VLLM · SGLANG · TENSORRT-LLM

# Same model. Same H100. Three serving engines.

**3** vs **82**

distinct outputs for the same request, 1,000 times under load — SGLang vs vLLM, both at defaults

**–37%**

cost per token from one setting (FP8 KV cache: vLLM fits 64 users instead of 32), with no detectable accuracy change (resolution ~3 points)

**1 flag**

made vLLM's batch-invariant mode exact on gpt-oss-120b

gpt-oss-120b, MLPerf accuracy method, 1–64 users, Nsight Systems profiles. **8 findings, $93 of cloud time.**
Swipe →

HOW IT WAS MEASURED

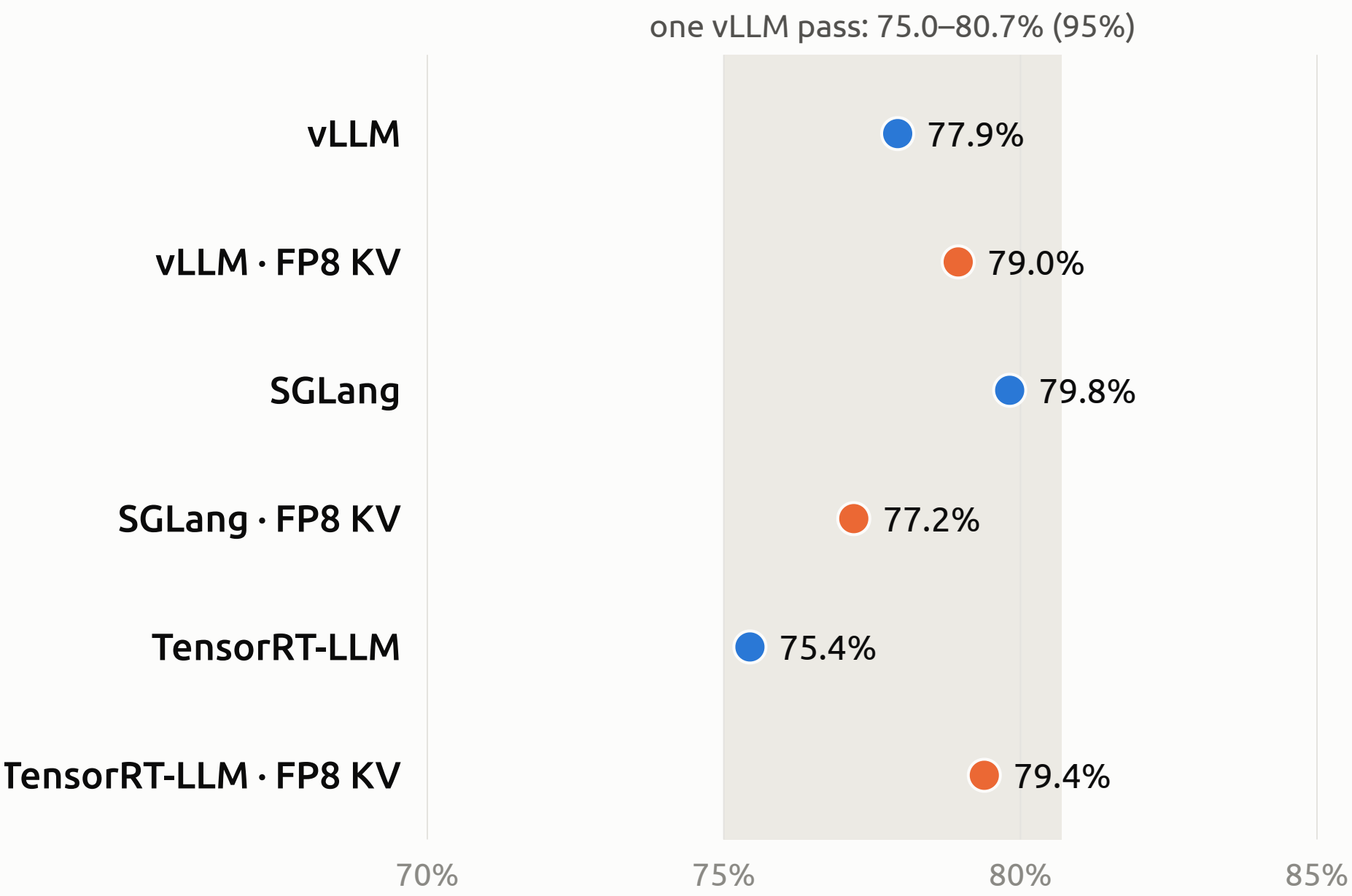
# One GPU. Every run sealed.

