AutoBench: Automating LLM Evaluation through Reciprocal Peer Assessment

Dario Loi[1], Elena Maria Muià[2], Federico Siciliano[2], Giovanni Trappolini[2], Vincenzo Crisà[2], Peter Kruger[3], Fabrizio Silvestri[2],

[1]Department of Computer Science, Sapienza University of Rome
[2]Department of Computer, Control, and Management Engineering, Sapienza University of Rome
[3]eZecute S.R.L.

Correspondence: loi.1940849@studenti.uniroma1.it

arXiv:2510.22593v1 [cs.CL] 26 Oct 2025

Abstract

We present AutoBench, a fully automated and self-sustaining framework for evaluating Large Language Models (LLMs) through reciprocal peer assessment. This paper provides a rigorous scientific validation of the AutoBench methodology, originally developed as an open-source project by eZecute S.R.L.. Unlike static benchmarks that suffer from test-set contamination and limited adaptability, AutoBench dynamically generates novel evaluation tasks while models alternately serve as question generators, contestants, and judges across diverse domains. An iterative weighting mechanism amplifies the influence of consistently reliable evaluators, aggregating peer judgments into consensus-based rankings that reflect collective model agreement. Our experiments demonstrate strong correlations with established benchmarks including MMLU-Pro and GPQA (respectively 78% and 63%), validating this peer-driven evaluation paradigm. The multi-judge design significantly outperforms single-judge baselines, confirming that distributed evaluation produces more robust and human-consistent assessments. AutoBench offers a scalable, contamination-resistant alternative to static benchmarks for the continuous evaluation of evolving language models.

1 Introduction

Evaluating Large Language Models (LLMs) remains a central challenge in natural language processing. While traditional benchmarks such as MMLU (Hendrycks et al., 2021a) and MATH (Hendrycks et al., 2021b) have been instrumental in measuring performance across a range of subjects, they are inherently static: the task sets are fixed, models can overfit to them, and their diagnostic power diminishes as LLMs rapidly improve.

This paper provides a rigorous scientific validation of the AutoBench framework, an open-source, self-sustaining benchmark for LLMs originally developed by eZecute S.R.L. and detailed publicly[1]. This public-facing project has already demonstrated the methodology's effectiveness at scale; its third public run, for instance, evaluated 33 models over 400 iterations, collecting ∼300,000 individual ranks and showing exceptionally high correlation with established benchmarks, such as 92.17% with Artificial Analysis Intelligence Index, and 86.85% with LMSYS Chatbot Arena (Kruger, 2025a). This paper's contribution is to formalize and analyze the system's core methodology, built on a fully automated framework of reciprocal peer evaluation. In this system, models autonomously conduct the entire evaluation lifecycle, from generating tasks to delivering final judgments. The process begins with the collaborative creation of diverse tasks across varying topics and difficulty levels. Each model then answers all tasks, producing a shared pool of responses for assessment. In the core peer-evaluation phase, models judge both the tasks and answers of their peers, forming a directed, weighted graph where nodes represent models and edges encode assigned scores. An iterative weighting algorithm aggregates these judgments into authority scores, amplifying the influence of consistent evaluators and establishing a dynamic, self-regulating standard for model assessment.

The benchmark is *dynamic*, generating new prompts each iteration; *reciprocal*, as models act as both examiners and reviewers; and *consensus-driven*, with scores reflecting weighted collective agreement. This automation eliminates human supervision, prevents test contamination, and yields stable, interpretable leaderboards beyond the limits of fixed datasets or single-judge setups.

Our contributions are threefold:

1. A peer-evaluation protocol where LLMs both generate and assess questions and answers

[1]See the official project website (eZecute S.R.L., 2024), the Hugging Face organization (AutoBench, 2024), and the initial release announcement (Kruger, 2025b)

across diverse topics and difficulty levels.

2. An iterative weighting algorithm that aggregates peer judgments into robust, consensus-based rankings.

3. Empirical validation showing coherent, adaptive leaderboards, establishing a foundation for continuous, self-sustaining LLM evaluation.

2 Related Works

The landscape of LLM evaluation has evolved from static, human-curated benchmarks to more dynamic and model-driven approaches. We categorize existing work into three main areas, highlighting the progression toward the self-evolving, reciprocal peer evaluation framework we propose.

Traditional LLM evaluation has largely relied on static, ground-truth benchmarks designed to measure performance across diverse domains. These include well-established datasets like MMLU (Hendrycks et al., 2021a), a comprehensive 57-task multiple-choice suite, and MATH (Hendrycks et al., 2021b), comprising $12,500$ competition-level problems. Further examples encompass specialized benchmarks such as GSM-8K (Cobbe et al., 2021) for mathematical reasoning, HumanEval (Li and Murr, 2024) for code generation, and broader assessments GPQA (Rein et al., 2024). While instrumental for quantifying factual knowledge and problem-solving abilities, these static benchmarks face inherent limitations: they risk test set contamination as models are increasingly trained on web corpora, and their diagnostic power diminishes as LLMs improve, often struggling to provide fine-grained insights into emergent capabilities or open-ended generation.

