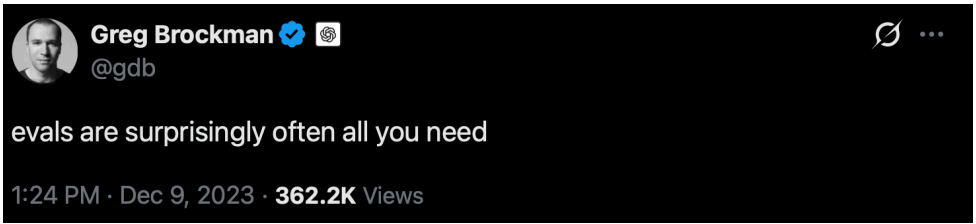
Hidden Measurement Error in LLM Pipelines Distorts Annotation, Evaluation, and Benchmarking

Solomon Messing

Center for Social Media, AI, and Politics, New York University · MLCommons

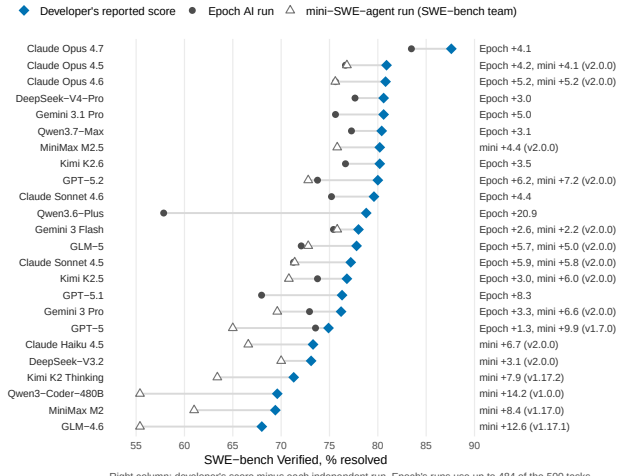
NeurIPS 2026 · arXiv 2604.11581

Evals drive AI development



Evals drive deployment, safety standards, and research.

Benchmark hacking: publish the best setup



Right column: developer's score minus each independent run. Epoch's runs use up to 484 of the 500 tasks.
Data: developers' launch posts; Epoch AI (CC BY 4.0); SWE−bench leaderboard (CC BY−NC 4.0).

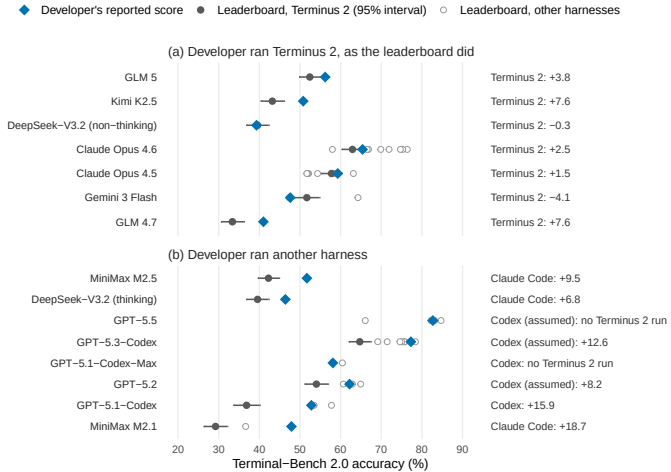
On SWE-bench, labs' launch scores beat independent runs by

4 - 5

points on average

+4.3 over Epoch AI
+5.1 over mini-SWE-agent

Terminal-Bench 2.0: lab runs vs. the leaderboard



Right column: the developer's harness and its score minus the leaderboard's Terminus 2 run.
Data: developers' launch posts; tbench.ai Terminal−Bench 2.0 leaderboard (read Nov 8, 2026).

Different harness and settings (b)

7 - 19 points above the default harness

Even one harness isn't one setup

In Qwen's reruns, the tool-call format alone moved scores by up to 10 points.

Benchmark hacking vs. benchmaxxing

Benchmaxxing

post-training on tasks
that mimic a benchmark

Benchmark hacking

trying many harnesses, prompts,
or compute settings and
reporting the best

Evaluators do try to combat other hacks,
but there's little to prevent this one.

