ChinaRxiv

English-access research papers from ChinaXiv

Statistical Reliability and Efficiency Bias in Machine Learning Benchmark Rankings

2026-07-24

ChinaRxiv

View the original and related papers at
https://chinarxiv.org/items/chinaxiv-202607.00273

Source: ChinaXiv. Machine translation. Verify with the original.

Abstract

Machine learning benchmark leaderboards are core infrastructure driving model selection, yet the statistical reliability of rankings themselves has long lacked systematic auditing. This paper aims to quantify the statistical uncertainty of leaderboard rankings and evaluate the systematic impact of the absence of efficiency metrics on ranking structures. A cross-domain meta-analysis framework is proposed: using public Elo rating data from 371 models across three LMSys Chatbot Arena domains (Text/Vision/Image), adjacent-rank z-tests, Bootstrap resampling (10000 times/model), ranking calibration error (RCE), and Cohen' s d effect size analysis are applied. Meanwhile, using BigCodeBench as the target, a synthetic Transformer efficiency benchmark experiment (4 parameter-scale groups) and leaderboard cross-analysis (12 models) are designed to quantify ranking bias caused by missing efficiency information. In the Text domain, only 4.2% of 265 adjacent ranking pairs are statistically significant ($p<0.05$). Bootstrap ranking stability ranges from 0.11 to 0.70, with larger leaderboards exhibiting more unstable rankings. RCE values range from 0.004 to 0.053, cross-domain validation confirming the approximate relationship RCE $(1-|\rho|)/2$. Efficiency experiments show that a 4-fold increase in parameter count leads to a 13.1-fold increase in latency, and models with the same score (52% pass@1) differ by 4.3-fold in inference latency; after introducing the efficiency dimension, rankings can drift by $\pm$4 positions. The analysis covers only Arena-type leaderboards and does not address fixed-test-set benchmarks. Bootstrap assumes symmetric normal CIs. Efficiency experiments are based on CPU synthetic benchmarks and linear extrapolation and have not been validated in real GPU environments. Current single-dimensional leaderboard rankings are methodologically insufficient to support model selection decisions. It is recommended that the leaderboard community introduce ranking confidence interval reporting, adopt RCE as a quality measurement standard, and provide accuracy-efficiency dual-dimensional sorting options in ranking interfaces.

Full Text

Statistical Reliability and Efficiency Bias in Machine Learning Benchmark Rankings

HAN Yuxuan*

School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou, Jiangsu 213159

Corresponding author: HAN Yuxuan*, E-mail: 2300770102@smail.cczu.edu.cn

Abstract:

[Objective] Machine learning benchmark leaderboards are core infrastructure driving model selection, yet the statistical reliability of the rankings themselves

has long lacked systematic auditing. This paper aims to quantify the statistical uncertainty of leaderboard rankings and evaluate the systematic impact of the absence of efficiency metrics on ranking structures.

[Methods] A cross-domain meta-analysis framework is proposed: adjacent-rank z-tests, Bootstrap resampling (10,000 iterations per model), Ranking Calibration Error (RCE), and Cohen' s d effect-size analysis are applied to public Elo rating data for 371 models across three LMSys Chatbot Arena domains (Text/Vision/Image). Meanwhile, using BigCodeBench as the object, a synthetic Transformer efficiency benchmark experiment (4 parameter-scale groups) and leaderboard cross-analysis (12 models) are designed to quantify ranking bias caused by missing efficiency information.

[Results] In the Text domain, only 4.2% of 265 adjacent-ranking pairs are statistically significant ($p < 0.05$); Bootstrap ranking stability is 0.11–0.70, and rankings on larger leaderboards are more unstable; RCE values are 0.004–0.053, and cross-domain analysis verifies the approximate relationship RCE $\approx (1 - |\rho|)/2$. The efficiency experiment finds that a 4-fold increase in parameter count leads to a 13.1-fold increase in latency; models with the same score (52% pass@1) differ in inference latency by 4.3-fold, and after introducing the efficiency dimension, rankings may drift by ± 4 positions.