To overcome the scalability and contamination issues of static benchmarks, a significant body of recent work leverages LLMs themselves as evaluators. This paradigm broadly divides into single-judge and multi-agent approaches. Single-judge methods typically employ a strong, often proprietary, LLM (e.g., GPT-4 (OpenAI, 2023)) to assess outputs. For instance, MT-Bench (Zheng et al., 2023) utilizes GPT-4 to compare answers to multi-turn queries, demonstrating substantial agreement with human preferences. Other efforts train LLMs for prompt-specific rankings (Prompt-to-Leaderboard(Frick et al., 2025)) or integrate human feedback into large-scale hybrid evaluations like Chatbot Arena (Chiang et al., 2024). Building on this, multi-agent schemes simulate panels of evaluators to achieve more robust judgments. ChatEval (Chan et al., 2023) involves multiple LLM "personas" or paired reviewers, often weighting judgments from more reliable agents. Auto-Arena (Zhao et al., 2024) introduces a multi-stage pipeline involving an examiner, debate, and judge committee, boosting alignment with human preferences over static tests. While these methods advance automated evaluation, they often rely on a centralized "strong" judge or pre-defined roles, and the aggregation of judgments can be based on fixed or learned weights rather than a dynamic, network-driven authority.

A complementary direction emphasizes the necessity of dynamic and evolving benchmarks to counter contamination and keep pace with rapid LLM advancements. Initiatives like DynaBench (Kiela et al., 2021) and LiveBench (White et al., 2025) advocate for "living" evaluations that continuously adapt. BenchBuilder (Li et al., 2024) exemplifies this by filtering Chatbot Arena queries to create Arena-Hard-Auto (Li et al., 2024), a 500-task benchmark achieving high agreement with human rankings at low cost. While these dynamic approaches mitigate dataset staleness, they remain limited in several ways: DynaBench continues to rely on human-in-the-loop task creation, LiveBench employs a fixed set of evaluator models, and Arena-Hard-Auto constructs its benchmark from filtered human–LLM interaction logs rather than autonomously generated tasks.

In contrast, our framework removes these dependencies through a fully automated, reciprocal peer-evaluation process. All participating models simultaneously generate tasks, produce answers, and assess one another, with judgments aggregated via an iterative weighting mechanism that amplifies reliable evaluators. This design achieves end-to-end automation without fixed roles or human oversight, yielding a self-evolving, contamination-resistant benchmark that continuously adapts alongside the LLMs it measures.

3 Methods

We introduce AUTOBENCH, a dynamic and self-contained framework for evaluating a set of language models, denoted by $\mathcal{M} = \{M_1, \ldots, M_n\}$. Each model can take on three distinct roles during evaluation: *task generator*, *contestant*, and *judge*.

The system evolves over a series of T iterations, progressively refining model evaluations to converge towards a stable consensus ranking.

At iteration t, the state of the system is represented by a judging weight vector $\mathbf{w}^{(t)} = (w_1^{(t)}, \ldots, w_n^{(t)})$, which quantifies the relative authority of each model when acting as a judge. Weights are initialized uniformly, such that $w_k^{(1)} = \frac{1}{n}$ for all models $k \in \{1, \ldots, n\}$.

At the start of each iteration, one model M_q is sampled from $\mathcal{M}$ to serve as the *task generator*. To ensure a comprehensive evaluation across diverse capabilities, tasks are drawn from a discrete set of predefined categories (see Section 4). The generated task, denoted by $q^{(t)}$, undergoes a quality assurance phase in which all models provide preliminary assessments. A task is accepted only if its mean and median weighted scores, computed using the current weight vector $\mathbf{w}^{(t)}$, exceed predefined thresholds λ_{mean} and λ_{median}, respectively. If no acceptable task is produced after k attempts, the process restarts with a newly sampled generator.

Once a task is accepted, every model in $\mathcal{M}$ participates in two roles: first as a *contestant*, producing a response, and then as a *judge*, evaluating the responses of all contestants. We represent these evaluations in a *judgment matrix* $J^{(t)} \in \mathbb{R}^{n \times n}$, where each row corresponds to a judge and each column to a contestant. The entry $J_{r,c}^{(t)}$ denotes the score given by the r-th judge to the c-th contestant. Because all models act in both roles, $J^{(t)}$ provides a complete pairwise assessment of the system.

