The Stack Is Part of the Benchmark: Serving-Induced Evaluation Instability in Vision-Language Models

Saloni Garg and Sukhesh Garg

Working paper · Draft v0.1 · November 2026

Abstract

Benchmark scores for vision-language models are reported as properties of the model. We show they are also properties of the software stack that serves the model. Holding the weights, the items, the decoding settings, and the input pixels exactly fixed, we evaluate Qwen2.5-VL-7B on 4,250 items from TextVQA, ChartQA, and POPE under two widely used inference stacks, Hugging Face Transformers and vLLM. The pooled benchmark score moves by only 0.26 points between the stacks, yet 127 individual verdicts flip (3.0% of items; bootstrap 95% CI 2.5 to 3.5) and 7.3% of answers change. The aggregate is stable because the flips are close to symmetric; the individual judgments underneath it are not. A repeat run of the reference stack flips zero items, so the effect is not sampling noise. Flips concentrate on items where the model itself is near its decision boundary, but roughly a quarter of answer flips occur on items the reference stack answers with high confidence. Batch size alone flips an order of magnitude fewer verdicts (0.2 to 0.3%), and the backend effect persists at both batch sizes and under each stack's default image preprocessing. In this study, flagging the items where the two stacks disagree (7.3% of items) captured every verdict flip, which we offer as an empirical observation and a candidate evaluation practice, not a general guarantee. Our results argue that per-item stability, not only aggregate accuracy, should be reported for VLM evaluations, and that the serving stack belongs in the methods section of every benchmark paper.

1. Introduction

A benchmark number is usually read as a statement about a model. When two groups report different scores for the same model on the same benchmark, the difference is attributed to prompts, decoding, or evaluation code. The inference stack itself, the library that loads the weights and executes the forward passes, is treated as transparent plumbing.

This paper tests that assumption directly. We fix everything that is supposed to matter: one checkpoint, one frozen item set, greedy decoding, and, in our primary arm, input images preprocessed once and verified to be pixel-identical across stacks. We then vary only the serving stack, comparing Hugging Face Transformers, the reference implementation most evaluation code is built on, against vLLM, the stack most production deployments actually run.

The stacks do not agree. On 4,250 items, 7.3% of answers differ between the two stacks and 3.0% of items change scored verdict, even though the pooled accuracy differs by only 0.26 points. The flips are close to symmetric, which is why the aggregate barely moves: each stack is right on a different subset of items. A leaderboard can therefore be stable while a substantial fraction of the individual judgments underneath it are

stack-dependent. Any conclusion drawn at the level of individual items, including error analysis, failure taxonomies, and per-category claims, inherits that instability.

Three controls make the effect hard to explain away. First, inputs in the primary arm are identical by construction and verified at the tensor level. Second, a byte-for-byte repeat of the reference configuration reproduces every one of the 4,250 verdicts, so the pipeline is deterministic and the noise floor is zero. Third, the effect is much larger than the batch-size effect measured in the same harness, and it persists across batch sizes and preprocessing regimes.

We also ask where the flips live. Using the reference stack's own first-token margins, flips concentrate on near-tie items, as expected if small numerical differences are amplified at the decision boundary. But the concentration is not total: about a quarter of answer flips involve items the reference stack answers confidently, which rules out a pure rounding story and points at implementation-level differences in the vision and language computation paths.

Finally, the instability is detectable without ground truth. In our study, the 7.3% of items where the two stacks disagree contain all 127 verdict flips, so cross-stack disagreement is a practical flag for unstable items. We present this as a finding of this study rather than a general solution: it says nothing about items both stacks get wrong together, and its behavior with other stack pairs is an empirical question. Even with those caveats, it suggests a cheap evaluation hygiene practice: report cross-stack disagreement alongside accuracy, and treat flagged items as unstable rather than as evidence about the model.

Contributions: - A controlled, preregistered measurement of serving-stack-induced instability in VLM evaluation, with pixel-identical inputs and a zero measured noise floor. - A decomposition of the effect across backend, batch size, and preprocessing, showing the backend dominates the batch effect by an order of magnitude. - A margin analysis localizing flips to near-boundary items, with a quantified confident-flip tail. - An empirical mitigation finding: in this study, dual-stack disagreement flagging captured all verdict flips at a 7.3% flag rate, with stated limits. Plus reporting recommendations for benchmark papers.

