown in Table 1.

Diversity. Compared to Direct Prompt, both Attribute-based Generation Technique and In-batch Diversity Boosting enhance lexical and semantic diversity. Noticeably, behavior diversity stays constant, suggesting that surface-level variation does not inherently result in a sample distribution more capable of distinguishing between model behaviors. Meanwhile, the diversity improvement leads to a slight drop in label correctness, possibly because of the attributes constraints.

Label Correctness. After applying Stepwise Self-correction and Conflict Guided Contrastive Discrimination, we observe a sustained improvement in label correctness. Meanwhile, we notice that the difficulty of the hardest subset decreases from 0.548 to 0.434. This may be due to the fact that a



(a) MMLU-Pro (b) BENCHMAKER (c) T-SNE VISUALIZATION

Figure 4: (a)-(b): Word cloud of MMLU-Pro and the benchmark generated by BENCHMAKER under similar assessment demands. (c): T-SNE results on the text embeddings of both benchmarks.

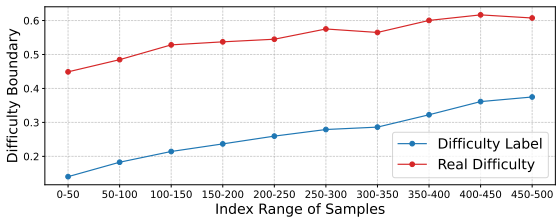


Figure 5: Trends of real and annotated difficulty over the index.

high label error rate often results in performance being underestimated. 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 0.424). Considering the 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.

Difficulty Boundary. With the proposed Difficulty Diffusion Mechanism and Difficulty Strategy Guidance, the difficulty boundary is significantly extended, as evidenced by an increase in error rate from 0.434 to 0.650. Additionally, we examine how actual difficulty and difficulty label change with generation order. As shown in Figure 5, both increase steadily, confirming the effectiveness of the Difficulty Diffusion Mechanism.

5.3 A Closer Look at Generated Benchmark

After metric analysis, we perform a more thorough examination of BENCHMAKER. Some of the generated samples are shown in Appendix K and the model performance on generated benchmarks are shown in Appendix F.

Lexical and Semantic. First, despite the obvious differences in word distribution between the generated benchmark and MMLU-Pro (Figure 4), it remains closely aligned with the domains covered by MMLU-Pro, demonstrating strong task relevance. Meanwhile, the semantic alignment between the two is more pronounced (Figure 4c). Notably, the input demands (Appendix O) 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.

Sample Number N	125	250	375	500
Effectiveness	0.898	0.940	0.961	0.969

Table 2: The impact of dataset size N.

Actual Label Correctness. While the detected potential bias of LLMs in judging label correctness has been addressed, we still carry out a manual review on 80 randomly chosen samples. We find that 3 samples have incorrect labels, 3 samples' questions are confusing, 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 label correctness; (2) LLM-as-a-judge can serve as a partial proxy for human evaluation.

Benchmark Size Analyses. We investigate the impact of dataset size N on Language Understanding task. As shown in Table 2, with the increases in N, the improvement in effectiveness gradually slows down, and acceptable effectiveness is already achieved at relatively small values of N. Therefore, in practical applications, we recommend selecting an appropriate N based on budget constraints.

Conclusions

The rapid evolution of LLMs has driven an urgent demand for an automated generic benchmark generator. To this end, we propose a comprehen-

sive framework for benchmark generator evaluation, based on which we develop BENCHMAKER method for reliable, generic, and efficient benchmark generation. Experiments across multiple tasks and LLMs demonstrate that BENCHMAKER achieves human-aligned benchmark quality, with superior efficiency and generalizability.

Limitations

Although we have verified the generalizability of BenchMaker across multiple settings (3 tasks, 3 generator LLMs, 21 tester LLMs), it still does not cover all scenarios. We anticipate that BenchMaker can undergo more extensive evaluation in real-world settings, which will help us further optimize it in the future.