[Limitations] The analysis covers only Arena-style leaderboards and does not involve fixed-test-set benchmarks; Bootstrap assumes symmetric normal CIs; the efficiency experiment is based on a CPU synthetic benchmark and linear extrapolation and has not been validated in a real GPU environment.

[Conclusion] The current single-dimensional rankings of leaderboards are methodologically insufficient to support model selection decisions. It is recommended that the leaderboard community introduce reporting of ranking confidence intervals, adopt RCE as a quality metric, and provide accuracy-efficiency two-dimensional sorting options in ranking interfaces.

Keywords: benchmark leaderboard; statistical reliability; ranking calibration; meta-analysis; machine learning evaluation

Classification number: TP391

Statistical Reliability and Efficiency Bias in Machine Learning Benchmark Rankings

HAN Yuxuan*

School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213159, China

Corresponding author*: HAN Yuxuan, E-mail: 2300770102@smail.cczu.edu.cn

Abstract:

[Objective] Machine learning benchmark leaderboards drive model selection and research prioritization, yet the statistical reliability of their rankings has rarely been systematically audited. This paper quantifies the statistical uncertainty of leaderboard rankings and evaluates the systematic impact of missing efficiency metrics on ranking structures.

[Methods] A cross-domain meta-analysis framework is proposed. Adjacent-rank z-tests, Bootstrap resampling (10,000 iterations per model), Ranking Calibration Error (RCE), and Cohen' s d are applied to public Elo ratings of 371 models across three LMSys Chatbot Arena domains (Text/Vision/Image). Additionally, a synthetic Transformer

efficiency benchmark (four parameter scales) and a leaderboard cross-analysis (12 models) are conducted on BigCodeBench to quantify ranking bias caused by missing efficiency dimensions.

[Results] Only 4.2% of 265 adjacent rank pairs in the Text domain are statistically significant ($p < 0.05$). Bootstrap rank stability ranges from 0.11 to 0.70, with larger leaderboards exhibiting lower stability. RCE values range from 0.004 to 0.053, validating the cross-domain applicability of the RCE $\approx (1 - |\rho|)/2$ relationship. Efficiency experiments reveal that a 4× increase in parameters leads to a 13.1× increase in inference latency; models with identical accuracy (52% pass@1) differ by a factor of 4.3 in inference latency; incorporating efficiency can shift rankings by ± 4 positions.

[Limitations] The analysis covers only Arena-style leaderboards, not fixed-test-set benchmarks. Bootstrap assumes symmetric normal CIs, which may not fully hold for Bradley–Terry model estimates. Efficiency experiments are based on CPU synthetic benchmarks and linear extrapolation, pending validation on real GPU hardware.

[Conclusions] Single-dimension leaderboard rankings are methodologically insufficient to support model selection decisions. We recommend that the leaderboard community adopt ranking confidence interval reporting, treat RCE as a quality standard, and provide accuracy–efficiency dual-dimension sorting options.

Keywords: Benchmark leaderboard; Statistical reliability; Rank calibration; Meta-analysis; Machine learning evaluation

1 Introduction

1.1 Research Background and Problem

Progress in machine learning has been driven to a large extent by standardized benchmark leaderboards. From the Open LLM Leaderboard in NLG, to BigCodeBench [5] for code generation, and then to T2I benchmarks for image generation, these leaderboards directly shape the flow of research resources, model

selection in industry, and even policymaking. Users make decisions according to rank order: the model ranked third is regarded as better than the one ranked fifth, while the model ranked tenth is simply ignored.

A fundamental question has long been overlooked: **how much of these differences in ranking are real, and how much is merely noise?**

In medical clinical trials or social-science surveys, any ranking or comparison is required to be accompanied by confidence intervals and significance tests. Yet the rankings on machine-learning leaderboards are almost entirely based on point estimates—an Elo mean, a pass@1 percentage—and lack any report of sampling variability. For two models whose scores are sufficiently close, their order on the leaderboard may be purely random.