Autoraters have the same problem

31%

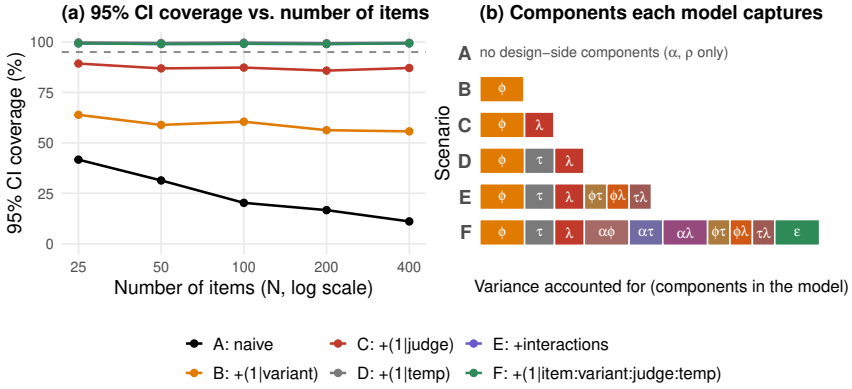
of hypothesis tests on LLM-labeled data reached wrong conclusions under other models and prompts

68%

of significant effects could flip sign under adversarial choices

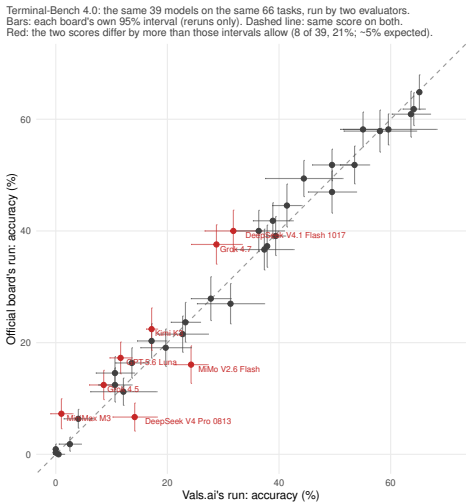
Baumann, Röttger, et al. (2025), Large Language Model Hacking

Naive CIs get worse with more data



Current CIs (A) treat items as a sample
but hold the prompt, judge, and harness fixed.

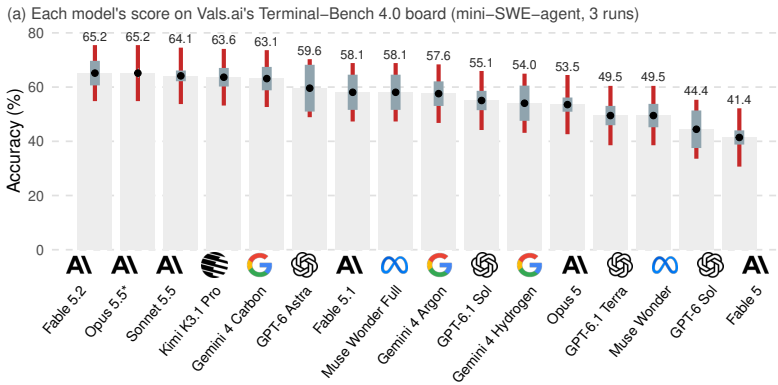
Terminal-Bench 4.0: two boards, same models



If each board's 95% CI accounts for noise properly, about 5% of the 39 pairs will differ by chance.