At the conclusion of iteration t, an aggregate score vector $\mathbf{r}^{(t)}$ is computed for the contestants. Each element of $\mathbf{r}^{(t)}$ is obtained as the weighted sum of the corresponding column of $J^{(t)}$:

$$r_c^{(t)} = \sum_{r=1}^{n} w_r^{(t)} J_{r,c}^{(t)}. \tag{1}$$

This instantaneous score vector is then used to update the models' cumulative performance scores, $\mathbf{c}^t$, which are maintained as a running average:

$$\mathbf{c}^{(t)} = \frac{(t-1)\mathbf{c}^{(t-1)} + \mathbf{r}^{(t)}}{t} \tag{2}$$

The resulting values, in turn, normalized to derive the judging weights for the subsequent iteration:

$$w_j^{(t+1)} = \frac{c_j^{(t)}}{\sum_{k=1}^{n} c_k^{(t)}} \tag{3}$$

This iterative re-weighting mechanism ensures that models demonstrating consistently high performance gain greater influence in the evaluation process, thereby enhancing the robustness of the AUTOBENCH framework. A detailed pseudo-code of the full procedure is available in Algorithm 1.

4 Experiments

Experimental Setup We evaluate the AUTOBENCH framework presented in Section 3 on a suite of twelve widely available open-source models, detailed in Table 2. All models were sourced from the Hugging Face Hub[2] and are governed by open licenses that permit research use. Our methodology adheres to their intended use as instruction-following and chat models. The framework was executed for $T = 40$ iterations, with task quality enforced via score thresholds of $\lambda_{mean} = 3.5$ and $\lambda_{median} = 3.0$, allowing up to $k = 3$ generation retries. For all model responses, we used a temperature of $\tau = 0.8$ and nucleus sampling with $p_{top} = 0.9$ to encourage diverse yet coherent outputs (Holtzman et al., 2019).

To validate the robustness of AUTOBENCH, we measure how closely its emergent model rankings align with established human and task-based benchmarks, including MMLU-Pro (Hendrycks et al., 2021a), MATH (Hendrycks et al., 2021b), and GPQA (Rein et al., 2024). This comparison assesses whether AUTOBENCH produces rankings that are consistent with external evaluations of model capability. To quantify this alignment we use Kendall's τ and Spearman's ρ (Spearman, 2015; Kendall, 1938) as correlation statistics, the result of this study are reported in Table 1.

To isolate the impact of the multi-judge design, we perform an ablation in which only the best open-source model, `gpt-oss-20b`, serves as judge. This single-judge configuration is compared directly with the full multi-judge pipeline in Table 1, enabling a direct assessment of how collective judgment influences benchmark correlations.

Our task-generating prompts are conditioned to generate tasks belonging to a diverse set of predefined categories: logic, coding, technology, history, science, general culture, creative writing, grammar, current news, and math. This set can be effortlessly expanded, simply by providing new categories for the models to generate. Our experiments show results aggregated across all categories, per-category

[2] https://huggingface.co

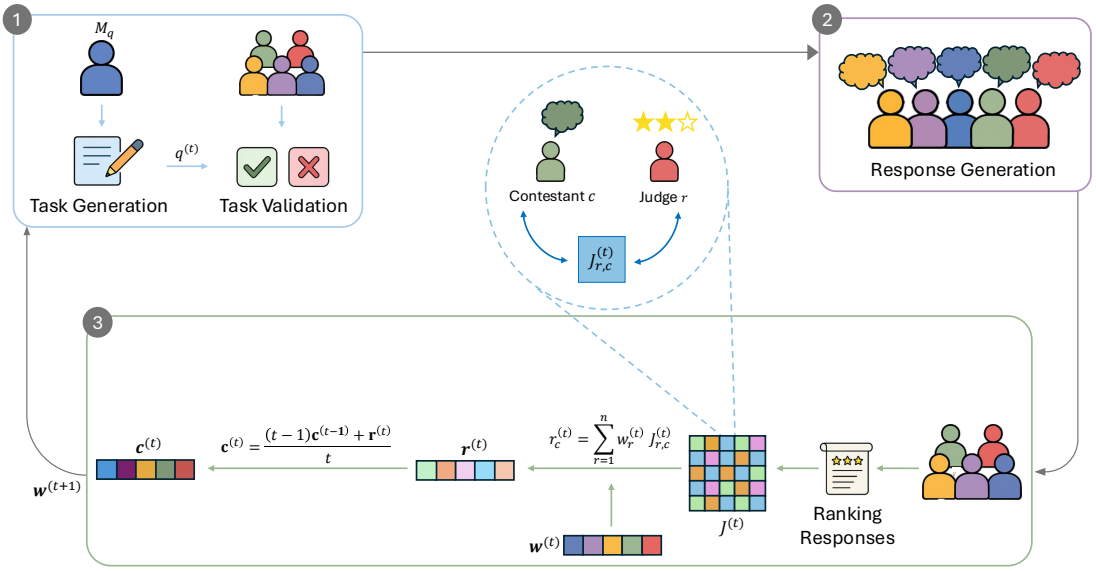


Figure 1: Diagram of the AUTOBENCH framework.