More troubling still, leaderboards look only at quality of effectiveness and not at computational efficiency. Inference latency, GPU memory, throughput—metrics equally critical in real deployment—are provided by no mainstream leaderboard. A row of models is ordered from high to low by accuracy, but you do not know which one costs ten times more to run. To clarify the magnitude of this bias, in addition to conducting a statistical audit of three Arena-style leaderboards, this paper also performs a set of supplementary efficiency experiments on BigCodeBench. These two lines of evidence corroborate each other and point to the same conclusion: the current leaderboard ecosystem suffers from structural defects.

1.2 Related Work

This work lies at the intersection of three research directions.

Ranking calibration and statistical auditing. Huang et al. [1] proposed the Rank-Calibration framework at NAACL 2024, for the first time formalizing the quantification of uncertainty in NLG evaluation as ranking calibration error (RCE), and demonstrating that the randomness of rankings in LLM evaluation far exceeds previous understanding. von Luxburg et al. [9] systematically criticized ML benchmark rankings from a statistical perspective, pointing out that small ranking differences on leaderboards lack statistical guarantees. The methodology of this paper directly inherits from these two lines of work, extending the RCE framework from NLG to cross-domain scenarios.

Analysis of LLM inference efficiency. Yuan et al. [6], based on the Roofline model [7], carried out a systematic bottleneck decomposition of LLM inference, revealing that the Decode stage is always memory-bottlenecked, while the Prefill stage shifts from a compute bottleneck to a memory bottleneck as sequence length increases. MLPerf [10] established a standardized evaluation paradigm for inference efficiency. However, these works focus on efficiency profiling of individual models and do not connect the efficiency dimension with the leaderboard ranking system—precisely the gap that Module 2 of this paper attempts to fill.

Figure 1: Research framework

Figure 1: Figure 1: Research framework

Benchmark auditing and meta-analysis. Zheng et al. [2] and Chiang et al. [3] constructed Chatbot Arena, an LLM evaluation platform based on human preferences; its publicly available Elo scores and 95% CI data form the data foundation for Module 1 of this paper. Chen et al. [4] and Zhuo et al. [5] established the pass@k paradigm for code-generation benchmarks. However, the common feature of these benchmarks is "single-dimensional scoring" —they do not examine whether their own rankings are statistically robust. In essence, this work is a meta-audit of these benchmarks, asking whether "the leaderboard itself is trustworthy."

Figure contents

- **Module 1: Ranking statistical stability**
  - Bootstrap resampling
  - Adjacent-rank Z test
  - RCE ranking calibration
  - Cohen' s d effect size
  - Data: LMSys Arena public leaderboard CSV
- **Module 2: Bias from missing efficiency indicators**
  - Synthetic Transformer baseline experiment
  - Function–efficiency cross-comparison analysis
  - Four-level efficiency rating
  - Data: BigCodeBench leaderboard + synthetic benchmark
- **Module 3: Cross-domain generalization analysis**
  - NLG Text (266 models)
  - Multimodal Vision (86 models)
  - Image generation Image (19 models)
  - Code generation BigCodeBench
  - Data: public data from 4 leaderboards
- **Core output: Recommendations for a joint accuracy–efficiency ranking system**
  - Only 4.2% of adjacent rankings are statistically significant
  - After introducing efficiency, rankings may drift by $±$4 positions
  - For the same model (52% pass@1), inference latency differs by a factor of 4.3

Figure 1 Research framework

1.3 Our Work

The work of this paper is divided into three modules: (1) **ranking statistical stability**—using Bootstrap and permutation tests to examine whether the rankings themselves are reliable; (2) **bias from missing efficiency indicators**—

using synthetic baselines and leaderboard cross-analysis to quantify what consequences arise when efficiency information is omitted from leaderboards; (3) **cross-domain comparison**–examining how the statistical characteristics of leaderboards differ across four different domains.