2. Related work

- Serving and inference systems: vLLM and PagedAttention; Transformers as reference implementation. Position: systems papers optimize throughput and latency; none treats the stack as an evaluation variable.
- Evaluation reproducibility: Meta-Video documents backend discrepancies for video VLMs and traces them to preprocessing divergence, proposing evaluation cards. Our aligned arm removes preprocessing divergence by construction and still finds the effect for still images, which is the unclaimed part. OmniEvaluator (infrastructure for standardized VLM evaluation) motivates standardization but does not measure per-item flips at matched preprocessing.
- Numerical nondeterminism in LLM inference: batch-size and kernel nondeterminism discussions in the LLM serving literature; our batch arm quantifies this for VLM evaluation and shows it is the smaller term.

- Benchmark validity and per-item analysis: work on annotation error and item-level instability under perturbations. Our perturbation is the stack, not the input.

3. Experimental setup

- Model: Qwen/Qwen2.5-VL-7B-Instruct, BF16 checkpoint, snapshot cc594898137f460bfe9f0759e9844b3ce807cfb5. Same weights in every cell.
- Stacks: Hugging Face Transformers 5.12.1 (torch 2.13.0, CUDA 13.0) and vLLM 0.29.0. Hardware: NVIDIA A100-SXM4-80GB.
- Decoding: greedy (temperature 0) everywhere; max output tokens 8 (POPE), 32 (TextVQA), 64 (ChartQA).
- Items: frozen set of 4,250. TextVQA validation 1,500; ChartQA test 1,250; POPE test 1,500 (random, popular, adversarial splits, 500 each). Identical item order in every cell.
- Preprocessing arms. Aligned: images pre-resized once with PIL LANCZOS to dimensions that make the Qwen smart-resize an identity map (multiples of 28, total pixels between 256 x 28 x 28 and 1280 x 28 x 28), stored as PNG, fed to both stacks. Verification: all 4,250 images satisfy the identity conditions; sampled processor outputs have patch counts exactly equal to the stored image grid. Default: original images, each stack's own preprocessing (secondary arm).
- Design: backend (HF, vLLM) x batch (1, 32) x preprocessing (aligned, default), plus an exact repeat of the reference cell (HF, batch 1, aligned) as the noise-floor control. Precision fixed at BF16 (FP8 arms were dropped before data collection; see limitations).
- Scoring: normalized exact match for TextVQA (any of the human answers) and ChartQA (numeric tolerance), yes/no parsing for POPE. Metrics: per-item verdict flip rate (primary), answer change rate, per-benchmark accuracy, McNemar tests with Holm correction, bootstrap 95% CIs (10,000 resamples, item level). First-token logit margins logged for every item in the HF batch-1 cells.
- Preregistration: hypotheses, endpoints, and the decision rule were frozen before running (internal record: cvpr-pilots PREREGISTRATION.md, 2026-11-03, with timestamped amendments for the FP8 removal and one runner padding bug that was caught and rerun; disclose in supplementary).

4. Results

4.1 The backend changes verdicts (primary result)

Table 1. Primary comparison, aligned arm, batch 1, BF16. - Accuracy: HF 85.74%, vLLM 86.00% (pooled delta 0.26 points). - Verdict flips: 127 of 4,250 (2.99%; bootstrap CI 2.49 to 3.51). - Answer changes: 311 of 4,250 (7.32%; CI 6.52 to 8.12). - Noise floor: repeat of the HF cell flips 0 of 4,250. - Per benchmark verdict flips: TextVQA 60 of 1,500 (4.0%), ChartQA 43 of 1,250 (3.4%), POPE 24 of 1,500 (1.6%). - Per benchmark answer changes: ChartQA 11.8%, TextVQA 9.3%, POPE 1.6%. - McNemar per benchmark (accuracy differences): not significant after Holm (adjusted p 0.13 to 0.16). Report this prominently: the aggregate difference is statistically invisible while 127 verdicts flip. Flips are near-symmetric (reference-

only-correct vs stack-only-correct: TextVQA 22 vs 38, ChartQA 28 vs 15, POPE 8 vs 16). - The effect replicates at batch 32: verdict flips 3.18%, answer changes 7.34%.

