

How Every Number Is Produced: A Verifiable Methodology for Evaluating Large Language Models

Suhas Bhairav

Independent researcher - suhasbhairav.com

Preprint - October 30, 2026

Abstract

Published accuracy figures for large language models are rarely traceable to the requests that produced them. This paper describes how every number in an independent evaluation of 72 models is produced, so that each one can be traced to a recorded API request and response. Every model answers the same 80 ten-option multiple-choice questions, sampled from two open datasets (SuperGPQA and MMLU-Pro), through OpenRouter's public API with identical settings (temperature 0, reasoning effort "low", at most 4000 output tokens). Answers are graded by a deterministic rule with no human or model judge. The verification score is the share of graded questions answered correctly, reported with a 95% Wilson interval; one model is called more accurate than another only when a two-proportion z-test gives $|z| \geq 1.96$, and speed, latency and price differences count only past a 10% margin. Cost is what was actually billed. Every request and response is kept as a proof file whose SHA-256 goes into a run manifest. An independent re-check confirmed 5,760 of 5,760 proof hashes and 5,756 of 5,756 re-graded verdicts; 12 answers not confirmed by their manifest were excluded, leaving 5,746 graded answers.

Keywords: large language models; LLM evaluation; reproducibility; SuperGPQA; MMLU-Pro; Wilson interval; two-proportion z-test; McNemar test; benchmark methodology; SHA-256 evidence; audit trail

1 Introduction

Comparisons of large language models usually rest on numbers the reader cannot check: scores copied from model cards, leaderboards that do not publish raw outputs, or results produced with prompts and settings that are only partly described. When two sources disagree there is no way to find out why.

Nothing in this work is copied from a vendor. Every figure is computed from requests I sent myself, and every one of those requests is kept as a record. This paper describes how each number in the two companion result papers, on frontier and non-frontier models, is produced, and how a reader can check it.

2 Calling the models

The benchmark harness (version 0.1.0) is a single Python script that uses only the standard library. It calls every model through OpenRouter's public chat completions API (<https://openrouter.ai/api/v1/chat/completions>), the same endpoint any developer can use, with streaming switched on. The model IDs tested are the ones listed in OpenRouter's model catalogue, and every table in the result papers gives the exact ID.

Every model gets the same request settings. Quality questions are sent at temperature 0, so the model picks its most likely answer rather than sampling. Models that accept a reasoning setting are sent effort "low"; models that don't accept one get no reasoning parameter at all. Each quality answer may use up to 4000 output tokens, reasoning included, and each speed run is capped at 700 output tokens. A request that runs longer than the harness's time limit is stopped and recorded as a failure.

Table 1. Request settings for the quality questions (identical in every run that contributed a score).

Setting	Value
Temperature	0
Reasoning effort (where supported)	low
Maximum output tokens (reasoning included)	4000
Socket timeout per request	180 s
Attempts per request	up to 3
Wait before attempt 2 / attempt 3	4 s / 8 s

For every request the harness records the exact request body, the full streamed response (answer text and any visible reasoning), the time each part arrived, the token counts and which host served it. A moment after each request ends, it fetches OpenRouter's generation record, which holds the native token counts and the amount actually charged.

3 The question suite

Every model answers the same 80 questions from "Verification suite v2". The questions are not written by me. They are sampled, with fixed random seeds so the draw can be repeated, from open datasets that anyone can download:

Table 2. Source datasets.

Dataset	Questions	Subset	Licence
SuperGPQA	50	difficulty = hard, stratified by discipline	ODC-BY 1.0
MMLU-Pro	30	test split, stratified by subject	MIT

Every question is multiple choice with ten options, labelled A to J, and exactly one correct answer from the dataset's own answer key. With ten options, a model that guessed at random would score about 10%. The questions fall into 6 subject groups: science (30), engineering (15), medicine (11), economics (9), law (6) and humanities & social science (9). Each subject holds only a handful of questions, so subject results are shown as counts (for example 7/11), not as rankings. Each item keeps the dataset's own identifier (the SuperGPQA UUID or the MMLU-Pro question_id), so any question can be looked up in the public dataset.