- **Hardware:** 1× NVIDIA H100 SXM5 80GB on a cloud VM; the same physical GPU in every session
- **Engines:** vLLM 0.33.0 · SGLang 0.5.23 · TensorRT-LLM 1.4.0rc2, images pinned by digest
- **Accuracy:** MLPerf v6.0 gpt-oss-120b method — AIME 2025 + GPQA Diamond, 228 questions; vLLM repeated the full 1,230-row set as the noise floor
- **Speed:** 1–64 concurrent users, 3 repeats per level; a level runs only if users × 2,304 tokens (the per-request limit) fits the KV cache
- **Repeatability:** identical temperature-0 requests, 512 tokens, 24 concurrent clients
- **Profiles:** Nsight Systems at 16 users
- **Guard rails:** hard $100 cap, two independent watchdogs, every run hash-sealed

FINDING 1 · ACCURACY

# No engine and no FP8 setting beat the noise

All six configurations land inside the range a single vLLM pass falls in by chance. At 228 questions, differences under ~3 points cannot be seen.




MLPerf v6.0 accuracy, 228 questions (AIME 2025 + GPQA Diamond)

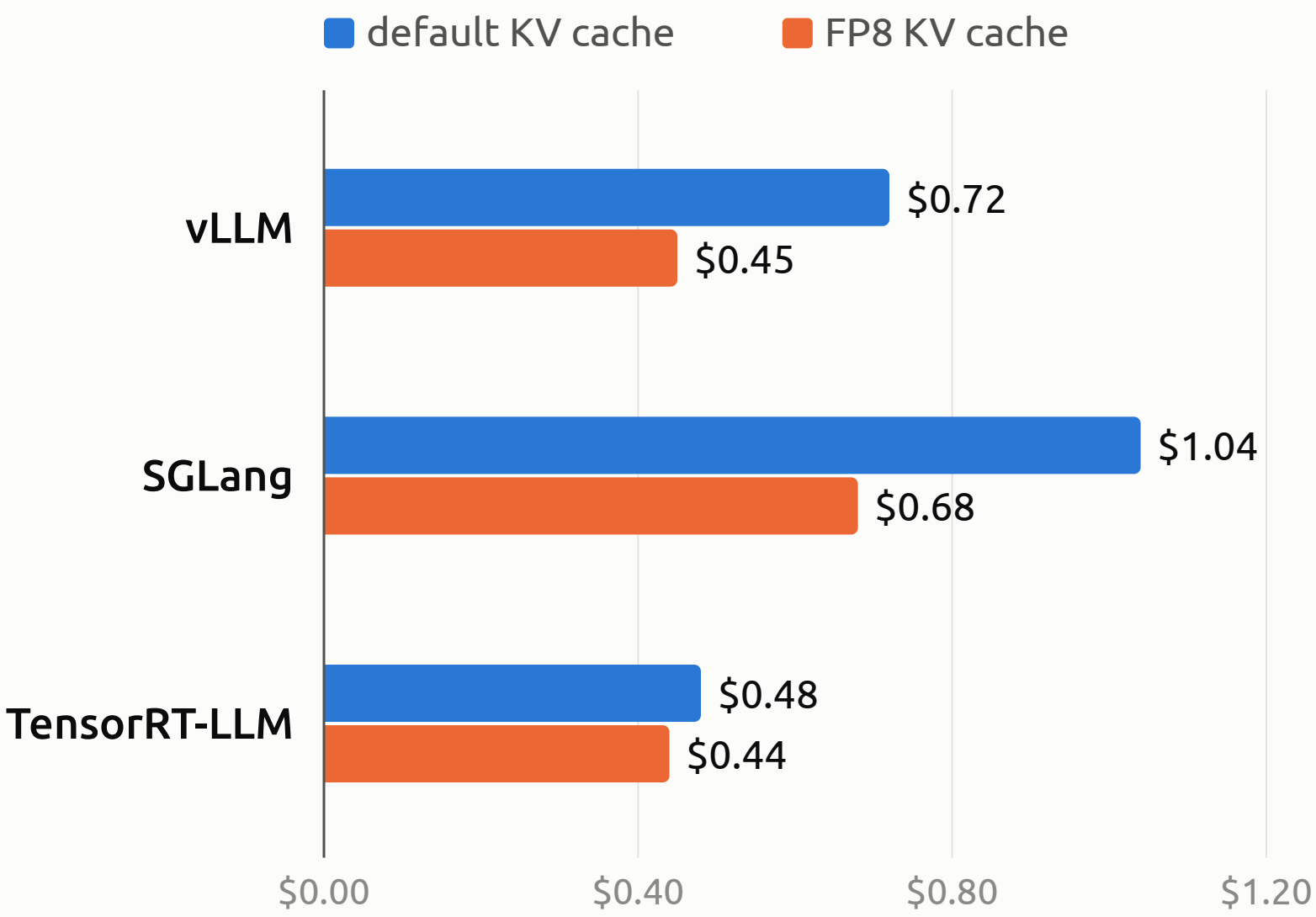
Blue = default KV cache · orange = FP8 KV cache. vLLM shows its mean over 1,230 rows.