2 Research Design and Methods

2.1 Data Sources

This paper uses data from three public leaderboards (Table 1):

Table 1 Sources of public leaderboard data used in this paper

Leaderboard	Domain	Number of models	Scoring metric	Data source
LMSys Chatbot-Arena (Text)	Natural language processing	266	Elo score +95%CI	lmarena_text.csv
LMSys Chatbot-Arena (Vision)	Multimodal vision	86	Elo score +95%CI	lmarena_vision.csv
LMSys Chatbot-Arena (Image)	Image generation	19	Elo score +95%CI	lmarena_image.csv

Each record contains: ranking, Elo score (`arena_score`), 95% confidence interval (`95_pct_ci`, in the format "+m/−n"), number of votes (`votes`), organization, and license type.

2.2 Significance Test for Adjacent Rankings

For adjacent model pairs after sorting (M_i, M_{i+1}), the standard error is back-calculated from the CI:

$$SE_i = \text{CI\_upper}_i / 1.96$$

A two-sample z test is performed:

$$z = (s_i - s_{i+1}) / \sqrt{SE_i^2 + SE_{i+1}^2}$$

s_i is the Elo score of model i; a two-sided $p < 0.05$ is judged significant.

2.3 Bootstrap Ranking Confidence Intervals

For each model, 10,000 parametric Bootstrap samples are performed:

$$s_i^{(b)} = s_i + \varepsilon \cdot SE_i, \qquad \varepsilon \sim N(0, I), \qquad b = 1, \ldots, 10000$$

In each Bootstrap run, the scores are resampled and the models are re-ranked, yielding an empirical distribution of ranks for each model. Two indicators are defined:

- **Rank stability:** Stability $= 1/(1+\Delta\bar{R})$, where $\Delta\bar{R}$ is the mean Bootstrap 95% CI rank width across all models. The closer it is to 1, the more stable the ranking.
- **Rank turnover rate:** the proportion of models whose Bootstrap rank width is greater than 3 positions.

2.4 Ranking Calibration Error (RCE)

We adopt the ranking calibration framework of Huang et al. [11]. RCE is defined as the normalized mean absolute deviation between the Bootstrap median rank and the original rank:

$$RCE = (1/N) \cdot \sum_i \left| \text{rank\_bootstrap}^{(i)} - \text{rank\_original}^{(i)} \right| / N$$

RCE and the Spearman rank correlation coefficient approximately satisfy: $RCE \approx (1 - |\rho|)/2$.

2.5 Effect Size Analysis

Cohen' s d is used to measure the score gap between the Top-N models and the remaining models:

$$d = (\bar{s}_{\text{top}} - \bar{s}_{\text{rest}}) / \sqrt{(s\bar{E}_{\text{top}})^2 + (s\bar{E}_{\text{rest}})^2}$$

$|d| < 0.5$ is considered a small effect, 0.5–1.2 a medium effect, 1.2–2.0 a large effect, and > 2.0 an extremely large effect.

2.6 Supplementary Experiments on the Efficiency Dimension (BigCodeBench)

To quantify the impact of the missing efficiency information in leaderboards, this paper conducts two independent experiments on BigCodeBench.

Experiment 1: Synthetic Transformer efficiency benchmark. A NumPy-based simulation framework for decoder-only Transformer forward propagation was built. Four parameter scales were configured: Tiny (~50M,

$d = 512$, $L = 8$), Small (~125M, $d = 768$, $L = 12$), Medium (~350M, $d = 1024$, $L = 24$), and Large (~500M, $d = 1024$, $L = 32$). Inference latency, throughput, and peak memory were measured on an Intel Core i7-1260P. The synthetic benchmark cannot replace real GPU inference, but it cleanly captures the first-order impact of parameter count on efficiency, without interference from implementation factors such as KV-cache strategies and batch scheduling.

Experiment 2: Leaderboard cross-analysis. Twelve representative models (0.35B–15.7B) were selected from the BigCodeBench leaderboard [5], and `pass@1_full` and `pass@1_hard` were recorded. Using the benchmark data from Experiment 1, latency and memory for each model in a unified environment were estimated by linear extrapolation according to parameter count. Four efficiency ratings were defined: A level (latency <1s, memory <4GB, edge-deployable), B level (latency <10s, memory <16GB, server-friendly), C level (latency <60s, memory <64GB, high-end server), and D level (exceeding the C-level thresholds). Finally, an "efficiency-weighted score" = pass@1 × log(throughput) was used to re-rank the models, in order to observe how the rankings change before and after incorporating efficiency information.