How an answer is graded

Each prompt ends with the instruction to give the final answer on the last line in the form ANSWER: <letter>. The harness takes the text after the last "ANSWER:" in the reply (so a model can reason first and change its mind), or the last non-empty line if there is none. It then normalises that text: lower case, surrounding quotes, asterisks and backticks removed, spaces collapsed, trailing punctuation dropped. The answer counts as correct only if it matches the answer key's letter. There is no partial credit, no human judging and no second model acting as a grader, so anyone with the response text can re-check the verdict.

A sample question

This is one of the 80 questions exactly as it is asked, from MMLU-Pro (Humanities & social science). The expected answer is A.

Of the two versions of the principle that Singer considers:

- A. he considers the stronger version to be correct.
- B. he considers the weaker version to be correct.
- C. he proposes a third, completely different principle.
- D. he argues that the weaker version is stronger.
- E. he claims that they are equivalent.
- F. he supports both the stronger and the weaker versions.
- G. he suggests that the principles are contradictory.
- H. he does not distinguish between the two principles.
- I. he states that both principles are incorrect.
- J. he ends up rejecting them both.

Choose the single best option.

Give your final answer on the last line in the form: ANSWER: <letter>

Suite SHA-256: cd444ee142dc6b87e74967c9925ea9b7b171b90fe0e18b5a214c4587d964121a.

4 Verification score and its uncertainty

The verification score is simply the share of graded questions a model answered correctly. A sample of 80 questions is enough to separate strong models from weak ones, but not to rank models that are a few points apart. I show that uncertainty instead of hiding it.

The usual way to put a range on a proportion like this is the Wilson score interval [1]. With 80 questions, a model scoring 50% has a 95% interval of 39.3% to 60.7%, about +/-10.7 points. At the median score of the 72 models evaluated, 62.5%, the interval is 51.5% to 72.3%, about +/-10.4 points. In plain terms: if two models are within a few points of each other, the order between them could easily flip on a different set of questions.

When a model is called more accurate

One model is only said to be more accurate than another when a two-proportion z-test on the two scores, with a pooled standard error, gives $|z|$ of at least 1.96, the usual threshold for 95% confidence on a two-sided test. Below that, the gap is reported as too close to call. In the result papers every model is tested against the highest-scoring model in its group.

For output speed, time to first token and price a simpler rule applies: a model is only better if it is at least 10% better, and anything closer is reported as similar. Tables still sort by the raw numbers, but the written verdicts follow these rules.

A paired check

The z-test treats the two scores as independent samples. Because every model answers the same questions, the result papers also report a paired check: an exact McNemar test on the questions where only one of the two models was right, with a Holm-Bonferroni correction for the number of comparisons in the group. The headline verdicts follow the z-test; where the two tests disagree, the papers say so.

The statistics in plain language

Wilson score interval

A confidence interval for a proportion such as "68 of 80 correct". Unlike the simple plus-or-minus formula it stays inside 0-100% and behaves well for small samples and scores near the extremes. See: https://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval#Wilson_score_interval

Two-proportion z-test

Tests whether two success rates differ by more than chance would explain. The gap between the two scores is divided by its standard error (pooled from both models); $|z| \geq 1.96$ corresponds to 95% confidence, two-sided. See: https://en.wikipedia.org/wiki/Two-proportion_Z-test

McNemar's test

A test for paired yes/no outcomes. Because both models answered the same questions, it looks only at the questions where exactly one of them was right and asks whether that split could be a coin flip. It ignores questions both got right or both got wrong. See: https://en.wikipedia.org/wiki/McNemar%27s_test

Holm-Bonferroni correction

When one model is compared with many others, some comparisons look significant by luck alone. Holm's method raises the bar step by step so that the chance of any false "significant" result across the whole set stays at 5%. See: https://en.wikipedia.org/wiki/Holm%E2%80%93Bonferroni_method

Spearman's rank correlation

Measures how consistently one quantity rises with another using only their rank order, from -1 to +1. Here it asks whether more expensive models tend to score higher. See: https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

Pareto set (cost-accuracy frontier)

The models for which no other model is both at least as accurate and at least as cheap. Any model off the set is beaten on both counts by one on it. See: https://en.wikipedia.org/wiki/Pareto_front

5 Output speed and latency

Each model streams the same fixed prompt 3 times. The prompt asks for about 300 words of plain prose, which gives a long enough answer to measure a steady generation rate:

Explain how a household refrigerator keeps food cold. Cover the refrigerant cycle, the compressor, the condenser and the evaporator. Write about 300 words of plain prose without headings or lists.