21% do

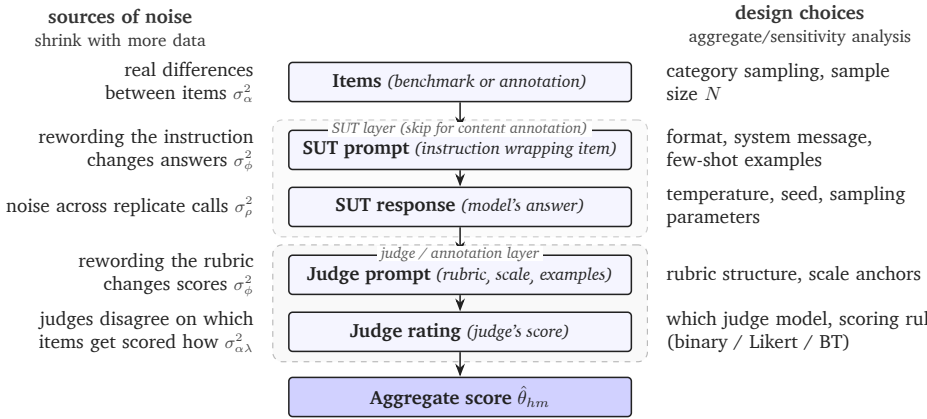
With reasonable CIs, a 2-point lead looks silly



Grey: CIs based on rerun noise

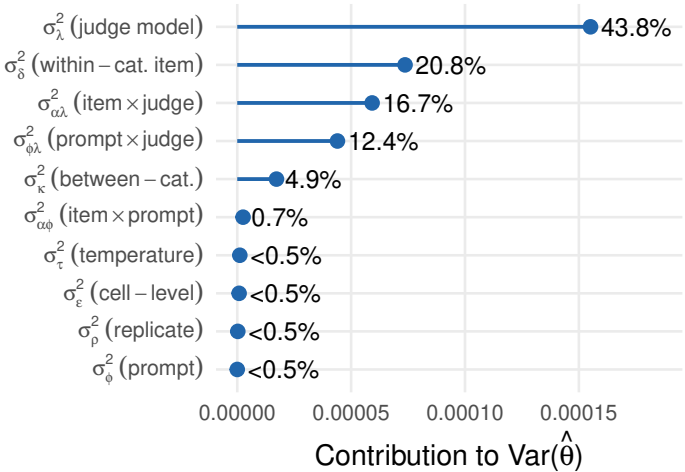
Red: also accounts for which tasks were picked and who runs the model

Total Evaluation Error (TEE)



We use a mixed model to split the variance among items, prompts, judges, temperatures, reruns, and their interactions.

AILuminate: judges account for most variance

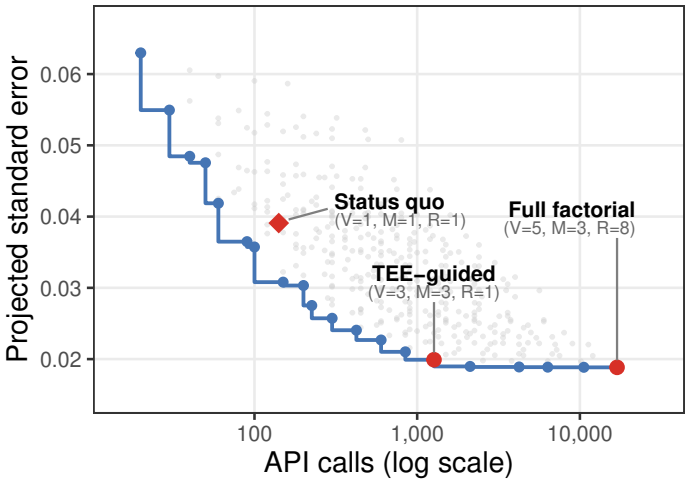


73%

of the variance comes from the three judge-related terms (68 - 83% after dropping any one judge)

In our safety and ideology applications, naive SEs are 53 - 70% smaller than the corrected SE.