3 Results Analysis

3.1 Significance Test for Adjacent Rankings

Figure 2. Distribution of Elo scores in the three domains

Across the adjacent rankings in the three domains, most differences are not statistically significant (Table 2; the score distributions are shown in Figure 2). In the Text domain, among 265 adjacent pairs of models, only 11 pairs (4.2%) reach $p < 0.05$. In other words, two models that are next to each other on the leaderboard—for example, ranks 42 and 43—may be ordered ahead of or behind one another for reasons that have little to do with true performance differences and are merely due to sampling fluctuations. The Vision domain is even worse: among 85 pairs, only 5 are significant (5.9%). Image is an exception (77.8%), because each model has, on average, 57,000 votes, the CIs are extremely narrow, and there are only 19 models in total, so the scores are less likely to overlap.

Table 2. Results of statistical significance tests for differences between adjacent rankings

Domain	Number of adjacent ranking pairs	Significant ($p < 0.05$)	Proportion
Text (NLG)	265	11	4.2%
Vision (multimodal)	85	5	5.9%

Domain	Number of adjacent ranking pairs	Significant ($p < 0.05$)	Proportion
Image (image generation)	18	14	77.8%

3.2 Bootstrap Ranking Uncertainty

Figure 3. Bootstrap ranking stability –Text domain (Top-30)

The Bootstrap results (Figure 3) reveal a counterintuitive phenomenon: the larger the leaderboard, the less stable the rankings. The ranking stability for Text (266 models) is only 0.111, whereas Image (19 models) reaches 0.704. The reason is not complicated—the more models there are, the more crowded the scores become, and the Bootstrap sampling range of any given model is more likely to overlap with its neighbors.

Table 3. Comparison of Bootstrap ranking-uncertainty indicators across three domains

Domain	Number of models	Stability	Top-10 average CI width (positions)	Flip rate (>3 positions)
Text (NLG)	266	0.111	3.4	91.4%
Vision (multi-modal)	86	0.141	3.5	86.0%
Image (image generation)	19	0.704	0.6	0.0%

The flip-rate column is particularly striking: in the Text domain, 91.4% of models have a Bootstrap ranking width exceeding 3 positions. For most models, the "true ranking"can drift within a fairly large range; the apparent determinacy of the leaderboard is an illusion.

3.3 Ranking Calibration Error

Figure 4. Comparison of cross-domain ranking stability and RCE

The RCE calculation results are: Text = 0.0040 (approximately $\rho = 0.992$), Vision = 0.0114 (approximately $\rho = 0.977$), and Image = 0.0526 (approximately

Figure 4 chart: comparison of ranking stability and RCE across domains. Legend: Ranking stability; RCE. Y-axis: indicator value. Text: stability 0.111, RCE 0.004. Vision: stability 0.141, RCE 0.011. Image: stability 0.704, RCE 0.053.

Figure 2: Figure 4 chart: comparison of ranking stability and RCE across domains. Legend: Ranking stability; RCE. Y-axis: indicator value. Text: stability 0.111, RCE 0.004. Vision: stability 0.141, RCE 0.011. Image: stability 0.704, RCE 0.053.

$\rho = 0.895$). Figure 4 compares the stability and RCE of the three domains side by side. These numbers are consistent with the estimates of Huang et al. [1] in the NLG domain (RCE $\approx$ 0.05–0.15), indicating that the approximate relationship RCE $\approx (1 - |\rho|)/2$ also holds across domains.

The absolute values of RCE are not large—the maximum is 0.0526, meaning that ranking variation accounts for only 5.3% of the total ranking. However, the distribution of this 5.3% is highly uneven: the rankings of the Top-5 models are much more stable, because the score gaps among the leading models are themselves large (Cohen' s d all falls within the "very large effect" range, 22–50).