Time to first token is measured from sending the request to the first streamed token of any kind, reasoning included, so a model that thinks for ten seconds before answering shows a ten-second latency. Output speed is the completion token count divided by the time between the first token and the end of the stream; the native token count from OpenRouter's generation record is used when it is available, because it counts what the model actually produced in its own tokenizer. End-to-end time is request sent to last token. For each measure I publish the median of the runs, which is less sensitive to a single slow request than the mean.

Some models think without streaming their thinking, then deliver the whole answer within about a second. Dividing hundreds of tokens by a near-zero stream time would produce an impossible speed. A run is flagged as a burst when nothing arrives for at least five seconds and then 300 or more tokens land in under 1.5 seconds. When most of a model's runs are bursts, its output speed is left blank; its latency and end-to-end time are still published, because they remain real measurements.

6 Cost

Cost per question is what OpenRouter actually billed for the quality questions, read from each request's usage block or generation record, added up and divided by the number of graded questions. Reasoning tokens are included because you pay for them too, which is why a model with a low list price can still be expensive per question if it thinks at length.

The bill is also checked against the list price: for every request, the harness takes the listed per-token price of the host that served it, multiplies it by the native token counts and compares the result with the amount charged. The cost figures in the result papers are always the billed amount. List prices come from OpenRouter's catalogue at test time and can change afterwards.

Frontier and non-frontier

A model is classed as frontier when its blended list price, $(3 \times \text{input} + 1 \times \text{output}) / 4$ in USD per million tokens from the OpenRouter catalogue at test time, is at least \$1.00; every other model is non-frontier. The threshold is a price boundary, not a claim about capability. It was chosen because list price is recorded for every model and can be checked by anyone; it is not derived from the scores.

7 Failures, retries and partial runs

- Retries. A request is attempted up to 3 times when it fails with HTTP 429 or a 5xx error, or with a host error such as "provider returned error", a timeout, a reset connection or an overloaded host. The harness waits 4 seconds before the second attempt and 8 seconds before the third. Only the final result is graded.
- Host failures are left out. A request that still fails says nothing about the model's knowledge, so it is excluded from the score rather than counted as wrong. A model that answers but gives no recognisable answer line is counted as wrong.
- Refusals count as wrong. When the model or its vendor's safety filter declines a question (OpenRouter's content_policy_violation error, or a response stopped with SAFETY), that is the model's answer, not a host failure, so it stays in the denominator as a wrong answer.
- No score for a broken run. If more than 20% of a model's questions fail, or the run stops before every question was asked, the model gets no score at all.
- Budget cap. Each run has a spending limit. Once it is reached the harness starts no new requests; any model left with an incomplete suite gets no score, as above.
- Merged runs. Not every model is re-tested in every run. Results are merged, and each model keeps the date of the run that actually tested it. If the question suite changes (detected by its hash), older scores from the previous suite are removed, so every score comes from the same questions.
- Unconfirmed answers are left out. An answer whose proof hash is not the hash listed for that file in its run's manifest (Section 8) is excluded from every published figure, in the same way as a host failure.

8 The evidence trail

1. Every request and response is written to its own JSON file: the request body and its SHA-256, the streamed text, the timings, the token counts, the host and the generation ID.
2. The SHA-256 hash of every one of those files goes into a run manifest, so changing any file afterwards would show.
3. The question suite and the harness script are hashed too, so anyone can confirm which questions and which code produced a run.

Scores were collected over 7 runs that used the same suite and the same request settings. The harness script changed between runs (3 distinct harness hashes, all version 0.1.0); each run's hash is listed below.

Table 3. Runs that contributed scored answers.