FINDING 2 · COST

# FP8 KV cache: up to 37% cheaper per token

It doubles the KV pool, so twice as many users fit at 2,304 tokens each — with no detectable accuracy change (finding 1). vLLM −37%, SGLang −35%; TensorRT-LLM already fit 64 users, so only −8%.

Most users that fit: vLLM 32 → 64 users · SGLang 16 → 32 users · TensorRT-LLM 64 → 64 users

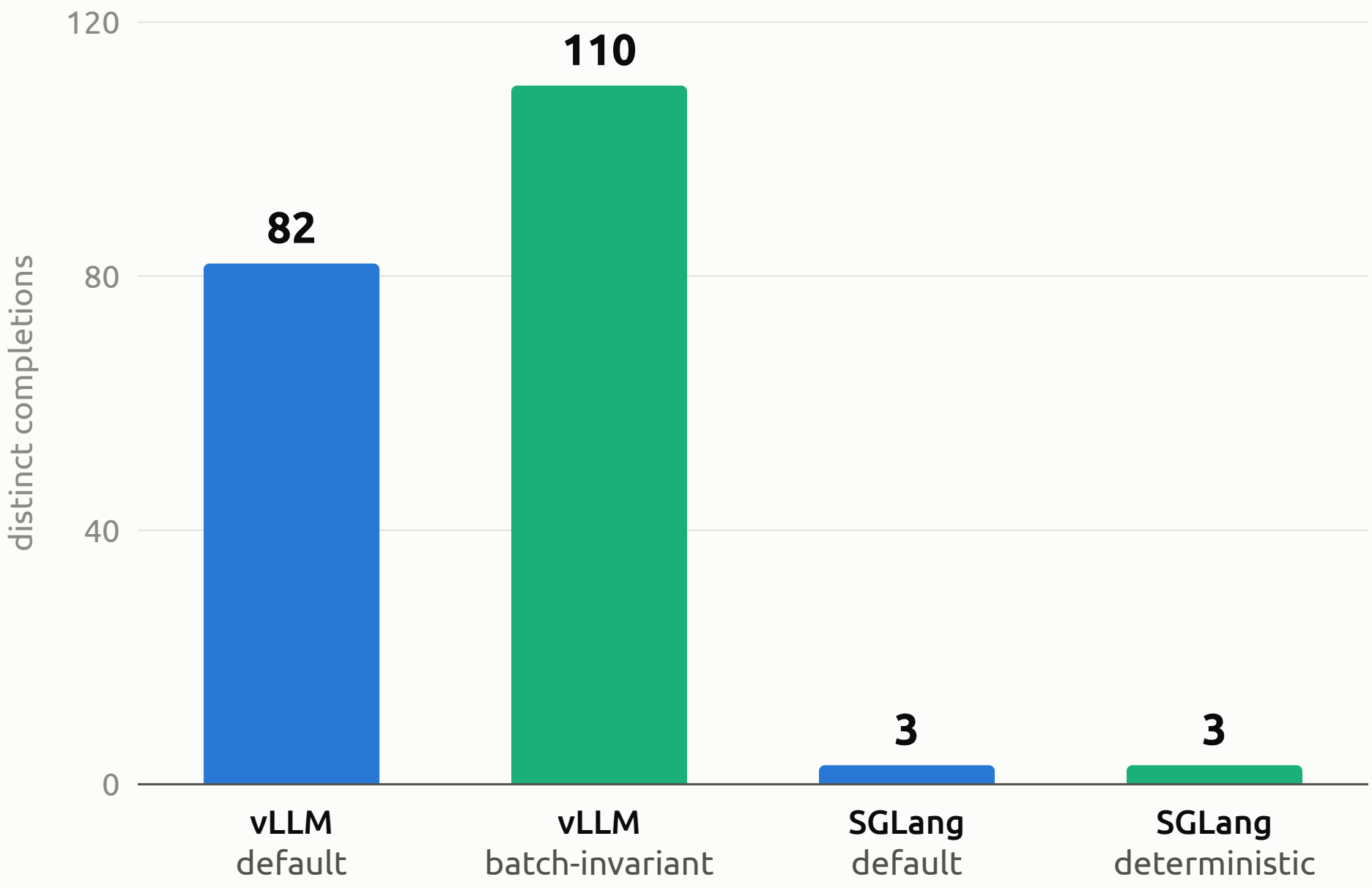



GPU rental only ($5.24/h). Synthetic load: ~1k-token prompts, 1,024 output tokens.

FINDING 3 · REPEATABILITY

# Same request, 1,000 times

Temperature 0, identical prompt, 24 concurrent clients. SGLang returned 3 distinct outputs; vLLM returned 82 — and 110 with its batch-invariant mode switched on. Outputs stop at 512 tokens, before a final answer.