3.4 Confidence Intervals and Sample Size

Figure 5. Relationship between CI width and number of votes (three domains)
Scatter plot legend: Text (NLG); Vision (multimodal); Image (image generation).
x-axis: Number of votes (Votes). y-axis: 95% CI width.

There is a stable negative correlation between the number of votes and CI width (Text: $r = -0.644$, Vision: $r = -0.628$, Image: $r = -0.573$, all $p < 0.001$; see the scatter plot in Figure 5). The direction is consistent with SE $\propto 1/\sqrt{n}$, but the slope is much flatter than theoretical expectations. This indicates that sample size is not the only factor affecting the CI—the randomness of the evaluation prompts and within-judge bias, i.e., these non-sampling errors, account for a nontrivial share of the total error.

3.5 Open Models vs. Closed-Source Models

The scores of closed-source models are systematically higher than those of open-source models, with $p < 0.01$ in all three domains. The gap is 111.6 points for Text ($t = -8.29$), 116.7 points for Vision ($t = -5.36$), and 108.7 points for Image ($t = -5.37$). Whether this is because closed-source models are genuinely stronger, or because human judges have a cognitive preference for closed-source models, cannot be distinguished by the current experimental design. But whatever the cause, the implication for rankings is the same: a leaderboard is not a

purely neutral arena of competition.

3.6 The Absence of Efficiency Metrics: Empirical Quantification Based on BigCodeBench

Figure 6. Inference latency and throughput of code models at different parameter scales
Plot legend: linear-growth expectation (based on the 50M baseline); measured inference latency (s); throughput (tok/s).
x-axis: parameter scale—Tiny (~50M), Small (~125M), Medium (~350M), Large (~500M).
left y-axis: inference latency (s). right y-axis: throughput (tok/s). Visible annotations: 4.5×, 19.6×.

Nonlinearity of efficiency-scale. The results of the synthetic benchmark experiment (Table 4, Figure 6) are very direct: when the parameter count increases from 125M to 500M, it grows by only 4×, but inference latency surges from 1.80 seconds to 23.29 seconds—13.1×. Each segment shows increasing acceleration: 125M$\rightarrow 350M(parameters increase by 2.8\times, latency rises by 4.4\times), 350M\rightarrow$500M (parameters increase

1.4 times, while latency increases by 3.0 times). The leaderboard lists only the parameter count, but the relationship from parameter count to latency is nonlinear—users cannot see this.

Table 4. Synthetic Transformer efficiency benchmark results for code models of different parameter scales

Parameter scale	Parameter count	Inference latency (s)	Throughput (tok/s)	Peak memory (MB)
Tiny (~50M)	50M	0.40	1286	33.4
Small (~125M)	125M	1.80	1018	34.2
Medium (~350M)	350M	7.84	882	37.0
Large (~500M)	500M	23.29	792	38.5

Figure 7. BigCodeBench leaderboard pass@1 vs. inference efficiency

Visible plot labels: the horizontal axis is "functional correctness pass@1 (%)" ; the vertical axis is "estimated inference latency (s, logarithmic scale)."The legend marks the "upper limit for real-time interaction (60s)" and the "edge real-time threshold (1s)." The annotation reads: "Both are 52% pass@1; latency differs by

4.3×." Labeled models include CodeGen-350M, CodeGen-2B-Mono, DeepSeek-1.3B, StarCoder2-3B, CodeLlama-7B, Qwen2.5-Coder-7B, CodeLlama-13B, and CodeLlama-15.7B.

Both are 52%, yet the difference is 4.3×. The most striking comparison in the cross-analysis of the leaderboard (the bubble plot in Figure 7 shows all 12 models) is that StarCoder2-3B and CodeLlama-13B both achieve 52% pass@1 [5], so they are tied on the leaderboard. But the latency of StarCoder2-3B is only 23% of CodeLlama-13B' s latency (140s vs. 606s), and memory is likewise different—12GB vs. 52GB. On a standard server with 16GB of memory, one can run, while the other cannot run at all. The leaderboard does not convey this information, because it has only one dimension.