We emphasize that the primary goal of this work is to systematically examine the current landscape of benchmark synthesis with LLMs and to provide initial mitigation strategies for the key issues observed in practice. We believe that designing a truly reliable and generic framework for LLM-based benchmark synthesis will require sustained efforts from the broader community. This includes, but is not limited to, (i) achieving better evaluation-goal alignment through iterative system–human interaction, (ii) identifying, quantifying, and mitigating potential biases introduced during synthesis and evaluation, and (iii) striking a principled balance among controllability, cost, and scalability. These directions are beyond the scope of the present study. Nevertheless, we view BenchMaker as a foundational step that offers reusable insights and practical guidance, and we hope it can help catalyze further advances in this emerging area.

Ethics Statement

All of the datasets used in this study were publicly available. 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. During the research process, AI was used solely to assist with code development and to help identify and correct writing errors in the manuscript; no AI usage prohibited by relevant policies was employed.

Acknowledgments

This work is supported by the Beijing Natural Science Foundation, China (Nos. 4262065, 4222037 and L181010).

References

Anthropic. 2024. Claude 3.5. https://www.anthropic.com/news/3-5-models-and-computer-use.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. 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*, pages 67–93. Association for Computational Linguistics.

Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *CoRR*, abs/2406.20094.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3):39:1–39:45.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, and 5 others. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in AI. *CoRR*, abs/2411.04872.

Xiaobo Guo and Soroush Vosoughi. 2023. 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*, pages 15869–15879. Association for Computational Linguistics.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. 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.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. 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.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 1827–1843. Association for Computational Linguistics.

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. *CoRR*, abs/2310.15147.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. 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*.

Jiatong Li, Renjun Hu, Kunzhe Huang, Yan Zhuang, Qi Liu, Mengxiao Zhu, Xing Shi, and Wei Lin. 2024a. Perteval: Unveiling real knowledge capacity of llms with knowledge-invariant perturbations. *CoRR*, abs/2405.19740.

Sihang Li, Jin Huang, Jiaxi Zhuang, Yaorui Shi, Xiaochen Cai, Mingjun Xu, Xiang Wang, Linfeng Zhang, Guolin Ke, and Hengxing Cai. 2024b. Scilitllm: How to adapt llms for scientific literature understanding. *CoRR*, abs/2408.15545.

Xiang Lisa Li, Farzaan Kaiyom, Evan Zheran Liu, Yifan Mai, Percy Liang, and Tatsunori Hashimoto. 2025. Autobencher: Towards declarative benchmark construction. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Zenan Li, Zhi Zhou, Yuan Yao, Yu-Feng Li, Chun Cao, Fan Yang, Xian Zhang, and Xiaoxing Ma. 2024c. Neuro-symbolic data generation for math reasoning. *CoRR*, abs/2412.04857.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. 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*, pages 2511–2522. Association for Computational Linguistics.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. 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*, pages 11065–11082. Association for Computational Linguistics.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Qingwei Lin, Jianguang Lou, Shifeng Chen, Yansong Tang, and Weizhu Chen. 2024a. Arena learning: Build data flywheel for llms post-training via simulated chatbot arena. *CoRR*, abs/2407.10627.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2024b. 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.

Gaurav Maheshwari, Dmitry Ivanov, and Kevin El Haddad. 2024. Efficacy of synthetic data as a benchmark. *CoRR*, abs/2409.11968.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. *CoRR*, abs/1904.03971.

OpenAI. 2022. text-embedding-ada-002. `https://platform.openai.com/docs/guides/embeddings`.

OpenAI. 2025. Openai o3 and o4-mini system card.

Arkil Patel, Siva Reddy, and Dzmitry Bahdanau. 2025. How to get your llm to generate challenging problems for evaluation. *arXiv preprint arXiv:2502.14678*.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, and 44 others. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13387–13434. Association for Computational Linguistics.

Yotam Perlitz, Ariel Gera, Ofir Arviv, Asaf Yehudai, Elron Bandel, Eyal Shnarch, Michal Shmueli-Scheuer, and Leshem Choshen. 2024. Benchmark agreement testing done right: A guide for LLM benchmark evaluation. *CoRR*, abs/2407.13696.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. 2024. A survey on self-evolution of large language models. *CoRR*, abs/2404.14387.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2024. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges. *CoRR*, abs/2406.12624.