Compared benchmark	Multi-judge (AUTOBENCH)		Single-judge (`gpt-oss-20b`)	
	Kendall's τ	Spearman's ρ	Kendall's τ	Spearman's ρ
MMLU-Pro	**0.64**	**0.78**	0.45	0.61
GPQA	**0.52**	**0.63**	0.45	0.59

Table 1: **Correlation analysis: multi-judge vs single-judge configurations.** Kendall's τ and Spearman's ρ computed between AUTOBENCH model rankings and established benchmarks. The left block corresponds to the full multi-judge pipeline; the right block corresponds to an ablation using only `gpt-oss-20b` as judge.

rankings on a select number of topics is available in Table 4.

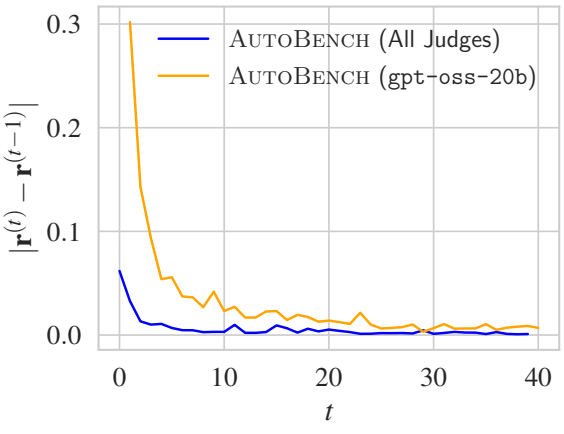


Figure 2: Convergence of weights over iterations, quantified by the L1 norm between successive evaluations.

Results and Discussion As shown in Table 1, AUTOBENCH yields rankings that are well-aligned with recognized human-validated benchmarks. The multi-judge configuration achieves substantially higher correlation than the single-judge baseline, confirming that collective evaluation leads to more robust and human-consistent judgments. Strong alignment with GPQA demonstrates the framework's ability to generalize beyond knowledge-based tests to assess deeper reasoning.

The weighting dynamics also show rapid convergence, as illustrated in Figure 2. The L1 norm between successive judging weights decreases quickly, indicating that the iterative process stabilizes as authority scores self-organize toward a consensus equilibrium. By contrast, the single-judge variant exhibits noisier convergence, underscoring the stabilizing effect of distributed judgment aggregation.

Comprehensive supplementary materials, including the aggregate cross-model judgment matrix (Figure 3), detailed per-topic results (Table 4), and all prompts, are available in the appendix.

5 Conclusions and Future Work

We introduced AUTOBENCH, a fully automated framework for LLM evaluation through recipro-

cal peer assessment. Our experiments validate this paradigm, demonstrating that its consensus-driven rankings correlate strongly with established benchmarks such as MMLU-Pro and GPQA. This confirms that a collective of LLMs can produce a reliable measure of capability without human supervision or a fixed ground truth. Crucially, our ablation study highlights the value of the multi-judge design, which proves more robust than a single-judge baseline by mitigating individual model biases.

By successfully automating the entire evaluation lifecycle, AUTOBENCH provides a scalable, contamination-resistant, and adaptive alternative to static test sets, establishing a viable paradigm for the continuous assessment of evolving language models.

6 Limitations

We acknowledge three primary limitations in the current design of AUTOBENCH. First, its evaluation is confined to open-source models. The dynamics of peer evaluation and authority convergence could differ significantly when including leading proprietary LLMs, which possess distinct capability profiles and training methodologies. Second, as a fully autonomous system, AUTOBENCH is susceptible to systemic bias. It risks creating an "echo chamber" where models reinforce shared weaknesses or converge on evaluation criteria that gradually diverge from human-defined notions of quality. Finally, our validation is indirect, relying on correlation with existing benchmarks. This approach, while informative, does not replace direct, large-scale human evaluation, which remains the gold standard for assessing nuanced aspects of quality like creativity, subtle reasoning, and factual accuracy.

