Evaluating Qwopus3.6-27B v1-preview: a Reasoning Fine-tune of Qwen3.6-27B on Consumer Blackwell Silicon

Kyle Hessling

Independent · RTX 5090 workstation · @KyleHessling1

Working evaluation report · April 22, 2026

Abstract — We evaluate *Qwopus3.6-27B v1-preview*, an early-preview reasoning fine-tune of Qwen3.6-27B released by Jackrong, against its Qwen3.6-27B base model on a 16-prompt suite spanning agentic reasoning (5), production-grade front-end design (5), and creative canvas / WebGL coding (6). Inference was performed with `llama.cpp` on a single RTX 5090 (Blackwell, 32 GB), using Q4_K_M for Qwopus and Q5_K_XL for the base. We report a 12.7 % mean throughput improvement (55.3 → 62.3 tok/s), a > 10× collapse in per-run variance, and a reduction of "thinking-starvation" failures from 3/5 to 1/5 agentic prompts under matched `max_tokens`. Most of the tok/s delta is explained by the quant-size gap; the reasoning-trace discipline and variance reduction are genuine fine-tune effects. Results are preliminary, drawn from a ∼12 K-example training run; a larger, cleaner full fine-tune is in progress.

Note on scope — *this paper evaluates v1-preview, not the final Qwopus 3.6 model. The author is currently collaborating with Jackrong to secure additional compute for a full fine-tune run — orders of magnitude larger training set, cleaner data pipeline. Numbers and qualitative claims here should be read as a directional signal on the fine-tune approach, not the final artefact.*

1. Introduction

Open-weight reasoning fine-tunes of mid-size dense models have become a practical lever for local inference on consumer GPUs. Jackrong's Qwopus3.6-27B v1-preview represents an early-preview iteration of this pattern: a reasoning-oriented fine-tune of Qwen3.6-27B trained on approximately 12 000 curated examples sourced from four distillation and reasoning datasets. The stated goals of the fine-tune are (i) more structured chain-of-thought, (ii) reduced stylistic drift, and (iii) improved cross-source alignment.

This report answers a narrow, operational question: *for a practitioner running a 27 B-class model on a single-GPU consumer workstation, does the preview fine-tune change the production calculus versus the vanilla Qwen3.6-27B base?* We hold the evaluation suite, hardware, inference harness, and prompt phrasing constant across both models, and report quantitative throughput metrics and qualitative notes on output quality.

2. Experimental Setup

2.1 Hardware

All runs were performed on a single-GPU workstation: NVIDIA GeForce RTX 5090 (Blackwell, 32 GB VRAM, sm_120), Intel Core Ultra 7 265K, 128 GB system RAM, NVIDIA driver 580, CUDA 12.8.

2.2 Models and quantization

Two models were served, one per round:

- **Baseline:** `unsloth/Qwen3.6-27B-GGUF` — `UD-Q5_K_XL` (≈ 19 GB).
- **Fine-tune (this work):** `Jackrong/Qwopus3.6-27B-v1-preview-GGUF` — `Q4_K_M` (≈ 16 GB).

The quantization mismatch is a real confound and is discussed in §4. At the time of writing, Jackrong has not published a Q5 quantization of Qwopus, and a re-quantization pass on both models would be required for a fully-matched comparison.

2.3 Inference runtime

Both models were served with the same `llama.cpp` CUDA-12.8 build, `--flash-attn on`, `--jinja`, context 65 536, `q8_0` K and V cache, single inference slot. Earlier attempts to use vLLM or SGLang with Multi-Token Prediction (MTP) speculative decoding failed: FP8 weights did not leave room for the KV cache on a 32 GB card; AWQ-INT4 used a group size incompatible with every available Blackwell kernel; NVFP4 required CUDA ≥ 12.9. Those results are documented in the companion base-model evaluation and not re-litigated here.

2.4 Prompt suite

The harness issues 16 prompts in three categories:

- **Agentic (5):** multi-step planning, tool-use JSON, code debugging with multiple bugs, structured JSON extraction, and self-critique of a palindrome-finding function. Thinking mode was enabled.
- **Web design (5):** production-quality HTML/CSS pages — SaaS landing, analytics dashboard, designer portfolio, pricing page, mobile-app marketing. Thinking mode was disabled to preserve the `max_tokens` budget for output.
- **Canvas / WebGL (6):** particle attractor, generative flow field, WebGL raymarched Mandelbulb, three.js transmissive-glass scene, 2D physics sandbox, audio-reactive visualizer. Thinking mode disabled.

All prompts, the harness, and the outputs are preserved at `huggingface.co/spaces/KyleHessling1/qwopus36-eval`.

3. Results

3.1 Aggregate throughput

Table 1 summarises throughput across the full 16-run suite.