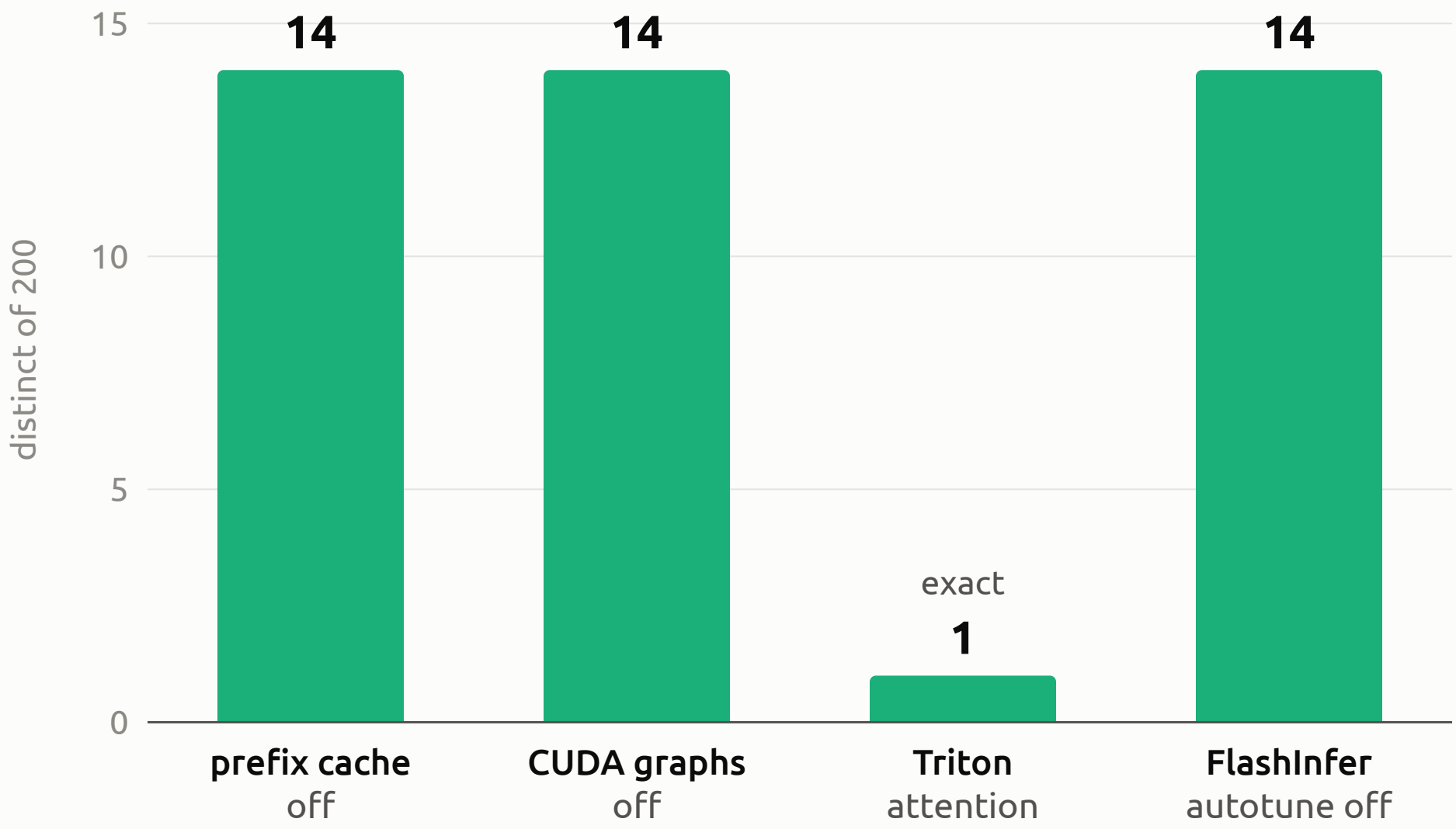



Blue = default · aqua = deterministic / batch-invariant mode.

FINDING 4 · ROOT CAUSE

# The attention backend broke vLLM's batch invariance

Batch-invariant mode plus one change at a time, 200 requests each. Only switching from FlashAttention 3 to Triton attention made the output exact on H100: **1 distinct output, 200 of 200 identical to the solo run**.