Figure 8. Distribution of BigCodeBench model efficiency grades

Visible chart labels: the pie chart is titled "Efficiency-grade distribution (n=12)" and contains grades A (edge-deployable: latency <1s, memory <4GB), B (server-friendly: <10s, <16GB), C (high-end server: <60s, <64GB), and D (cloud only: exceeds grade C), with proportions 8.3%, 33.3%, 50.0%, and 8.3%, respectively. The bar chart is titled "Inference latency and efficiency grade of each model" ; its horizontal axis is "inference latency (s, logarithmic scale)," with reference lines for grade A: 1s, grade B: 10s, and grade C: 60s.

Ninety percent of models cannot run at the edge. The distribution of the four-level A/B/C/D rating (Figure 8) is very clear: among the 12 models, only CodeGen-350M reaches grade A (edge real-time deployment). Four are in grade B (server-friendly), and seven are in grades C/D. If you choose a model only by pass@1, there is a greater than 90% chance that you will choose something that cannot run on your target hardware.

Figure 9. Changes in ranking before and after introducing the efficiency dimension

After re-ranking with `pass@1 \times \log(\text{throughput})` (Figure 9), StarCoder2-3B rises from 5th to 2nd, while CodeLlama-13B drops from 3rd to 7th. DeepSeek-Coder-1.3B rises by 4 places. Efficiency is not merely "supplementary information" —once it is added, the relative ordering among models undergoes structural changes.

Table 4. Experimental results on the synthetic Transformer efficiency benchmark for code models of different parameter scales

Model	Parameters	pass@1	Original Rank	Efficiency-Weighted Rank	Rank Change
StarCoder2-3B	3B	52%	5	2	+3↑
CodeLlama-13B	13B	52%	3	7	−4↓

Model	Parameters	pass@1	Original Rank	Efficiency-Weighted Rank	Rank Change
DeepSeek-Coder-1.3B	1.3B	45%	8	4	+4↑
CodeGen-2B-Mono	2B	36%	10	11	−1↓

Putting these together: the absence of efficiency metrics is not just a matter of incomplete information, but a systematic ranking bias. Two models with roughly similar capabilities may differ by orders of magnitude in deployment feasibility—something that is completely invisible on the leaderboard. This conclusion is consistent with the "memory wall" problem in computing systems discussed in the Roofline-model literature [6][7]; this paper simply places it in the context of leaderboard rankings and quantifies it there.

4 Discussion and Conclusions

4.1 The "Statistical Dark Matter" of Leaderboards

This paper calls the statistical uncertainty in leaderboard rankings "statistical dark matter" —it is large in quantity, has a large impact, but is rarely seen or reported. The main findings can be summarized in four points:

1. **Adjacent rankings are mostly unreliable.** Choosing a model by looking at adjacent ranks is not much different from flipping a coin.
2. **The larger the scale, the less stable the ranking.** A leaderboard ranking 266 models is more uncertain than one ranking 19 models—contrary to intuition.
3. **RCE is a good metric.** A single number can capture differences in stability across leaderboards, and cross-domain validation confirms its relationship with Spearman ρ [1].
4. **The absence of efficiency distorts rankings.** Two models placed together by BigCodeBench may differ by several times in deployment difficulty. Once the efficiency dimension is added, rankings change—sometimes substantially.

4.2 Recommendations for the Leaderboard Community

Based on these results, there are four things that can be done immediately:

- **Report confidence intervals for rankings.** Provide a Bootstrap 95% CI for every rank position. The cost is extremely low, and the amount of information gained is enormous.
- **Use RCE as a quality threshold.** $RCE \leq 0.05$ can serve as the passing line for a ranking to be considered "basically reliable."
- **Incorporate efficiency.** At minimum, report FP16 inference throughput and GPU memory usage. Leaderboard interfaces should provide a two-dimensional "accuracy–efficiency" ranking, allowing users to choose their own trade-off weights.
- **Indicate sample size.** The number of votes or evaluations for each model should be made explicit. Models with very small sample sizes should be flagged, so that users do not mistake them for having been sufficiently validated.