Ethical Consideration

The autonomous nature of AUTOBENCH introduces key ethical considerations. The reciprocal peer-evaluation mechanism risks amplifying shared societal biases present in the participating models, as they may reward one another for biased responses. Furthermore, the framework currently lacks explicit safety filters, posing a risk that models could generate and inadequately penalize harmful or toxic content. We therefore caution that the leaderboards reflect an emergent, inter-model consensus and should not be interpreted as a definitive or holistic measure of a model's quality, safety, or alignment. AUTOBENCH is intended as a research tool to complement, not replace, other evaluation methods, particularly direct human oversight, and should not be the sole basis for high-stakes decisions like model deployment.

Acknowledgements

This research is derived from the original open-source AutoBench project (eZecute S.R.L., 2024), an eZecute S.R.L. initiative with the substantial financial and expertize support by Translated S.R.L.. The authors would like to acknowledge the foundational work of the original AutoBench team. This paper was focused on bringing strong scientific validation to the project's core methodology.

References

AutoBench. 2024. Autobench 1.0: A collective-llm-as-a-judge benchmark system. https://huggingface.co/AutoBench. Accessed: 2025-10-21.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *Preprint*, arXiv:2308.07201.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, and 1 others. 2024. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*.

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

eZecute S.R.L. 2024. Autobench: Delivering transparency in llm benchmarking. https://autobench.org/. Accessed: 2025-10-21.

Evan Frick, Connor Chen, Joseph Tennyson, Tianle Li, Wei-Lin Chiang, Anastasios N. Angelopoulos, and Ion Stoica. 2025. Prompt-to-leaderboard. *Preprint*, arXiv:2502.14855.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

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. *CoRR*, abs/2103.03874.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *ArXiv*, abs/1904.09751.

M. G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in nlp. *Preprint*, arXiv:2104.14337.

Peter Kruger. 2025a. Autobench third run: Revolutionizing llm evaluation with record-breaking scale, accuracy, and a new home at autobench.org. `https://huggingface.co/blog/PeterKruger/autobench-run-3`. Accessed: 2025-10-21.

Peter Kruger. 2025b. Escape the benchmark trap: Autobench – the collective-llm-as-a-judge system for evaluating ai models (asi-ready!). `https://huggingface.co/blog/PeterKruger/autobench`. Accessed: 2025-10-21.

Daniel Li and Lincoln Murr. 2024. Humaneval on latest gpt models–2024. *arXiv preprint arXiv:2402.14852*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *Preprint*, arXiv:2406.11939.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

C. Spearman. 2015. The proof and measurement of association between two things. *International journal of epidemiology*, 39 5:1137–50.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. Livebench: A challenging, contamination-limited llm benchmark. *Preprint*, arXiv:2406.19314.

Ruochen Zhao, Wenxuan Zhang, Yew Ken Chia, Weiwen Xu, Deli Zhao, and Lidong Bing. 2024. Auto-arena: Automating llm evaluations with agent peer battles and committee discussions. *Preprint*, arXiv:2405.20267.

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. *Preprint*, arXiv:2306.05685.

A Appendix

Prompts and Sampling Mechanisms This section of the appendix details the core prompts and sampling mechanisms that drive the AUTOBENCH framework. First, a `difficulty` level is sampled, with the generated token being `'a very difficult'`, `'a difficult'`, or `'a'` (standard difficulty) with probabilities of 0.6, 0.3, and 0.1, respectively, to favor more challenging problems. A `topic` is then sampled uniformly from the set: {math, current news, creative writing, logic, grammar, coding, history, general culture, science, technology}. After sampling a `topic` and `difficulty`, the framework constructs a sequence of task-specific prompts.

The task prompt (Table 5) instructs a model to formulate a clear, self-contained task. Once generated, all models act as contestants, producing answers following the guidelines in Table 6. Subsequently, models evaluate both task and answer quality. The task evaluation prompt (Table 7) scores clarity, relevance, and appropriateness for the specified difficulty, while the answer evaluation prompt (Table 8) applies a strict correctness gate and ranks accuracy, clarity, and depth. Together, these structured prompts ensure coherent task generation, consistent answer formulation, and reliable model assessment throughout the AUTOBENCH pipeline.

Pseudocode To ensure clarity and facilitate the reproducibility of our framework, we formalize the complete iterative procedure of AUTOBENCH in Algorithm 1. The algorithm details each phase of an iteration, from the initial task generation and its quality assurance check against score thresholds, to the subsequent peer evaluation where all models judge all answers. This pseudocode provides a complete blueprint of the methodology presented in this paper.

Model Suite Details To provide context for our experiments, we present the full list of language models evaluated in this study in Table 2. The selection was curated to include a diverse suite of contemporary open-source models, varying signifi-

cantly in parameter count, architecture, and developer. This diversity is intended to ensure a comprehensive and robust assessment of the AUTOBENCH framework's ability to differentiate between a wide range of model capabilities.