Run ID	Started (UTC)	Harness SHA-256 (prefix)	Answer proofs
20261025T170652Z	2026-10-25 17:06	ad9b5de9fa2c9608	80
20261025T172733Z	2026-10-25 17:56	daea9ded7865329a	1280

Run ID	Started (UTC)	Harness SHA-256 (prefix)	Answer proofs
20261025T195000Z	2026-10-26 03:44	daea9ded7865329a	320
20261025T200000Z	2026-10-26 05:22	daea9ded7865329a	2160
20261027T100730Z	2026-10-27 10:07	6d522eab4cfc8cdb	160
20261027T103825Z	2026-10-27 17:30	6d522eab4cfc8cdb	560
20261027T120517Z	2026-10-27 17:31	6d522eab4cfc8cdb	1200

9 Verification of this dataset

Before any figure was computed, a separate program, written independently of the harness and in a different language, re-checked the evidence and was set to stop on any failure:

- Suite hash recomputed from the suite file: matches the recorded value for all 72 scored models.
- Answer proof files: 5,760 of 5,760 hash to the SHA-256 stored with the answer.
- Proof identity: 5,760 of 5,760 proof records name the same model, item and OpenRouter generation ID as the results.
- Re-grading: 5,756 of 5,756 non-failed answers, re-graded from the raw answer text with an independent implementation of the rule in Section 3, give the recorded verdict and extracted answer. The 2 refusals have no answer text to re-grade; for each, the proof records the refusal error and the answer is scored wrong.
- Scores and per-subject counts recomputed from the verdicts: 72 of 72 models match.
- Billed cost: for 72 of 72 models, the per-request charges in the proof files sum to the recorded cost.
- Speed proof files: 215 of 215 hash to the stored SHA-256.
- Request settings: identical across all 7 scoring runs.

Answers carried forward between runs

14 answer proofs sit in one run's folder but record an earlier run ID (20261025T195000Z and 20261025T172733Z): the answer was made in the earlier run and carried into the later one. They are kept, because the later run's manifest lists each file with its exact hash, the record names the same model, item and OpenRouter generation ID, and the earlier run used the identical suite and settings. They are listed so a reader can apply a stricter rule.

Table 4. Answers carried forward from an earlier run (kept).

Model	Item	Made in run	Filed in run
openai/gpt-oss-20b	sgpqa-8d28e5d004	20261025T195000Z	20261025T200000Z
openai/gpt-oss-20b	sgpqa-7c99298107	20261025T195000Z	20261025T200000Z
ibm-granite/granite-4.0-h-micro	sgpqa-8d28e5d004	20261025T195000Z	20261025T200000Z
mistralai/ministral-3b-2512	sgpqa-8d28e5d004	20261025T195000Z	20261025T200000Z
mistralai/ministral-3b-2512	sgpqa-7c99298107	20261025T195000Z	20261025T200000Z
amazon/nova-micro-v1	sgpqa-8d28e5d004	20261025T195000Z	20261025T200000Z
amazon/nova-micro-v1	sgpqa-7c99298107	20261025T195000Z	20261025T200000Z
inception/mercury-2.5	sgpqa-8d28e5d004	20261025T195000Z	20261025T200000Z
inception/mercury-2.5	sgpqa-7c99298107	20261025T195000Z	20261025T200000Z
google/gemini-3.1-pro-preview	sgpqa-8d28e5d004	20261025T172733Z	20261025T195000Z
google/gemini-3.1-pro-preview	sgpqa-7c99298107	20261025T172733Z	20261025T195000Z
google/gemini-3.1-pro-preview	sgpqa-ec33474441	20261025T172733Z	20261025T195000Z
deepseek/deepseek-v4-pro-0813	sgpqa-8d28e5d004	20261025T172733Z	20261025T195000Z
deepseek/deepseek-v4-pro-0813	sgpqa-7c99298107	20261025T172733Z	20261025T195000Z

Manifest discrepancy and exclusions

5,748 of 5,760 answer proofs are listed in their run's manifest with the same hash. The remaining 12, all from run 20261027T120517Z, are listed with a different hash. In each case the proof file matches the hash in the results, so the answer and its verdict are internally consistent, but the manifest does not confirm them. That run's manifest file also differs from the one recorded in the results (1600 files listed against 1200 recorded), which indicates it was rewritten by a later invocation with the same run ID. Because the evidence trail rests on the manifest, these 12 answers are excluded from every published figure.

Table 5. Answers excluded because the run manifest does not confirm their proof hash.