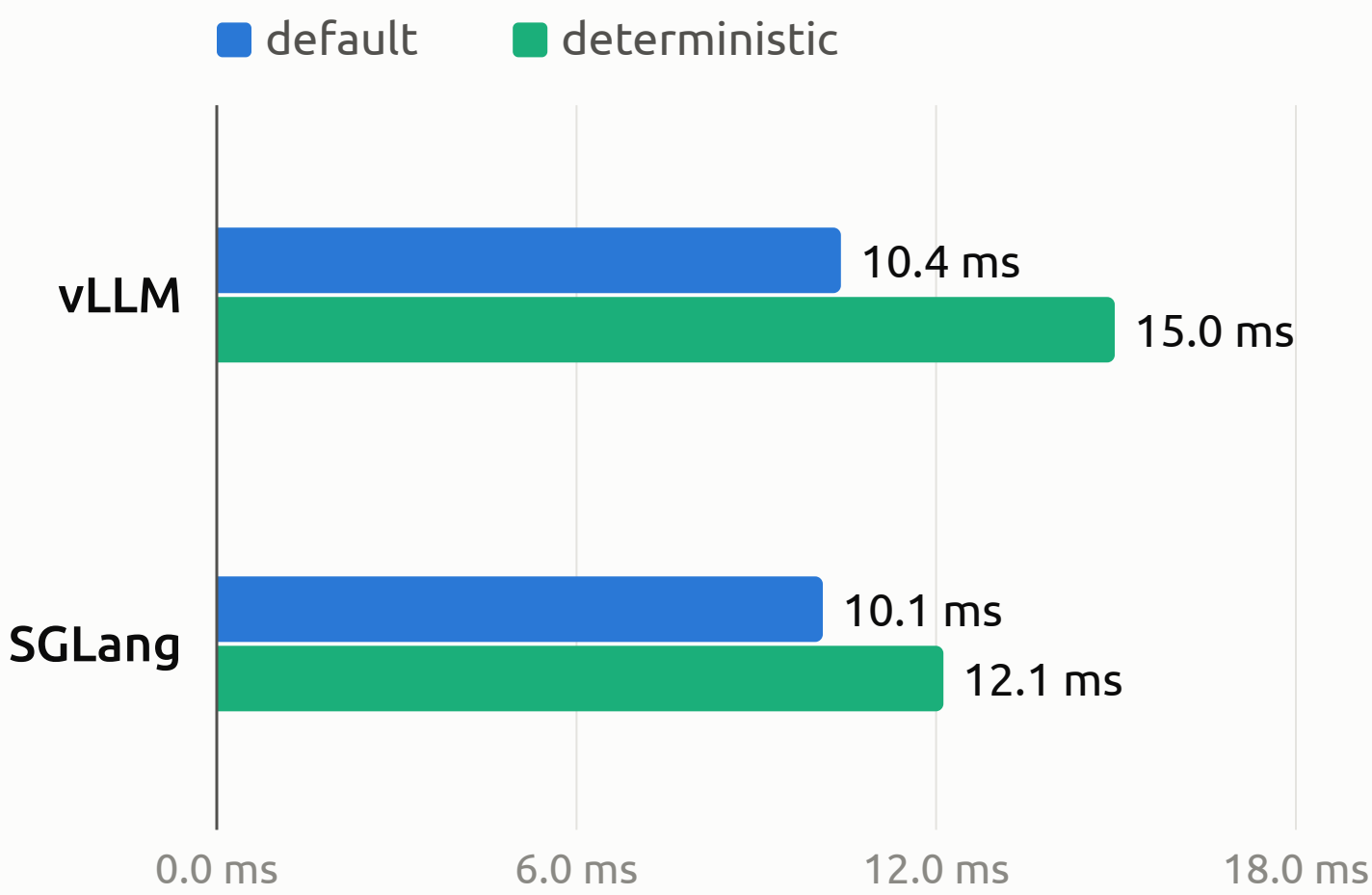



The speed cost of Triton attention on H100 is not measured yet.

FINDING 5 · PRICE OF DETERMINISM

# Exact output costs decode speed

vLLM swaps its dense GEMMs for a persistent Triton matmul and changes MoE kernels: +44% per decode step. SGLang swaps cuBLAS for DeepGEMM: +20%. Per token at 16 users: +30% and +16%.