Supplementary Results We provide some supplementary results to complement the analysis in the main paper. Figure 3 offers a granular view of the inter-model evaluation dynamics through a heatmap of the aggregate judgment matrix. We present the full aggregate leaderboard in Table 3, showing the final AUTOBENCH scores alongside their corresponding performance on MMLU-Pro and GPQA. Finally, Table 4 offers a fine-grained breakdown of model performance across different task categories, highlighting specific areas of strength and weakness.

LLM Usage Disclosure In accordance with ethical guidelines, the authors disclose the use of a large language model (`Gemini 2.5 Pro`) during the preparation of this paper. The model's application was confined to an advisory role for improving the clarity of the manuscript. The conceptual framework, experimental results, and all scientific claims are the exclusive contributions of the authors, who retain full responsibility for the work.

Computational Details The experiments were conducted on a single machine equipped with four NVIDIA A6000 GPUs. All models were hosted locally and served using the vLLM inference engine for optimized throughput. A full run of the AUTOBENCH framework, comprising 40 iterations, required approximately 30 hours of computation time. Due to our limited memory budget, models were loaded into and unloaded from GPU memory between evaluation steps, with this process accounting for a significant portion of the total runtime.

Algorithm 1 Autobench Evaluation Algorithm

Require: Models $\mathcal{M} = \{M_1, \ldots, M_n\}$, iterations T, thresholds $\lambda_{\text{mean}}, \lambda_{\text{median}}$, max retries k.
Ensure: Final scores $\mathbf{c}^{(T)}$ and weights $\mathbf{w}^{(T+1)}$.

Initialize: $\mathbf{c}^{(0)} \leftarrow \mathbf{0}$; $\mathbf{w}^{(1)} \leftarrow (1/n, \ldots, 1/n)$
for $t = 1, \ldots, T$ **do**
  $retries \leftarrow 0$; $task_accepted \leftarrow$ false
  repeat
    $M_q \leftarrow sample(\mathcal{M})$
    $q^{(t)} \leftarrow generate(M_q)$
    $\mathbf{s} \leftarrow \big(score(M_1, q^{(t)}), \ldots, score(M_n, q^{(t)})\big)$
    $\phi_1 \leftarrow w_mean(\mathbf{s}, \mathbf{w}^{(t)}) \geq \lambda_{\text{mean}}$
    $\phi_2 \leftarrow w_median(\mathbf{s}, \mathbf{w}^{(t)}) \geq \lambda_{\text{median}}$
    if $\phi_1 \wedge \phi_2$ **then**
      $task_accepted \leftarrow$ true
    else
      $retries \leftarrow retries + 1$
    end if
  until $task_accepted$ **or** $retries \geq k$
  if $task_accepted$ **then**
    $\forall i, j \in \{1, \ldots, n\},\ J_{i,j}^{(t)} \leftarrow \texttt{score}\big(M_i, \texttt{answer}(M_j, q^{(t)})\big)$
    $\mathbf{r}^{(t)} \leftarrow (J^{(t)})^\top \mathbf{w}^{(t)}$
    $\mathbf{c}^{(t)} \leftarrow \dfrac{(t-1)\mathbf{c}^{(t-1)} + \mathbf{r}^{(t)}}{t}$
    $\mathbf{w}^{(t+1)} \leftarrow \mathbf{c}^{(t)} / \sum_{i=1}^{n} c_i^{(t)}$
  else
    $\mathbf{c}^{(t)} \leftarrow \mathbf{c}^{(t-1)}$; $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)}$
  end if
end for
return $\mathbf{c}^{(T)}, \mathbf{w}^{(T+1)}$

Table 2: Overview of the models evaluated on the AutoBench benchmark.

Model	Parameters	Reasoning	Release	Access/License
SmolLM2-1.7B-Instruct	1.71 B	✗	2025	apache-2.0
Llama-3.2-1B-Instruct	1.24 B	✗	2024	llama3.2
Llama-3.2-3B-Instruct	3.21 B	✗	2024	llama3.2
Qwen2.5-3B-Instruct	3.09 B	✗	2024	qwen
Qwen2.5-7B-Instruct	8.29 B	✗	2025	apache-2.0
Qwen2.5-14B-Instruct	14.80 B	✗	2024	apache-2.0
Meta-Llama-3-8B-Instruct	8.03 B	✗	2024	llama3.1
gemma-7b-it	8.54 B	✗	2024	gemma
Mistral-Small-3.1-24B-Instruct-2503	24.0 B	✗	2025	apache-2.0
gpt-oss-20b	21.50 B	✓	2025	apache-2.0
gemma-3-1b-it	1.00 B	✗	2025	gemma
Phi-3-mini-4k-instruct	3.82 B	✗	2024	MIT