Raphael Vallat. 2018. Pingouin: statistics in python. *J. Open Source Softw.*, 3(31):1026.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. 2024. Anchor points: Benchmarking models with much fewer examples. 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*, pages 1576–1601. Association for Computational Linguistics.

Ke Wang, Jiahui Zhu, Minjie Ren, Zeming Liu, Shiwei Li, Zongye Zhang, Chenkai Zhang, Xiaoyu Wu, Qiqi Zhan, Qingjie Liu, and Yunhong Wang. 2024a. A survey on data synthesis and augmentation for large language models. *CoRR*, abs/2410.12896.

Xinglin Wang, Yiwei Li, Shaoxiong Feng, Peiwen Yuan, Boyuan Pan, Heda Wang, Yao Hu, and Kan Li. 2024b. Integrate the essence and eliminate the dross: Fine-grained self-consistency for free-form language generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11782–11794, Bangkok, Thailand. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. 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.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. 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*, pages 13484–13508. Association for Computational Linguistics.

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

Siyuan Wu, Yue Huang, Chujie Gao, Dongping Chen, Qihui Zhang, Yao Wan, Tianyi Zhou, Xiangliang Zhang, Jianfeng Gao, Chaowei Xiao, and Lichao Sun. 2024. Unigen: A unified framework for textual dataset generation using large language models. *CoRR*, abs/2406.18966.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *CoRR*, abs/2407.10671.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. 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.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J. Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. 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*.

Peiwen Yuan, Yueqi Zhang, Shaoxiong Feng, Yiwei Li, Xinglin Wang, Jiayi Shi, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. 2025. Beyond one-size-fits-all: Tailored benchmarks for efficient evaluation. *arXiv preprint arXiv:2502.13576*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. 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*, pages 4791–4800. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. 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*.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2024a. 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.

Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. 2024b. Dyval 2: Dynamic evaluation of large language models by meta probing agents. *CoRR*, abs/2402.14865.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. 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*, pages 1097–1100. ACM.

A Results Credibility under Noised Labels.

Since label correctness of the generated benchmark cannot be ensured, we are curious about the effects of these noised samples: Assume a generated benchmark of size N contains a fraction K of randomly labeled samples. Given the presence of label noise, we suppose the probability that both models A and B agree with the provided labels on the corrupted samples is p, and their true accuracies on clean samples are θ_A and θ_B. The expected observed accuracies satisfy

$$\pi_A = (1-K)\theta_A + Kp, \;\; \pi_B = (1-K)\theta_B + Kp, \tag{4}$$

For large N we test $H_0 : \theta_A \le \theta_B$ versus $H_1 : \theta_A > \theta_B$ with the one-sided z-statistic (see details in Appendix A.1)

$$\begin{aligned} z &= \frac{\hat{\theta}_A - \hat{\theta}_B}{\sqrt{[\hat{\theta}_A(1-\hat{\theta}_A) + \hat{\theta}_B(1-\hat{\theta}_B)]/N}} \\ &\overset{\hat{\theta}=\frac{\hat{\pi}-Kp}{1-K}}{=\!=} \frac{(1-K)(\hat{\pi}_A - \hat{\pi}_B)}{\sqrt{(1-K)^2[\hat{\pi}_A(1-\hat{\pi}_A) + \hat{\pi}_B(1-\hat{\pi}_B)]/N}} \\ &= \frac{\hat{\pi}_A - \hat{\pi}_B}{\sqrt{(\hat{\pi}_A(1-\hat{\pi}_A) + \hat{\pi}_B(1-\hat{\pi}_B))/N}} \overset{H_0}{\sim} \mathcal{N}(0,1) \end{aligned} \tag{5}$$

This means that we can directly compute z using the observed accuracies $\hat{\pi}_A$ and $\hat{\pi}_B$, based on which the corresponding p-value can be calculated to determine whether the conclusion that model A outperforms model B (H_1) is credible. We also observe that the factor K vanishes completely in the computation of z, implying that a certain proportion of label noise in the benchmark will not affect the statistical significance of model ranking.

