Running Gemma 4 Locally: Serving, Evaluation, and Gemini CLI Integration on a Single RTX 5090

Jayson Stone

Stoneforge Labs

Technical Report SFL-TR-2026-01 · July 27, 2026 · Version 1.0

Abstract

This report documents a local Gemma 4 inference and agent stack built on one NVIDIA GeForce RTX 5090 under WSL2. Three quantized operating points were measured with vLLM 0.25.1: a 26B-A4B FP8 checkpoint reached 180 output tokens per second at 32k context after cold-expert offload; a 26B-A4B NVFP4 checkpoint reached 189 tokens per second at 64k context; and a dense 31B NVFP4 checkpoint reached 63 tokens per second at 32k context. On MMLU-Pro with a five-shot chat-template protocol, the 26B FP8 checkpoint scored 82.86% and its NVFP4 counterpart scored 79.71%. A small Rust gateway connects the upstream Gemini CLI to the local vLLM OpenAI-compatible endpoint while preserving tool calls, structured responses, streaming, reasoning fields, and token counts. The report separates independently reproduced results from failed or confounded runs and records the engineering changes that made the stack reliable.

Keywords: Gemma 4, local inference, vLLM, MMLU-Pro, model evaluation, Gemini CLI, RTX 5090, open-source systems.

---

1. Scope

The goal was practical: run capable open-weight models locally, measure the configurations that fit a single workstation GPU, and use those models through an existing coding-agent interface. The work therefore covers both inference performance and the surrounding system required for repeatable use. It is not a general ranking of Gemma 4 checkpoints or serving engines.

Only results reproduced on the reported machine are treated as findings. Vendor figures are comparison context. Interrupted, truncated, or concurrently contaminated runs are excluded rather than repaired statistically.

2. System and model configurations

Component	Configuration
GPU	NVIDIA GeForce RTX 5090
Host environment	Windows Subsystem for Linux 2
Serving engine	vLLM 0.25.1, OpenAI-compatible HTTP endpoint
Fast profile	Gemma 4 26B-A4B instruction model, NVFP4 weights, FP8 KV cache, 64k context
Quality profile	Gemma 4 26B-A4B instruction model, dynamic FP8 weights, FP8 KV cache, 32k context, 64 cold experts offloaded
Smart profile	Gemma 4 31B dense instruction model, NVFP4 weights, FP8 KV cache, 32k context
Evaluation	EleutherAI lm-evaluation-harness; MMLU-Pro; five-shot; model chat template

The serving profiles are operating points, not claims that one quantization is best for every workload. Context length, CUDA-graph capture, KV-cache format, model density, and expert placement all affect the final behavior.

3. Method

3.1 Serving measurements

Decode throughput was measured with the same local serving benchmark against a quiet GPU. Reported values are batch-one output-token rates unless stated otherwise. Each retained row came from a configuration that booted cleanly and completed independently. Startup logs were retained to identify the selected kernel and to confirm model, context, cache, and graph-capture settings.

3.2 Accuracy measurements

MMLU-Pro was run through lm-evaluation-harness using five examples and the checkpoint chat template. The FP8 and NVFP4 26B-A4B measurements used the same protocol, so their 3.15-point difference is a within-system comparison. The published 82.6% Gemma 4 result is included only as external context; the local 82.86% result is not presented as evidence of a statistically meaningful improvement.

3.3 Evidence policy

A broad checkpoint sweep was rejected after inspection found three independent confounders: an 8k output truncation, a 7,200-second timeout, and interactive boots that displaced models under measurement. Those rows do not appear in the headline table. The retained numbers were rerun outside the compromised sweep.

---

4. Results

Configuration or measurement	Result	Evidence
26B-A4B FP8 with expert offload	**180 tok/s at 32k**	Serving benchmark, quiet GPU
26B-A4B NVFP4	**189 tok/s at 64k**	Same serving benchmark
31B dense NVFP4	**63 tok/s at 32k**	Same serving benchmark
MMLU-Pro, 26B-A4B FP8	**82.86%**	Five-shot, chat template
MMLU-Pro, 26B-A4B NVFP4	**79.71%**	Identical protocol
NVFP4 accuracy difference	−3.15 percentage points	Difference between preceding rows
Expert-offload change	97 → 180 tok/s	Before and after full CUDA-graph capture became possible
Repeated agent prompts	99.6% prefix-cache hit rate	Serving telemetry
NVFP4 execution path	`FlashInferCutlassNvFp4LinearKernel`	vLLM startup log

The local FP8 MMLU-Pro result closely matches the published Gemma 4 figure. The measured NVFP4 checkpoint provided the fastest decode rate and longest context of the three profiles, with a visible accuracy cost under the chosen evaluation. The dense 31B profile traded throughput for a different model architecture and was retained as a slower operating point rather than described as universally smarter.