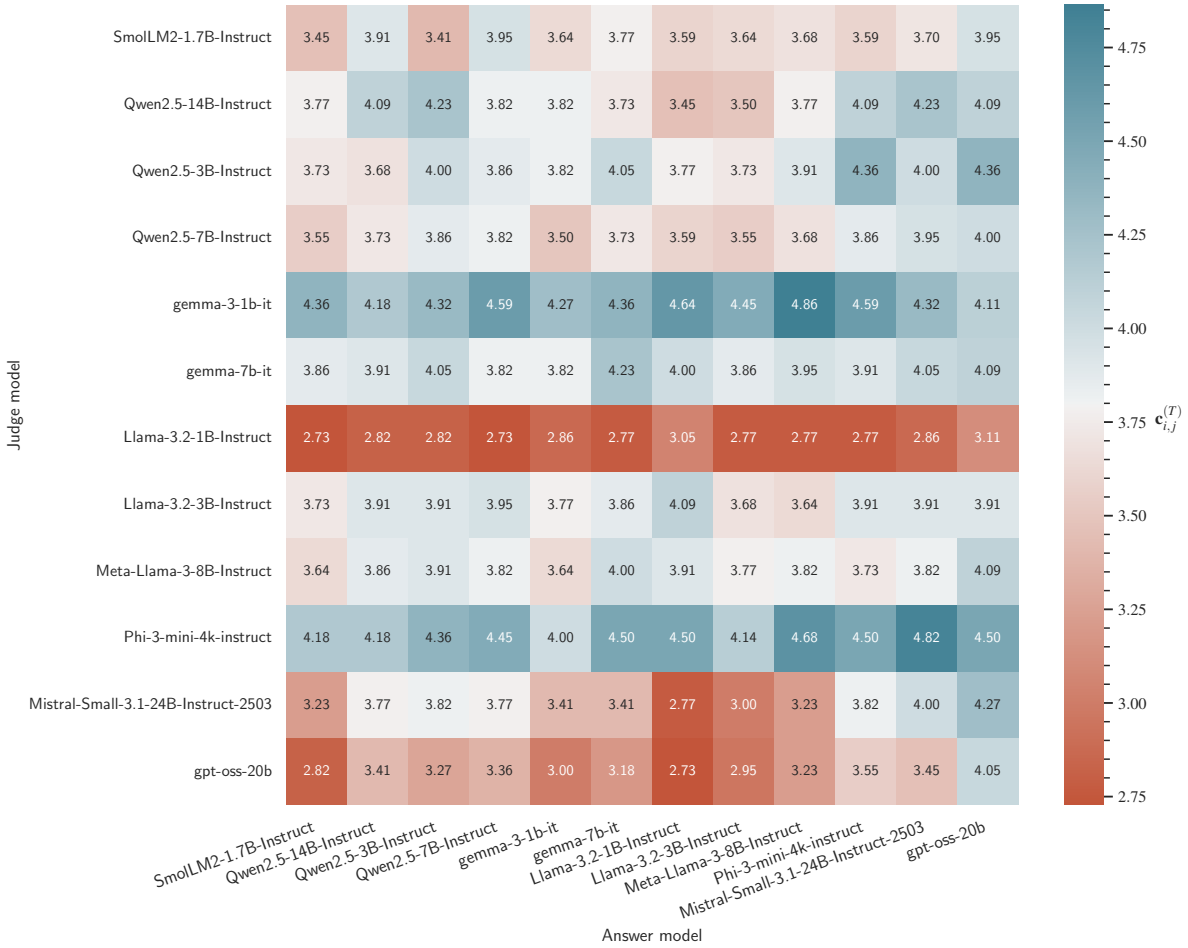


Figure 3: Heatmap of the aggregate judgment matrix, averaged over all T iterations, each cell (r, c) shows the mean score assigned by judge r to the contestant model c. The main diagonal $(r = c)$ reveals potential self-preference, as it shows the score each model gives to itself.

Table 3: Aggregate model performance across benchmarks. We report the primary score for each benchmark.

Model	**AutoBench (Ours)**	**MMLU-Pro**	**GPQA**
`SmolLM2-1.7B-Instruct`	3.80	11.71%	3.91%
`Llama-3.2-1B-Instruct`	3.58	7.58%	3.36%
`Llama-3.2-3B-Instruct`	3.73	24.39%	3.80%
`Qwen2.5-3B-Instruct`	3.77	25.05%	3.02%
`Qwen2.5-7B-Instruct`	3.80	36.52%	5.48%
`Qwen2.5-14B-Instruct`	3.91	43.38%	9.62%
`Meta-Llama-3-8B-Instruct`	3.85	29.60%	1.23%
`gemma-7b-it`	3.85	7.72%	4.59%
`Mistral-Small-3.1-24B-Instruct-2503`	3.95	66.76%	44.42%
`gpt-oss-20b`	4.17	73.14%	71.50%
`gemma-3-1b-it`	3.77	14.70%	19.20%
`Phi-3-mini-4k-instruct`	3.91	33.58%	10.96%

Table 4: Detailed score overview of model performance on the different categories within the AutoBench benchmark. This breakdown highlights specific strengths and weaknesses of each model.