Metric	Qwen3.6-27B base · Q5_K_XL	Qwopus v1-preview · Q4_K_M	Δ
Mean gen tok/s	55.3	62.3	+12.7 %
Min / max tok/s	51.3 / 57.0	61.8 / 62.7	variance ↓ 10×
Total completion tokens	93 899	87 394	−6.9 %
Total wall-clock gen time	28.0 min	23.4 min	−16 %
VRAM resident (peak)	24.5 GB	≈ 20 GB	−4.5 GB

Table 1. *Aggregate throughput across 16 runs per model. The Qwopus column uses a lighter quant (Q4 vs Q5); the bandwidth delta alone accounts for most of the tok/s improvement (see §4.1).*

3.2 Agentic reasoning

A recurring failure mode of Qwen3.6-27B with thinking mode enabled is *token-budget starvation*: the model consumes the entire `max_tokens` allowance inside `<think>`, leaving an empty final `content` channel. Table 2 compares the two models on the five agentic prompts under matched budgets.

Task	Qwen3.6-27B base	Qwopus v1-preview
Multi-step deployment plan	Pass (3 802 reasoning tok)	Pass (3 158 reasoning tok)
Tool-use JSON	Empty — needed nothink	Pass (1 174 tok)
Code debug (4 bugs)	Empty — needed nothink	Pass (1 628 tok)
Structured JSON extraction	Empty — needed nothink	Empty — needed nothink
Self-critique ($O(n^3) \to O(n^2)$)	Pass (2 837 tok)	Pass (1 277 tok)

Table 2. *Agentic-prompt starvation behaviour under matched `max_tokens` with thinking mode enabled. Qwopus converts three "empty" outcomes to "pass" through shorter, more disciplined reasoning traces.*

In qualitative inspection, both models caught every bug in the code-debug task (sort order, `=`-vs-`==`, bounds logic, `nums[k]` vs `nums[k-1]`) and produced a correct self-critique progression from a naïve $O(n^3)$ palindrome scan to an expand-around-center $O(n^2)$ variant. The one still-starving task in Qwopus (structured extraction) required ∼ 1 500 reasoning tokens before producing content; lifting `max_tokens` to 6 K or disabling thinking resolved it in both models.

3.3 Front-end design

All ten design outputs (five per model) passed a basic structural audit: each begins with `<!DOCTYPE html>`, terminates cleanly with `</html>`, and parses as valid HTML. Table 3 summarises output sizes.

Prompt	Qwen base (KB · tok)	Qwopus (KB · tok)
SaaS landing	35.8 · 9 991	36.7 · 9 961
Analytics dashboard	40.8 · 12 733	37.4 · 13 190
Designer portfolio	20.9 · 5 387	23.1 · 7 356
Pricing page	29.2 · 7 849	24.3 · 8 061
Mobile app marketing	32.4 · 9 243	29.3 · 8 005

Table 3. *Per-prompt output size for the five design tasks. Qwopus collapses the spread from 20.9–40.8 KB to 23.1–37.4 KB, producing more uniform page sizes without either truncation or padding.*

Both models handle the brief consistently: Inter + JetBrains Mono on the SaaS page, actual hand-coded SVG charts on the analytics dashboard (not placeholder rectangles), a magnetic cursor-following CTA on the portfolio, a rotating conic-gradient border on the recommended pricing tier, and a CSS-only iPhone mockup with a 4-7-8 breathing animation on the Stillwater marketing page. No regressions were observed on design quality between the base and the fine-tune.

3.4 Canvas and WebGL

Qwopus produces tighter canvas output on average (mean 13.5 KB vs 17.6 KB for base). All submitted demos include the required animation loop, resize handling, and interaction bindings (pointer, keyboard, audio). The raymarched Mandelbulb compiled cleanly on first inspection; the base-model version required a one-line GLSL type-promotion patch to run (documented in the base evaluation).

4. Discussion

4.1 Separating quant effect from fine-tune effect

The headline +12.7 % tok/s is primarily a memory-bandwidth effect. Q4_K_M weights (≈ 16 GB) move ∼ 16 % less data per decode step than Q5_K_XL (≈ 19 GB), which closely matches the measured speedup on an otherwise bandwidth-bound single-stream decode. A same-quant comparison would be needed to isolate any fine-tune contribution to raw throughput. Based on the Qwen base-model's prior behaviour on the same harness, we expect same-quant difference to be under 2 %.

4.2 Variance collapse is not a quant effect

More interesting is the 10× reduction in per-run throughput variance (base: 51.3–57.0 tok/s; Qwopus: 61.8–62.7 tok/s). Quant choice does not predict this — bandwidth is roughly constant per-decode. The plausible mechanism is that Qwopus's fine-tune produces a more consistent logit distribution, reducing sampling-stage variability that the base model exhibits across different prompt types. Confirming this would require token-level latency tracing per prompt, which is beyond the scope of this report.

4.3 Reasoning discipline as the real win

The three-of-five → one-of-five starvation reduction is the most operationally meaningful result. On the base model, three common agentic prompts (structured JSON, code review, tool-use) required a second pass with thinking mode disabled to produce content. On Qwopus, only one did. Reasoning traces are approximately 30 %

shorter at equivalent answer quality, consistent with the stated fine-tuning objective of "more structured chain-of-thought." For callers with tight `max_tokens` budgets, this directly translates to fewer empty-content retries and lower observed latency variance.