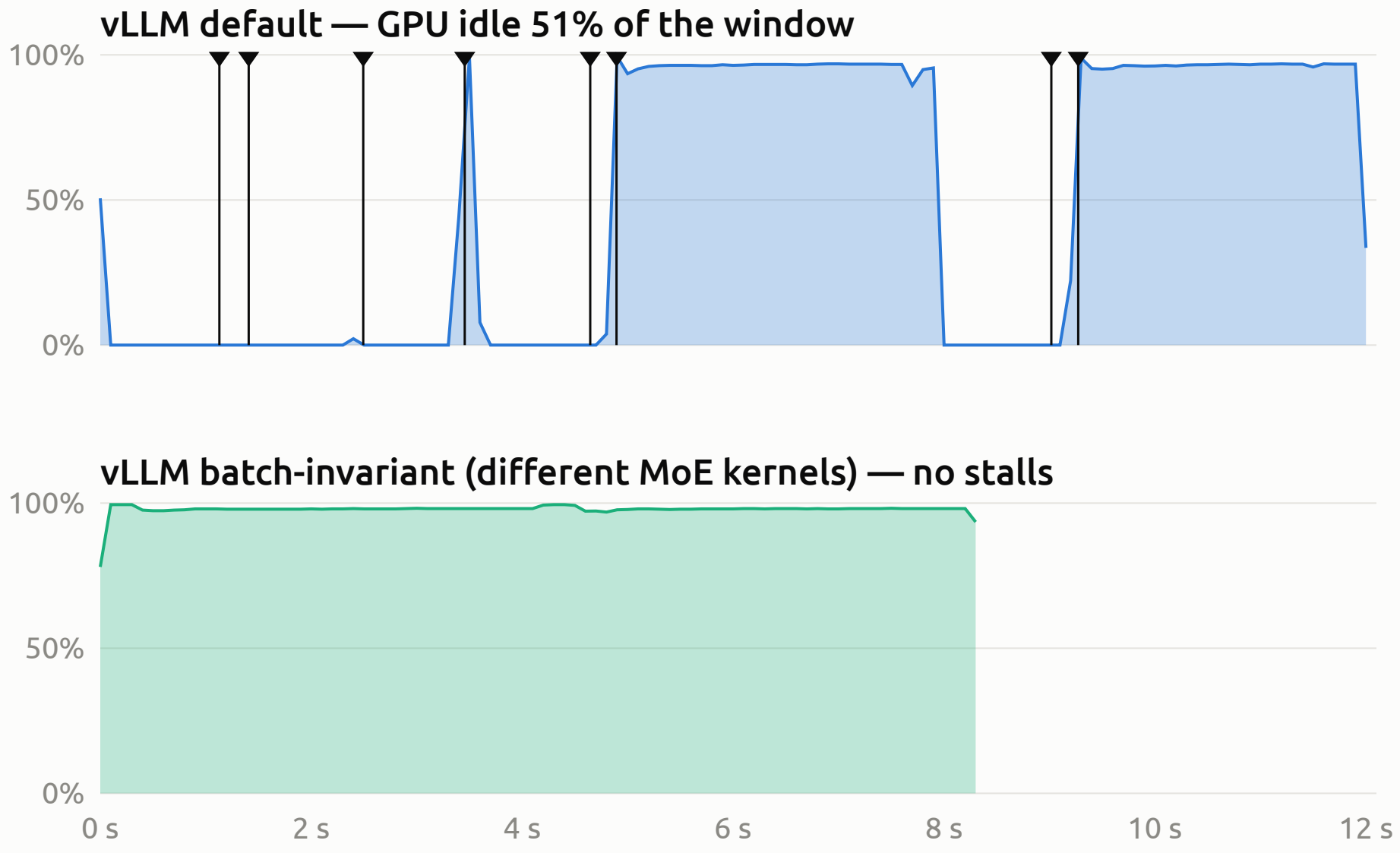



milliseconds per decode step (Nsight Systems, 16 users)

FINDING 6 · NSIGHT

# A “ready” server still compiled kernels under load

After warm-up, vLLM's default gpt-oss path left the GPU idle for ~1 s at a time — 8 times in 12 s — and each gap ended with a new GPU module load (▼), consistent with JIT compilation for batch sizes not seen in warm-up.