Model	General Culture	Grammar	History	Math	Science	Technology
SmolLM2-1.7B-Instruct	4.06	3.75	3.87	3.02	3.52	3.75
Llama-3.2-1B-Instruct	4.08	3.57	3.60	3.79	3.28	3.70
Llama-3.2-3B-Instruct	4.06	2.32	3.84	3.47	3.12	4.00
Qwen2.5-3B-Instruct	4.08	3.74	3.83	3.80	3.71	3.83
Qwen2.5-7B-Instruct	4.13	3.42	3.91	3.71	3.70	3.91
Qwen2.5-14B-Instruct	3.95	3.75	3.90	3.44	3.98	3.87
Meta-Llama-3-8B-Instruct	4.02	3.83	3.91	3.58	3.66	3.70
gemma-7b-it	4.11	3.67	3.81	4.03	3.12	4.00
Mistral-Small-3.1-24B-Instruct-2503	4.00	4.00	3.96	3.76	4.06	3.92
gpt-oss-20b	4.45	3.93	4.08	3.70	4.14	4.29
gemma-3-1b-it	4.08	1.82	3.86	3.75	3.11	3.79
Phi-3-mini-4k-instruct	4.11	2.32	3.94	3.96	3.83	4.09

Table 5: Prompt for Task Generation.

Task Generation

```
Generate {difficulty} question about {topic}. Keep the question clear,
self-contained, and unambiguous.

Requires analysis and multi-step reasoning; expect intermediate
difficulty.

Pose a math problem (word problem or symbolic) that requires
calculation or reasoning. Include units if relevant.

Return exactly the final question text only - do NOT include
chain-of-thought, commentary, steps, or explanations.

Do not surround the question with quotes or extraneous punctuation.
Keep it concise and well-formed.

If you are a thinking/reasoning model, ALWAYS ensure the final
question appears in the standard 'content' field only; do NOT
place it in 'reasoning_content', do NOT leave the 'content'
field empty once you finish reasoning.
```

Table 6: Prompt for Answer Generation.

Answer Generation

```
Answer the question clearly, correctly, and concisely. Avoid
hallucinations and speculative claims; include only facts you
are confident about. Do NOT reveal chain-of-thought, internal
reasoning steps, or chain-of-thought tags; output only the final
answer. If you are a thinking/reasoning model, ALWAYS place the
final answer in the standard 'content' field only; do NOT place
it in 'reasoning_content', do NOT leave the 'content' field
empty once you finish reasoning.

Question: [Question text is inserted here]

Answer:
```

Table 7: Prompt for Ranking a Task's Quality.

Task Rating

```
Evaluate the quality of the question on an integer scale 1-5.
Consider clarity, appropriateness for the stated difficulty,
relevance to the topic, grammar/formulation, and engagement.

Return ONLY the final score as XML exactly like this: <rank>n</rank>
with no extra text. If you are a thinking/reasoning model, place
the final <rank> tag in the 'content' field only; do not place
it in 'reasoning_content'.

Question: [Question text is inserted here]
Difficulty: [Difficulty label, e.g., 'difficult']
Topic: [Topic label, e.g., 'math']

Return:
```

Table 8: Prompt for Ranking an Answer's Quality.

Answer Rating

```
You are an expert evaluator. Rank the answer on an integer scale
1–5 using the criteria below. **Strict correctness gate:** if the
answer contains any factual error, clear hallucination, or an
answer that is broken/irrelevant for the question, the maximum
score allowed is 2. After applying the gate, judge on
correctness, clarity, relevance, structure/grammar, and depth.

Scale (post-gate):
1: Poor/off-topic or contains substantial errors.
2: Partially correct but contains important mistakes or omissions.
3: Correct, clear, and relevant; adequate but not deep.
4: Correct with good structure, depth, and precise coverage.
5: Correct and exceptional — excellent clarity, original insight,
   or thoroughness.

IMPORTANT: Return ONLY the rank wrapped in XML exactly like this:
<rank>n</rank> where n is an integer 1-5. No additional text,
no punctuation, no explanation. If you are a thinking/reasoning
model, place the final <rank> tag in the 'content' field only;
do not place it in 'reasoning_content'.

Question: [Question text is inserted here]
Answer: [Answer text is inserted here]

Return:
```