4.4 Date-math regression

One qualitative regression was noted in the structured-extraction task with thinking disabled: "next Tuesday" from 2025-04-21 was resolved to 2025-04-28 (a Monday) by Qwopus nothink, to 2025-04-29 (a valid Tuesday interpretation) by the base, and to 2025-04-22 (the arguably-correct "immediately upcoming Tuesday" interpretation) by Qwopus with thinking enabled. The thinking-on trace contained correct calendar reasoning but was starved before emitting the final JSON. This is consistent with a pattern where Qwopus's fine-tune moves reasoning depth into the `<think>` channel at the expense of shallow nothink inference.

5. Limitations

- **Quant mismatch.** Q4 vs Q5 confounds the throughput comparison. A matched-quant rerun is pending on Jackrong's side.
- **Single-stream, single-GPU.** All measurements are `--parallel 1`, sm_120, 32 GB. Multi-stream, multi-GPU, or non-Blackwell results may differ.
- **Single seed, single run per prompt.** No inter-run variance characterisation; reported numbers are point estimates.
- **Evaluation is qualitative on design / canvas.** Output was structurally validated and spot-checked in a browser but not programmatically graded against a rubric.
- **Preview status.** v1-preview is explicitly not the final model. The full fine-tune in progress is sized orders of magnitude larger; these numbers will shift.

6. Qualitative Notes on Design Polish and Creativity

Beyond the structural audit of §3.3, the author manually inspected all rendered UI outputs side-by-side in a browser. Two qualitative observations inform the conclusion and are reported here as informal impressions, not as the result of a controlled rubric:

Polish. On the standard design briefs — SaaS landing, pricing tier, analytics dashboard — the Qwen3.6-27B base outputs feel slightly more polished than Qwopus's. The base lands closer to "near-perfect" on conventional UI patterns, while Qwopus ships clean, functional work that is typically a half-step behind on the finest visual details. This gap is consistent with a preview trained on ∼ 12 K examples and is the kind of deficit that typically closes with scaled training data.

Creativity. On open-ended prompts, Qwopus occasionally exceeds the base in creative range at the cost of the last ten percent of polish. The clearest single example is the audio-reactive visualizer prompt: Qwopus produced a structurally distinct interpretation — different rendering approach, different visual language, different micro-interaction vocabulary — compared to the base model's more conventional concentric-ring solution. The Qwopus version shows marginally less refinement in certain corners but is substantially more original in the whole. This trade — a shade less finish in exchange for more distinct creative swings — is a reasonable profile for an early

reasoning-focused fine-tune and is aligned with the stated training objective of reducing stylistic drift across distilled sources.

7. Conclusion

On the studied workloads, Qwopus3.6-27B v1-preview is a practical improvement over the Qwen3.6-27B base on a single 32 GB Blackwell GPU. The throughput gain is largely a quant-size effect; the variance collapse and the reduction in thinking-starvation failures are genuine fine-tune contributions. Qualitatively, the base retains a slight edge in polish on standard UI briefs while the preview fine-tune shows more creative range on open-ended prompts — a polish-for-creativity trade we expect to narrow as training data scales. For local inference of 27 B-class reasoning models on consumer hardware, the preview is a defensible drop-in replacement today, with room for larger gains when the full-scale training run lands.

8. Acknowledgements

Thanks to Jackrong for training and releasing the Qwopus3.6-27B v1-preview checkpoints and for collaborating on the forthcoming full fine-tune run. Thanks to the Unsloth team for the Qwen3.6-27B GGUF conversions used as the baseline model in this study, and to the llama.cpp maintainers for timely Blackwell/sm_120 kernel support.

9. References

1. Jackrong. *Qwopus3.6-27B-v1-preview-GGUF.* Hugging Face model repository, 2026. https://huggingface.co/Jackrong/Qwopus3.6-27B-v1-preview-GGUF
2. Qwen Team. *Qwen3.6-27B.* Hugging Face model repository, 2026. https://huggingface.co/Qwen/Qwen3.6-27B
3. Unsloth. *Qwen3.6-27B-GGUF (UD-Q5_K_XL).* Hugging Face model repository, 2026. https://huggingface.co/unsloth/Qwen3.6-27B-GGUF
4. Hessling, K. *Qwen3.6-27B base-model evaluation.* Hugging Face Space, 2026. https://huggingface.co/spaces/KyleHessling1/qwen36-eval
5. Hessling, K. *Qwopus3.6-27B v1-preview evaluation (this work — raw outputs and harness).* Hugging Face Space, 2026. https://huggingface.co/spaces/KyleHessling1/qwopus36-eval
6. llama.cpp contributors. *llama.cpp: inference of LLaMA and compatible architectures in C/C++*. 2023–2026. https://github.com/ggerganov/llama.cpp

---

Preprint v1 · prepared for the Qwopus3.6 evaluation Space · source HTML and raw model outputs available under the companion repository cited above. Not peer-reviewed; this is an operational evaluation, not a controlled study.