4.3 Limitations

There are several things this paper did not do, or did not do well enough. (1) The analysis covers only Arena-type leaderboards (Elo scores plus human voting); it does not yet address fixed-test-set benchmarks such as GLUE and MMLU. (2) Bootstrap assumes that the CI is symmetric and normal, but the CIs in the original reports may come from the Bradley–Terry model [2][3], and therefore may not fully satisfy this assumption. (3) The synthetic benchmark used in the efficiency experiments ran matrix simulations on CPU rather than real GPU inference. The efficiency estimates used linear extrapolation based on parameter count, and did not yet incorporate real-scenario complexities such as KV-cache management [6] and memory-bandwidth saturation [7]. The next step is to complete the real efficiency data for the 12 models on a unified GPU platform.

4.4 Conclusion

This paper conducts a systematic audit of the statistical reliability of three mainstream ML benchmark leaderboards, and supplements BigCodeBench with independent experiments along the efficiency dimension. The main conclusions are as follows: the vast majority of adjacent ranking differences are not significant; there is a counterintuitive negative correlation between leaderboard scale and ranking stability; RCE can effectively measure ranking quality in cross-domain scenarios; and the general absence of efficiency metrics leads to systematic ranking bias—the deployment feasibility of models with the same score can differ by several fold. Taken together, these findings point to a common judgment: the current "one-dimensional ranking" used by leaderboards is methodologically insufficient to support model-selection decisions. A joint ranking framework that simultaneously reports accuracy and efficiency, together with statistical confidence intervals, is urgently needed as foundational infrastructure for the next stage.

Author Contribution Statement

Han Yuying: proposed the research question and the cross-domain meta-analysis framework; completed LMSys leaderboard data collection and the code for Bootstrap statistical-stability analysis, as well as all experimental execution; completed the design of the BigCodeBench synthetic Transformer efficiency-benchmark experiments and the cross-analysis of leaderboards; completed the generation of all figures and tables and the data analysis; wrote, revised, and finalized the full text.

References:

[1] Huang, X., et al. Rank-Calibration: Towards reliable uncertainty quantification for NLG evaluation[C]. Proceedings of NAACL 2024, 2024.

[2] Zheng, L., et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena[C]. Proceedings of NeurIPS 2023, 2023. DOI: 10.48550/arXiv.2306.05685.

[3] Chiang, W.-L., et al. Chatbot Arena: An open platform for evaluating LLMs by human preference[J]. arXiv preprint arXiv:2403.04132, 2024. Retrieved from https://arxiv.org/abs/2403.04132.

[4] Chen, M., et al. Evaluating large language models trained on code[J]. arXiv preprint arXiv:2107.03374, 2021. Retrieved from https://arxiv.org/abs/2107.03374.

[5] Zhuo, T. Y., et al. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions[C]. Proceedings of NeurIPS 2024, 2024. DOI: 10.48550/arXiv.2406.15877.

[6] Yuan, Z., et al. LLM inference unveiled: Survey and Roofline model insights[J]. arXiv preprint arXiv:2402.16363, 2024. Retrieved from https://arxiv.org/abs/2402.16363.

[7] Williams, S., Waterman, A., & Patterson, D. Roofline: An insightful visual performance model for multicore architectures[J]. Communications of the ACM, 2009, 52(4): 65–76. DOI: 10.1145/1498765.1498785.

[8] Hu, E. J., et al. LoRA: Low-rank adaptation of large language models[C]. Proceedings of ICLR 2022, 2022. Retrieved from https://arxiv.org/abs/2106.09685.

[9] von Luxburg, U., et al. Statistical inference for ML benchmark rankings[C]. ICML 2024 Workshop on Evaluating Evaluations, 2024.

[10] Reddi, V. J., et al. MLPerf inference benchmark[C]. Proceedings of ISCA 2020, 2020: 446–459. DOI: 10.1109/ISCA45697.2020.00045.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv. Machine translation. Verify with the original.