A.1 Proof of the one–sided z-statistic in Eq. (5)

Let N be the benchmark size, $K \in [0,1]$ the fraction of randomly labelled items, suppose that the probability that both models hit the corrupted items is p, θ_A, θ_B the true accuracies on the clean subset, and $\hat{\pi}_A, \hat{\pi}_B$ the empirically observed accuracies on the noisy benchmark. Throughout, hats denote sample estimates and subscripts A, B identify the two models.

***step 1*. Linking noisy and true accuracies**

Since only a proportion $1-K$ of the labels are correct, the expectations of the observed accuracies satisfy

$$\pi_A = (1-K)\theta_A + Kp, \;\; \pi_B = (1-K)\theta_B + Kp. \tag{6}$$

Solving for the latent accuracies yields

$$\theta_A = \frac{\pi_A - Kp}{1-K}, \qquad \theta_B = \frac{\pi_B - Kp}{1-K}. \tag{7}$$

***step 2*. Difference of true accuracies**

Subtracting the two equations in (6) eliminates p and gives

$$\theta_A - \theta_B = \frac{\pi_A - \pi_B}{1-K}.$$

Hence $\theta_A > \theta_B$ is equivalent to $\pi_A > \pi_B$; the noise ratio K plays no role in the sign of the difference.

***step 3*. Sampling distribution of $\hat{\pi}_A - \hat{\pi}_B$**

For large N each $\hat{\pi}$ is approximately normal by the Central Limit Theorem:

$$\hat{\pi}_A \sim \mathcal{N}\big(\pi_A, \tfrac{\pi_A(1-\pi_A)}{N}\big), \; \hat{\pi}_B \sim \mathcal{N}\big(\pi_B, \tfrac{\pi_B(1-\pi_B)}{N}\big),$$

and they are asymptotically independent. Therefore

$$\hat{\pi}_A - \hat{\pi}_B \sim \mathcal{N}\Big(\pi_A - \pi_B, \; \tfrac{\pi_A(1-\pi_A)+\pi_B(1-\pi_B)}{N}\Big).$$

***step 4*. One–sided test $H_0 : \theta_A \le \theta_B$ vs. $H_1 : \theta_A > \theta_B$**

Using $\hat{\pi}_A - \hat{\pi}_B$ as the test statistic and plugging the estimated means into the variance gives the empirical z-score reported in Eq. (5):

$$z = \frac{\hat{\pi}_A - \hat{\pi}_B}{\sqrt{\big[\hat{\pi}_A(1-\hat{\pi}_A) + \hat{\pi}_B(1-\hat{\pi}_B)\big]/N}} \overset{H_0}{\sim} \mathcal{N}(0,1).$$

Crucially, all factors of $(1-K)$ cancel; the presence of an arbitrary proportion K of noisy labels

does not alter the null distribution of z. The p-value of the one-sided test is therefore

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

where Φ is the standard normal CDF. A small p-value allows us to reject H_0 and conclude that model A outperforms model B on the underlying clean benchmark.

B Unsuccessful Attempts for Optimizing Benchmark Generator

B.1 Label Correctness

We explored the widely studied self-correction strategy to improve the label correctness 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 language understanding 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., 2024c) 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 Attr-Prompt, 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 Significance of a Comprehensive Evaluation Framework for Benchmark Generator