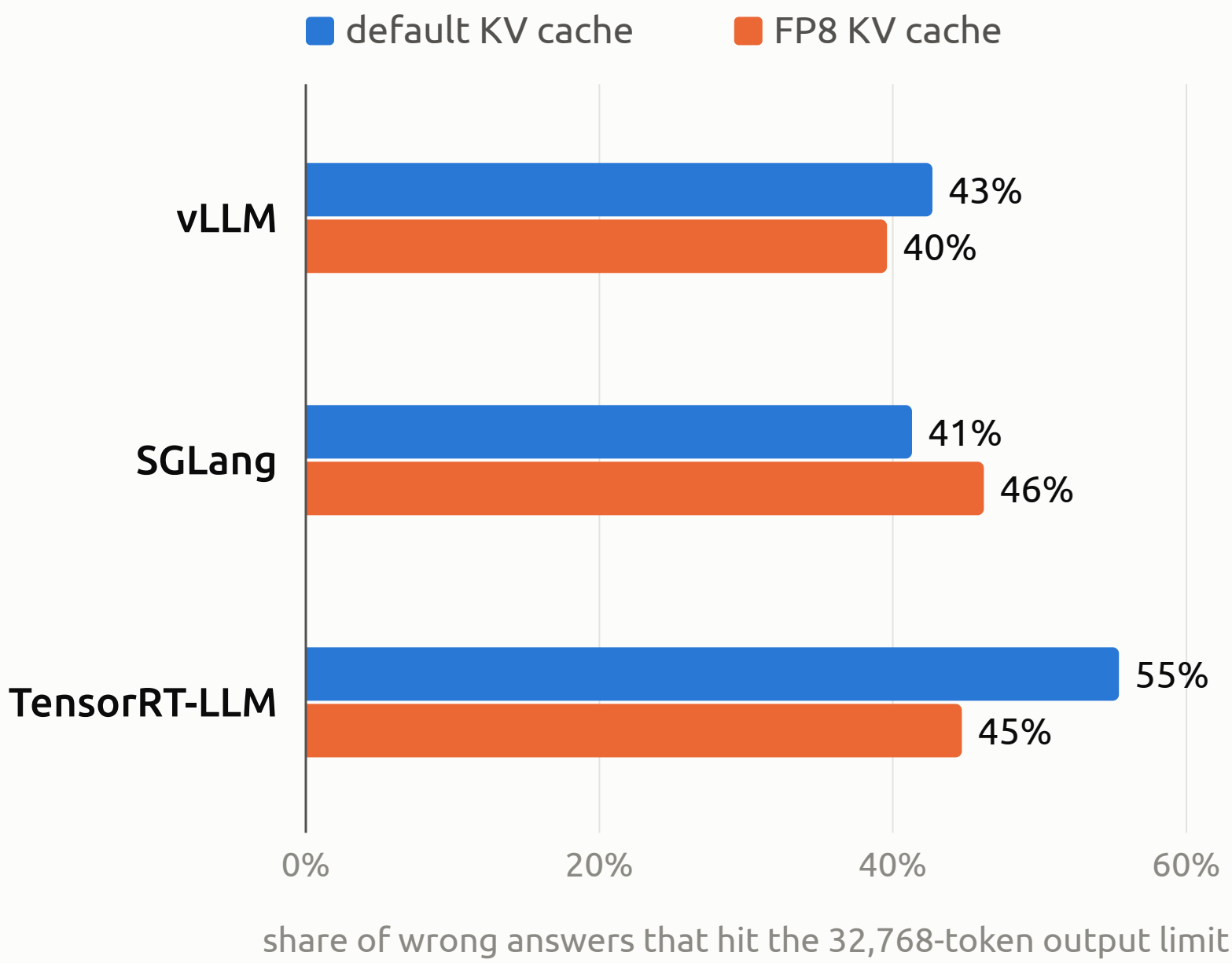



▼ = cuModuleLoadData. Same capture for SGLang: 2 short gaps (0.3 s in total).

FINDING 7 · ACCURACY BUDGET

# About half of the wrong answers never finished

10–14% of all answers hit the 32,768-token limit, and they make up 40–55% of the wrong answers. It is the same ~44 hard questions on every engine — a model property. At ~74 tokens/s per user, such an answer runs ~7 minutes.



FINDING 8 + TAKEAWAYS

# If you serve gpt-oss-120b on H100

- **Turn on FP8 KV cache.** Up to 37% cheaper per token, no detectable accuracy cost here.
- **Pick by latency under load.** Time to first token at 16 users: SGLang 0.49 s, TensorRT-LLM 1.14 s, vLLM 1.54 s (finding 8).
- **Warm up at every batch size** before taking traffic, and watch for first-use compile stalls.
- **Need repeatable output?** SGLang deterministic mode, or vLLM batch-invariant mode with Triton attention — and budget 16–30% more per token.
- **Set output limits and timeouts deliberately** — they are part of the accuracy and latency budget.

ABOUT THIS WORK

# Built, run and analysed end to end

Benchmark harness in Rust, analysis in Python, profiling with Nsight Systems, deployment tested on Kubernetes (k3s + NVIDIA GPU Operator + Helm). Budget enforced by two independent watchdogs; all runs sealed and reproducible from raw data.

**Open source:** SGLang PR #52026 (open) — warn when the KV pool silently shortens a request's output, a cause of truncated answers found during this study.

Open to **GPU inference and ML infrastructure** roles, and contract work.

www.linkedin.com/in/isaaclevinsky/
mcmlv1.com
github.com/IsaacLevinsky

Engines: vLLM 0.33.0 · SGLang 0.5.23 · TensorRT-LLM 1.4.0rc2, measured November 2026 on 1× H100 SXM5 80GB. Limits: one GPU type, one model, synthetic speed load; accuracy differences under ~3 points are not detectable.