5. Local Gemini CLI integration

Gemma Gateway is a roughly 1,300-line Rust compatibility bridge. It accepts the Gemini REST request shape used by the open-source Gemini CLI, translates it to vLLM's OpenAI-compatible interface, and maps responses back to the Gemini shape. The gateway covers text and image inputs, tool calls, JSON schemas, streaming, reasoning content, token accounting, and model discovery from the upstream `/models` endpoint. The public repository includes 31 translation tests and a live-model verifier.

```
Gemini CLI / gemma launcher
        |
        v
Gemma Gateway :8899  ----  local MCP tools
        |
        v
vLLM :8891  ----  Gemma 4 checkpoint

Workspace  ----  CodeGraph index
```

This arrangement keeps the upstream CLI intact. A local launcher selects the profile, starts vLLM, attaches user-supplied files or images, exposes status, and records logs. Workspace-side CodeGraph indexing can supply relevant source, call paths, and change blast radius before edits; it is external tooling and is not part of the model weights.

6. Engineering findings

6.1 Expert placement changed the usable performance envelope

Observed routing was not uniform: the hottest expert received 5.6 times as many selections as the coldest, while no expert was unused. Offloading the 64 coldest experts freed enough VRAM for full CUDA-graph capture. On this configuration, that gain outweighed the transfer cost and raised measured decode throughput from 97 to 180 tokens per second while the configured context doubled.

6.2 The tested TurboQuant KV-cache presets were incompatible

Four attempted TurboQuant presets failed when multimodal Gemma 4 selected `TRITON_ATTN`, which rejected the KV-cache data type. The profiles therefore use FP8 KV cache as the verified default. A boot failure is not counted as a performance result.

6.3 Reasoning output changes time-to-first-token accounting

Some responses emit their first token in `reasoning_content` rather than `content`. The measurement path was updated to count the first token from either field; otherwise, reasoning-enabled configurations appear to have

an artificially high time to first token.

6.4 Interactive boots and measurement sweeps require mutual exclusion

An interactive boot can evict the checkpoint currently being measured and create false serving failures. The launcher now refuses that conflict. This is a system-level validity control: a benchmark is only useful when the process under measurement remains the intended one.

---

7. Limitations

- The measurements come from one RTX 5090 workstation under WSL2; driver, kernel, thermal, and host differences may change results.
- Throughput values describe selected batch-one decode workloads, not maximum aggregate server throughput.
- The checkpoints come from different publishers and quantization pipelines. Weight format alone does not explain every observed difference.
- MMLU-Pro is one benchmark. Its score does not measure every capability needed for agent or coding work.
- The prefix-cache figure reflects repeated prompts in this local agent workload and should not be generalized to unrelated traffic.
- Excluded sweep rows are not available as valid comparisons. This report favors a smaller defensible result set over a larger contaminated matrix.

---

8. Artifacts and reproduction

- Gemma Rig project page and current commands
- Report source and version history
- Gemma Gateway source, tests, and live verifier
- 26B-A4B NVFP4 checkpoint
- 26B-A4B FP8 checkpoint
- 31B dense NVFP4 checkpoint

The repository history is the record of changes to this report. Reproduction should preserve the reported checkpoint, serving version, context, KV-cache format, graph-capture state, and evaluation protocol; omitting any of those can change the comparison.

References

1. Gemma Team. “Gemma 4 Technical Report.” arXiv:2607.02770, 2026. https://arxiv.org/abs/2607.02770.
2. Google DeepMind. “Gemma 4.” 2026. https://deepmind.google/models/gemma/gemma-4/.
3. Yubo 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.
4. EleutherAI. “Language Model Evaluation Harness.” https://github.com/EleutherAI/lm-evaluation-harness.
5. Woosuk Kwon et al. “Efficient Memory Management for Large Language Model Serving with PagedAttention.” arXiv:2309.06180, 2023. https://arxiv.org/abs/2309.06180.
6. vLLM Project. “OpenAI-Compatible Server.” https://docs.vllm.ai/en/latest/serving/online_serving/openai_compatible_server/.
7. Google. “Gemini CLI.” https://github.com/google-gemini/gemini-cli.
8. Jayson Stone. “Gemma Gateway.” Stoneforge Labs, 2026. https://github.com/Stoneforge-Labs/gemma-gateway.