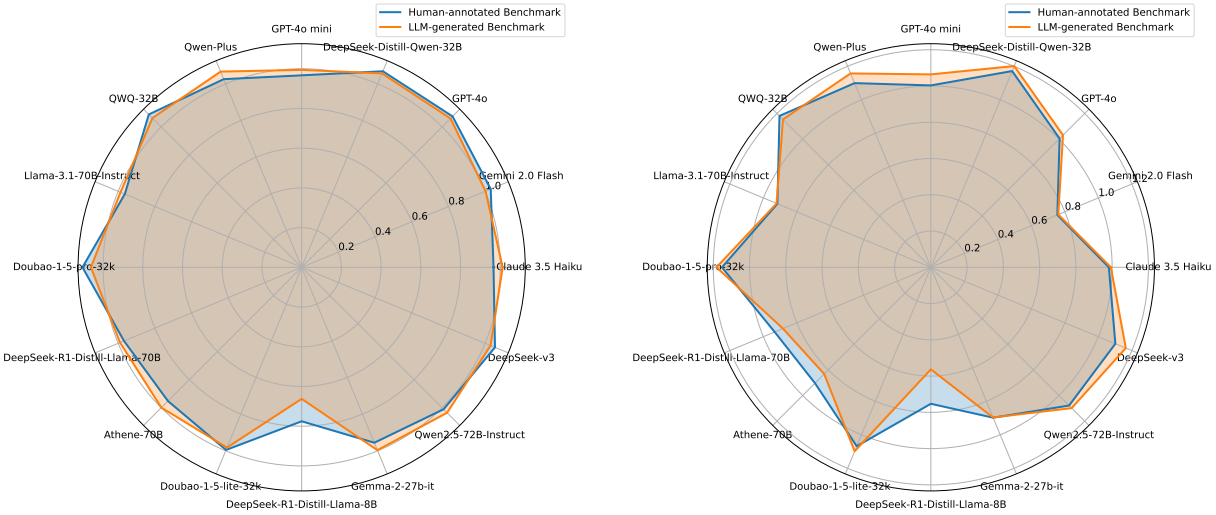
Different scenarios call for different requirements, so it is essential to assess benchmark quality from multiple dimensions.

- When developing benchmark generation methods, tasks with high-quality human-annotated datasets can serve as testbeds, allowing us to directly evaluate the quality of generated benchmarks via effectiveness. Efficiency and robustness are also critical, as they reflect the implementation efficiency and generation stability of different methods.
- When selecting appropriate models for customized scenarios without access to high-quality human annotations, label correctness, task relevance, and diversity become more important. These metrics allow users to quantify whether the generated benchmark accurately evaluates model capabilities under the desired conditions and whether it provides sufficient domain coverage.
- For evaluating state-of-the-art models, the concept of difficulty boundary helps determine whether a benchmark is sufficiently challenging and capable of distinguishing between models, thus indicating whether it has become saturated.
- In benchmark compression and efficient evaluation settings, difficulty controllability reflects the accuracy of the provided difficulty labels, while behavior diversity indicates sample irreplaceability. Accurate difficulty labels enable strategies such as uniformly downsampling based on difficulty or difficulty-based sampling tailored to model capabilities (Yuan et al., 2025), leading to more efficient evaluations.

D Results on Creative Writing Task

To enable BenchMaker to support open-ended tasks, we have implemented the following adaptations:

- We adopted the FSC (Wang et al., 2024b) technique to facilitate self-consistency voting for open-domain texts, utilizing an LLM-based discriminator to evaluate semantic consistency.



(a) Language Understanding

(b) Math Reasoning

Figure 6: LLM Performance on human-annotated and LLM-generated benchmarks.

Methods	Label Correct Unbias Score↑	TaskRelevance Unbias Score↑	LexicalDiv Entropy↑	SemanticDiv EuclideanDis↑	BehaviorDiv HammingDis↑	DifControl Spearman↑	DifBoundary Error Rate↑	Effectiveness Pearson↑	Robustness Pearson↑	Efficiency $/item, min/item↓
Human Benchmark	**1.000**	1.000	10.972	0.725	0.392	0.000	0.582	-	-	high
BenchMaker	0.991	1.072	10.825	0.728	0.437	0.422	0.536	0.972	0.991	0.107, 2.08

Table 3: Results under the proposed evaluation framework on Creative Writing Task.

- We employed an LLM-as-a-Judge approach to determine whether the generated content adheres to rubric-based criteria, thereby assigning scores to the model's predictions.
- The normalized average score is utilized as the difficulty label, achieving alignment with the methodology used for closed-domain tasks.

