Backend wars, Mac math, and the back-catalog refresh

May 25, 2026 · by Mark Bartlett

Download this post as PDF

InsiderLLM Weekly issue 10 – May 25, 2026

The Qwen 3.6 ecosystem stopped being new this week and started being mapped. Three backends benched head to head on a single RTX 3090. The VRAM calculator finally got Qwen 3.5 and Qwen 3.6 added — with a real architectural gotcha worth knowing. And the back-catalog refresh started, which is the polite way of saying I found a lot of our own guides still recommending Qwen 2.5 when Qwen 3.6 was the right pick. We're fixing it.

Backend Wars on Qwen 3.6 27B

Three speculative-decoding backends. Same model (Qwen 3.6 27B at Q4-ish quant per each backend's canonical config). Same RTX 3090. The am17an 9-prompt harness, run head to head.

Headline numbers from the firsthand bench (May 22):

- **Mainline llama.cpp** — 37.22s total wall time, baseline
- **ik_llama.cpp + MTP** — 1.66x speedup, 88.5% acceptance rate
- **BeeLlama + DFlash** — 1.62x speedup, 37.4% acceptance rate, peaks at 119 tok/s on code_python

The editorial hook hiding inside those numbers: same wall-clock, opposite strategies. ik_llama is conservative — accept the draft most of the time, keep the average steady. BeeLlama is aggressive — accept less often, but burst higher when it does. That's the acceptance-rate trap I see people fall into all the time: high acceptance doesn't mean high speed. They're measuring different things.

ik_llama wins translation tasks 2x over BeeLlama. BeeLlama wins code_python by a similar margin. Different backends, different shaped workloads. The answer to "which one is fastest" is "for what."

Update May 25: BeeLlama v0.2.0 dropped Saturday with 100+ commits, 86% of them DFlash hardening — invalid-draft fallback, accepted-prefix KV clamping, prefill GPU staging fixes, the "fail-closed on unsafe state" kind of work that doesn't add features but makes the feature

actually trustworthy. Plus initial Gemma 4 iSWA support as a side thread. I re-benched it this morning on the same harness, and the result is genuinely mixed: code-heavy peak improved (+8% on code_python, hitting 128.5 t/s), but acceptance rate dropped 6.65 percentage points, total wall-clock slightly regressed (+3.6%), and the 164 t/s community claim circulating on Reddit doesn't replicate against the am17an harness. The hardening is real stability work; the throughput story is workload-dependent. Code wins, translation loses, average is flat-to-slightly-worse. v0.3.0 has appeared as a branch on origin — different release, different bench. Updated article with the full comparison tables.

📖 **Full backend shootout with all 9 prompts, methodology, and the v0.2.0 update is here.**

Three Things Worth Pulling This Week

Qwen3-Coder-Next on Ollama. Quietly the canonical agentic coder right now — 80B total / 3B active MoE, 256K context, 1.3M downloads, hybrid attention. Built on Qwen3-Next-80B-A3B-Base. At Q4 it's 52GB, so you need 64GB unified memory or multi-GPU. On Mac Studio 96GB+ it's the most capable Qwen coder you can run locally. `ollama pull qwen3-coder-next`.

hipEngine for RDNA3. Native Qwen 3.6 inference on Strix Halo and 7900 XTX. AMD finally getting first-class day-zero Qwen support, not the usual 3-week-after-the-fact backport. Reddit thread blew up over the weekend. If you have RDNA3 hardware, this is the path forward.

NuExtract3. Open-weight 4B vision-language model for Markdown extraction, OCR, and structured output. Self-hostable. 4B size means it runs on basically anything. The niche local AI hasn't had a great answer for has been "structured extraction from messy documents" — this is that answer.

Still Waiting On Qwen 3.7

The Max preview hit AAI 57 on May 20 (#1 of 218 models tracked). The open weights still haven't dropped. Based on the 3.6 release pattern, June 2026 is the realistic window. Alibaba's been "stay tuned" for 12 days now. The 3.7 preview piece is updated when there's news to update; right now there isn't.

📖 **Current Qwen 3.7 watch with AAI numbers and timing analysis is here.**

The Calculator Caught Up — And We Found Some Stale Guides Of Our Own

The VRAM calculator update added Qwen 3.5, Qwen 3.6, Gemma 4, Llama 4 Scout, and Qwen3-Coder-Next. While building the new entries, the real gotcha surfaced: Qwen 3.5 and Qwen 3.6 use hybrid attention. Every 4th layer is full attention, the rest are linear (Gated DeltaNet) with no KV cache. For Qwen 3.6 27B that's 16 attention-bearing layers out of 64 total. Any VRAM calculator or rule-of-thumb that assumes every layer has KV cache will overestimate by roughly 4x at long context. Our calculator now uses the right math.

Other things shipped while we were in there: sync between the model dropdown and the JavaScript model array (a bug since launch — 5 models were in the dropdown but missing from the calculation backend). Notes for MTP overhead, sliding-window attention (Gemma 4), and the Llama 4 Scout "10M context is theoretical max" caveat.

Then I went looking through the rest of the catalog and found 120 articles still mentioning Qwen 2.5. Triage came back: 16 of them are fine (historical references), 38 need light comparison-table updates, 39 actively recommend Qwen 2.5 as the current pick in May 2026, and 22 need full refreshes. The two highest-traffic offenders shipped fixes yesterday — best-local-llms-mac-2026 (8,168 humans/month, now showing Qwen 3.6 and Llama 4 Scout) and qwen-models-guide (2,857 humans/month, now correctly positioning Qwen 3.6 as the frontier).

The underlying story: prior freshness passes had been updating the top of articles (description, quick answer) without touching the body recommendations. So readers were getting Qwen 3.6 in the lede and Qwen 2.5 in the actual pick paragraphs. That's a real bug in our own freshness methodology, and now I know to check for it.

📖 **Updated VRAM calculator is here. Updated Mac guide here. Updated Qwen models guide here.**

What We Shipped This Week

Monday May 19 through Sunday May 25:

- **New:** Backend shootout — ik_llama vs BeeLlama vs mainline llama.cpp on RTX 3090 (May 22, firsthand bench)
- **Updated:** VRAM calculator — Qwen 3.5/3.6, Gemma 4, Llama 4 Scout, Qwen3-Coder-Next, hybrid attention math (May 24)
- **Refreshed:** best-local-llms-mac-2026 — M5 era, Llama 4 Scout for 96GB+ Macs (May 24)

- **Refreshed:** qwen-models-guide — Qwen 3.6 promoted to frontier, coder section rewritten (May 24)
- **Updated:** Backend shootout — BeeLlama v0.2.0 mixed results (May 25)
- Earlier in the week: used-gpu-buying-guide rewrite kickoff, Llama 4 guide refresh, Hugo byline policy ship

Six pieces. The back-catalog work is the biggest cadence shift — going from one-off refreshes to systematic cleanup of an entire 200+ article archive.

Next issue lands June 1. The Qwen 2.5 → Qwen 3.6 catalog refresh continues at three articles per day (the sustainable cadence). I'll have v0.3.0 BeeLlama numbers if the new branch ships before then. Phase 1 of the cross-vendor used GPU guide is still in progress.

— Mark, InsiderLLM

Running local AI on weird hardware? Built something novel with it? Drop us a line at hello@insiderllm.com.