We can also reduce the error itself

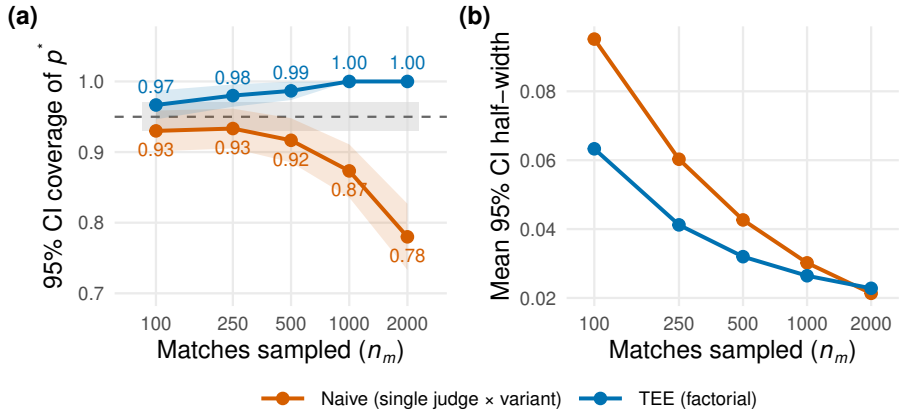


about ½

the error, with 3 prompts and 3 judges instead of one

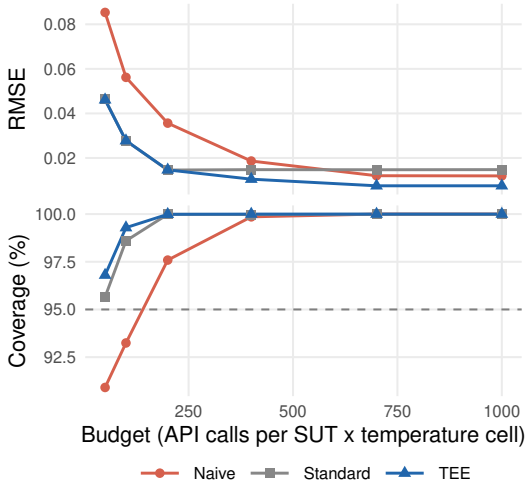
for 9× the API calls

Chatbot Arena: more data, worse coverage



Naive CIs (one judge, one prompt) cover 78% at 2,000 matches.

Closer to ground truth



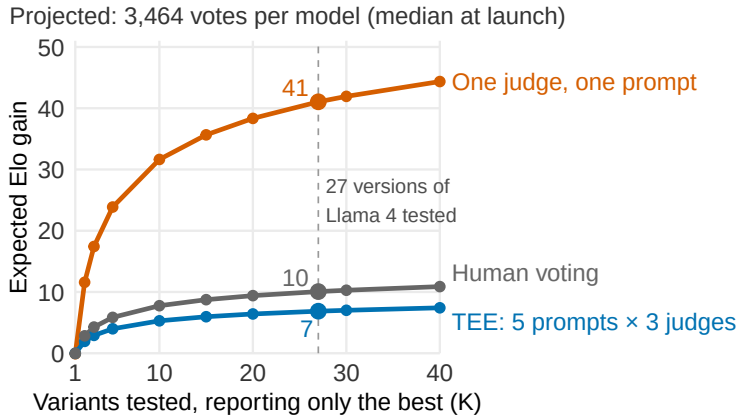
MMLU

about half as far from true accuracy, same API budget

Chatbot Arena

3 judges × 5 prompts matched the human vote 74.9% of the time, vs. 67.0% for one of each

Averaging also curbs benchmark hacking



27 versions of Llama 4 were tested on Chatbot Arena before launch.

Singh, Hooker, et al. (2025), The Leaderboard Illusion

So what should you do?

Average across multiple judges	Sample more items
Don’t rest results on one judge model or prompt template	Run a small pilot first

TEE focuses on reliability. Whether a judge is correct and whether the items cover the construct are validity questions that need external ground truth.

Bottom line

When LLM evals report error bars at all, the bars are too narrow.

TEE widens them to include judge and prompt noise, and averaging over both makes the scores themselves more accurate.

Paper **R package**

Paper	arxiv.org/abs/2604.11581
SI Appendix	solomonmg.github.io/pdf/totalevalerror_si.pdf
R package	github.com/SolomonMg/totalevalerror
Replication	github.com/SolomonMg/totalevalerror-replication

Poster in Sydney: Wed, Dec 9, 10 AM - 1 PM AEDT, Hall 1-4

Thanks

Thanks to Peter Mattson and MLCommons, and to NYU's Center for Social Media, AI, and Politics, for supporting this work.