Model	Item	Run
mistralai/mistral-medium-3-5	sgpqa-ae0c184921	20261027T120517Z
qwen/qwen3.8-2.4t-a95b	sgpqa-13ab6cdccf	20261027T120517Z
moonshotai/kimi-k2.7-code	sgpqa-0fb66d16c9	20261027T120517Z
cohere/command-a	sgpqa-6c535032a6	20261027T120517Z
rekaai/reka-flash-3	sgpqa-81712a85d0	20261027T120517Z
meta-llama/llama-4-scout	sgpqa-2f5f6db254	20261027T120517Z
qwen/qwen3.7-plus	sgpqa-d14d338a85	20261027T120517Z
google/gemma-4-26b-a4b-it	sgpqa-6c535032a6	20261027T120517Z
google/gemma-4-26b-a4b-it	sgpqa-cb1f3d75e5	20261027T120517Z
google/gemma-4-31b-it	sgpqa-e716eb0e8b	20261027T120517Z
inclusionai/ling-3.1-flash	sgpqa-e716eb0e8b	20261027T120517Z
nousresearch/hermes-4-405b	sgpqa-f42d8cc728	20261027T120517Z

In total 5,746 answers were graded, after excluding 2 host failures and 12 unconfirmed answers. 2 graded answers were refusals, scored as wrong. Of the other graded answers, 939 contained no recognisable answer line; 927 of these ended with finish reason "length" and 932 had an empty answer text, which is consistent with the output-token limit being reached during reasoning.

10 Reproducing the results

The questions are public: every one comes from the open datasets in Table 2, under their own licences. The models are public too, through OpenRouter, so anyone with an OpenRouter API key can send the same prompts with the same settings and compare the answers. Because sampling is at temperature 0, most models give the same or a very similar answer on a repeat run, though hosts can differ slightly and models are sometimes updated behind the same ID.

1. Download the per-question evidence (CSV or JSON). Each row gives the model, item ID, dataset identifier, expected and extracted letter, verdict, finish reason, serving host, run ID, OpenRouter generation ID and proof SHA-256.
2. Recompute any score from the rows where errored and excluded_not_in_manifest are both false.
3. Look up the question in the public dataset by its identifier, build the prompt as in Section 3, and send it to the same model ID with the settings in Table 1.
4. To audit a record, request its proof file from the author: its SHA-256 must equal the published proof_sha256, and the generation ID inside it can be checked against OpenRouter's generation endpoint by the account that made the request.

The proof files and the harness itself are not published in bulk, because they contain full model outputs. To check a specific number, a question's result or the harness hash, contact the author with the model and run concerned.

11 Limitations

- Sample size. 80 questions give the uncertainty described in Section 4. Small gaps between models are noise, and subject results with only a few questions each are very noisy.
- One kind of task. These are expert-level multiple-choice questions. They say a lot about knowledge and reasoning, and nothing directly about coding agents, long documents, writing style or tool use.
- Fixed budget. Every model gets the same low reasoning effort and token limit. A model that spends the whole budget reasoning returns no answer line and is scored as wrong; higher settings may change its result.
- One location. Speed and latency are measured from one client machine. Network distance to the host affects time to first token, so your own numbers may be higher or lower.
- Routing and hosts. OpenRouter may send a request to any host that serves the model. Hosts differ in speed, and occasionally in quantisation or configuration. The host that served each request is recorded.
- Prices change. Billed costs are what was paid on the test date. List prices may have moved since.

12 What I don't do

- No paid placement. Vendors can't pay to be included or ranked.
- No numbers copied from model cards or vendor announcements.
- No hand-editing of results. If a number is wrong, the cause is fixed and the model is re-run. Exclusions are made only by the published rules in Section 7, and every excluded answer is listed in Section 9.

13 Data availability

The per-model results (llm-evaluation-models.csv) and the per-question evidence (llm-evaluation-evidence.csv, llm-evaluation-evidence.json) are published with this paper at suhasbhairav.com/research under CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>). The questions themselves are available from the source datasets under their licences (ODC-BY 1.0 (SuperGPQA), MIT (MMLU-Pro)).

References

- [1] E. B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209-212, 1927. <https://doi.org/10.1080/01621459.1927.10502953>
- [2] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153-157, 1947. <https://doi.org/10.1007/BF02295996>
- [3] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65-70, 1979. <https://www.jstor.org/stable/4615733>
- [4] M-A-P Team et al. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines. arXiv:2502.14739, 2025. <https://arxiv.org/abs/2502.14739>
- [5] Y. Wang et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. arXiv:2406.01574, 2024. <https://arxiv.org/abs/2406.01574>
- [6] OpenRouter. API reference: chat completions and generation records. <https://openrouter.ai/docs/api-reference/overview>