4.2 Decomposition: backend dominates batch and rivals preprocessing

Table 2. Verdict flip rates (answer change rates in parentheses), aligned arm unless noted. - Backend (HF vs vLLM, batch 1): 2.99% (7.32%). - Backend (batch 32): 3.18% (7.34%). - Batch (1 vs 32) within HF: 0.33% (1.25%). - Batch within vLLM: 0.24% (0.92%). - Preprocessing (aligned vs default) within HF, batch 1: 1.62% (4.09%). - Preprocessing within vLLM, batch 1: 1.81% (3.95%). - Backend within the default arm, batch 1: 2.89% (7.18%). Reading: preprocessing divergence matters (consistent with Meta-Video for video), but even with preprocessing removed the backend term is the largest single factor, and it is roughly ten times the batch term.

4.3 Where the flips live: margin analysis

Figure 1. Flip rate by reference-margin band; margin distribution of flipped vs stable items. - Median first-token margin: 0.875 (verdict flips) vs 4.875 (stable). Answer flips: 1.375 vs 5.000. - Bottom margin quartile contains 66.1% of verdict flips and 54.0% of answer flips; bottom decile contains 37.8% of verdict flips. - Confident tail: 13.4% of verdict flips and 23.8% of answer flips occur in the top half of the margin distribution. State plainly that near-tie numerics explain most but not all of the effect.

4.4 Mitigation: disagreement flagging

- Flag rule: run both stacks, flag items where normalized answers disagree. In this study: flag rate 7.32%; captures 127 of 127 verdict flips.
- Limits, stated in the paper, not buried: (1) Agreement is not correctness. The stacks agree on 3,939 items and are both wrong on 453 of them (11.5% of agreed items); flagging detects instability, not error. (2) With three or more stacks the flagged set is the union of pairwise disagreements and can only grow; the flag rate and capture rate for other stack combinations are empirical questions this study does not answer. (3) The rule needs a second stack run, cheap relative to human adjudication but not free.
- Recommendation box: report stack and version, decoding, preprocessing, and a stability statistic (cross-stack or cross-batch flip rate) alongside accuracy in benchmark papers.

5. Discussion

- Why aggregates hide it: flips are near-symmetric, so accuracy is a difference of two large, nearly equal flip counts. Leaderboard gaps of the size routinely reported between models (often under 1 point) are smaller than the verdict churn a stack change induces on the same model.
- Mechanism, observations separated from hypotheses. Observations: flips concentrate at low first-token margin in the reference stack; a confident tail exists (13.4% of verdict flips in the top margin half); the batch term is an order of magnitude smaller than the backend term; the effect persists across preprocessing regimes. Hypotheses (labelled as such, not established by our data): kernel selection

and reduction order in BF16 attention and GEMM paths, vision tower batching and padding handling, and sampling implementation differences produce small logit differences that flip near-tie items; the confident tail suggests that on some inputs the two paths diverge by more than rounding, for reasons we do not isolate. We claim the phenomenon and its localization, not a root cause.

- Relation to preprocessing: the default arm shows preprocessing adds its own flips, but the aligned arm isolates a residual backend effect that preprocessing fixes cannot remove.
- Implications: reproducibility claims that name the model but not the stack are incomplete; per-item analyses (error taxonomies, category claims, case studies) are the most exposed; multi-stack evaluation is a cheap robustness check.

6. Limitations

- One model family and one size (Qwen2.5-VL-7B). The phenomenon is demonstrated, not shown to be universal; magnitude will vary by model and stack versions.
- Two stacks, one hardware type (A100), one precision (BF16). FP8 evaluation was planned and dropped because both FP8 routes failed on this stack; precision as a factor is future work.
- Short-answer benchmarks only; open-ended generation may behave differently.
- Mechanism is localized (margin analysis) but not isolated to a single kernel or code path.

7. Conclusion

Restate: identical weights, items, and pixels; different stack; 7.3% of answers and 3.0% of verdicts change; aggregates hide it; disagreements are detectable and flaggable. The serving stack is part of the measurement instrument and should be reported and, where possible, varied as a robustness check.

LLM usage statement

Large language models were used for code assistance in the evaluation harness, for literature search support, and for language editing of this manuscript. All experimental design decisions, preregistered hypotheses and decision rules, analyses, and conclusions are the authors' own, and all reported numbers were computed from the logged run outputs. (Adjust to match the final author policy and the exact CVPR wording before submission.)