The experimental results are presented in Table 3. As demonstrated, BenchMaker achieves performance comparable to the high-quality WritingBench across various metrics, while demonstrating significant improvements in Task Relevance, Behavior Diversity, and Difficulty Control. We also observed that BenchMaker scores slightly lower than WritingBench in Difficulty Boundary. This suggests a promising future direction: integrating a RAG module into BenchMaker to incorporate external knowledge, which could further enhance the factuality and difficulty of the generated questions.

E 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.

F Model Performance on Generated Benchmarks

We present the performance of some mainstream LLMs on human-annotated and LLM-generated benchmarks to compare them from the evaluation effectiveness perspective, as shown in Figure 6. We normalize the accuracy of the evaluated models to have a mean of 1, facilitating comparison. As shown, despite some differences, the model-generated benchmark and the human-annotated benchmark yield aligned overall performance trends, demonstrating strong evaluation effectiveness.

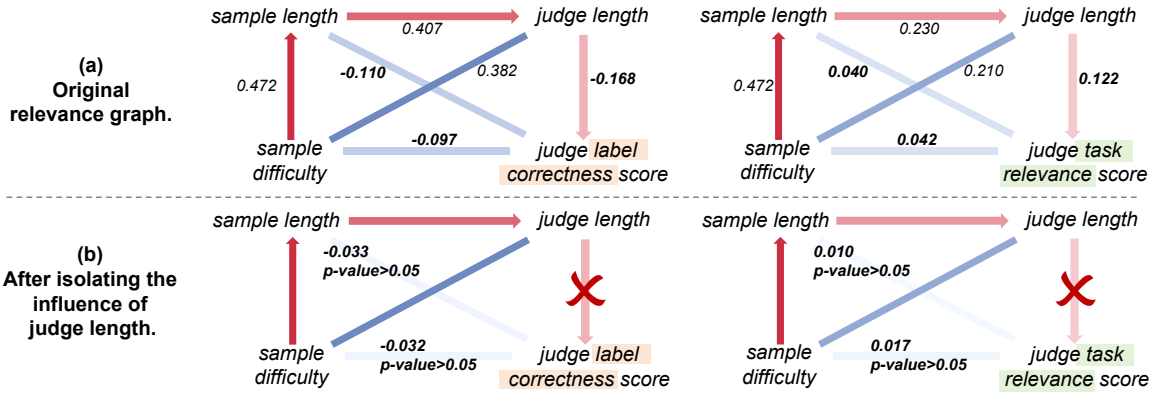


Figure 7: Pearson correlations among key factors of benchmark evaluation and LLM (GPT-4o mini) judge scores (faithfulness and alignment). The most relevant path of each subject is highlighted in red to show the possible causal chain.

G Human Evaluation Settings

Multiple authors jointly conducted the manual check for the Actual Error Rate section, and no extra annotators were employed for our data collection.

H Difficulty Levels

In Hendrycks et al. (2021a), the questions are categorized into the following four difficulty levels.

- **Elementary Level**: Basic grade-school questions
- **High School Level**: More challenging high-school curriculum questions
- **College Level**: Undergraduate-level questions
- **Professional Level**: Expert or graduate-level questions

I 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/
- **doubao-1-5-pro-32k-250115**: https://www.volcengine.com/product/doubao
- **DeepSeek-V3-0324**: https://huggingface.co/deepseek-ai/DeepSeek-V3-0324
- **doubao-1-5-lite-32k-250115**: https://www.volcengine.com/product/doubao
- **DeepSeek-R1-Distill-Llama-70B**: https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B
- **Qwen2.5-72B-Instruct**: https://huggingface.co/Qwen/Qwen2.5-72B-Instruct

- **Llama-3.1-70B-Instruct**: https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct
- **DeepSeek-R1-Distill-Llama-8B**: https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B
- **Athene-70B**: https://huggingface.co/Nexusflow/Athene-70B
- **gemma-2-27b-it**: https://huggingface.co/google/gemma-2-27b-it

J 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:

$$\text{Calibrate\_Difficulty} = \text{Difficulty\_Label} \times 0.9^{t_i/\text{Reference\_Nu}} \quad (8)$$

   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.

K 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
```

L Examples of Model Generated Descriptions on Math Reasoning (Algebra)

Task Description:

```
Evaluate the model's ability to solve
    algebra problems, including solving
    equations, simplifying expressions,
    and interpreting algebraic
    relationships.
```

Query Description:

```
A clearly stated algebraic problem that
    includes sufficient details and
    conditions for solving. The question
     should test the model's ability to
    manipulate variables, apply
    algebraic principles, solve
    equations, or interpret graphs and
    systems of equations. The problem
    may involve concepts such as linear
    equations, quadratic equations,
    inequalities, exponents, or word
    problems requiring algebraic
    modeling.
```

Label Format Description:

```
Includes one correct answer and several
    incorrect options. Incorrect options
     should reflect common mistakes,
    such as arithmetic errors,
    misunderstanding of algebraic rules,
     or misinterpretation of the problem
    . These options should be designed
    to test whether the model can
    reliably differentiate between valid
     and invalid approaches to solving
    the problem.
```

M Prompt List

LLM as Label Correctness 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 Label Correctness 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 Task 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 **Label Format 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,
    Label Format Description, Generation
     Guidelines, and Output Description
    to help build a benchmark that
    assesses the given task.

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

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

### Label Format Description:
{{label 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 the right label strictly
    following the Label Format
    Description, begin with "Right Label
    :".

### 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}}
##Right Label:##{{Your label to the
    question, strictly following the
    Label Format Description}}
**Template End**

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

N 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,
    Label Format 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}}


### Label Format Description:
{{label 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. Reevaluate your proposed question,
    answer 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 the right label to the
    question strictly following the
    Label Format Description, begin with
     "##Right Label:##".

### 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}}
##Right Label:##{{Strictly follow the
    Label Format Description to offer
    the right label here}}
**Template End**


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

O Assessment Demands List

Math Reasoning

```
Subset Name: Prealgebra
Assessment Demands: Construct Prealgebra
     test question with exactly one
    correct answer for each. Use \boxed
    {} to denote the correct label.

Subset Name: Algebra
Assessment Demands: Construct Algebra
    test question with exactly one
    correct answer for each. Use \boxed
    {} to denote the correct label.

Subset Name: Number Theory
Assessment Demands: Construct Number
    Theory test question with exactly
    one correct answer for each. Use \
    boxed{} to denote the correct label.

Subset Name: Counting & Probability
Assessment Demands: Construct Counting &
     Probability test question with
    exactly one correct answer for each.
     Use \boxed{} to denote the correct
    label.

Subset Name: Geometry
Assessment Demands: Construct Geometry
    test question with exactly one
    correct answer for each. Use \boxed
    {} to denote the correct label.

Subset Name: Intermediate Algebra
Assessment Demands: Construct
    Intermediate Algebra test question
    with exactly one correct answer for
    each. Use \boxed{} to denote the
    correct label.

Subset Name: Precalculus
Assessment Demands: Construct
    Precalculus test question with
    exactly one correct answer for each.
     Use \boxed{} to denote the correct
    label.
```

Language Understanding

```
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. Use \boxed{}
     to denote the correct label.

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. Use \boxed{}
     to denote the correct label.

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. Use \boxed{}
     to denote the correct label.

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. Use \boxed{}
     to denote the correct label.

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. Use \boxed{}
     to denote the correct label.
```

```
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. Use \boxed{}
     to denote the correct label.

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. Use \boxed{}
     to denote the correct label.

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. Use \boxed{}
     to denote the correct label.

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. Use \boxed{}
     to denote the correct label.

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
    . Use \boxed{} to denote the correct
     label.

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. Use \boxed{}
     to denote the correct label.

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. Use \boxed{}
     to denote the correct label.

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. Use \boxed{} to
    denote the correct label.
```

Commonsense Reasoning

```
Subset Name: NLI
Assessment Demands: The task is to
    evaluate the model's commonsense
    natural language inference ability,
    using **four multiple-choice
    questions** as the evaluation format
    . 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. Use \
    boxed{} to denote the correct label.
```