A 27B for your 12 GB card. It lost one field.

September 28, 2026 · by Mark Bartlett

Download this post as PDF

InsiderLLM Weekly issue 23 – September 28, 2026

A 27B now fits a 12 GB card and decodes 1.8x faster than the Q4 it was squeezed from. It also lost one field on my routing test — the same way every time. That's the main piece. First, a reader caught a stale 3090 price, and it wasn't the only one.

New This Week

- **Bonsai 2 vs Qwen3.8-27B on an RTX 3090: Speed Doubled, Scope Broke.** Bonsai 2 27B, PrismML's ternary Qwen3.8, on a 3090: 1.8x the decode of Q4 from a 7.2 GB file, then 12 of 47 against 19 on a routing split. 📖 Read it.
- **What Is Jev, and Can You Run One on Your Own GPU?** Jev picks from options you supply instead of writing, in one pass, with a probability per answer. 📖 Read it.

Updated:

- **23 articles corrected for used GPU prices** after a reader caught a stale 3090 figure. Each carries a dated correction line saying what it used to say.
- **Best Used GPUs for Local AI: the fair-price ladder.** It had been calling fair 3060 listings overpriced. Against 107 sold listings the median is $300, so fair is now $275-325, a great price is under $250, and overpriced starts above $350. 📖 The guide.
- **Best Way to Run Qwen 3.6 35B MoE Locally.** "A separate drafter doesn't help" was a 3060 offload result. On a resident 3090, z-lab's DFlash drafter gave 1.09x overall, from 1.64x on Python down to 0.41x on a short translation. 📖 The guide.
    - Correction, 2026-09-28: the "3060 offload result" was not ours. The 11 tok/s behind it was Codacus's GTX 1060 6GB measurement, mis-attributed to our 3060 in a July edit. Our own 3060 figures: the 0.8B drafter at 0.42x; the DFlash drafter at `-ncmoe 26`, the lowest offload it fits on 12 GB, at 0.58x of the plain `-ncmoe 24` config; Python code is the only prompt that wins (1.06x). The cost is CPU-side verification of 16-token batches against the experts in system RAM, not PCIe traffic. The record.

Quick Hits

- **Qwen Image 2.1.** 7B generator under the Qwen Research License, not Apache. gguf-org/qwen-image-2.1-gguf landed next day: NVFP4 transformer, Q4 text encoder, VAE, 10 GB, so 12 GB cards fit. 📖 The GGUF · the model.
- **Qwen Audio 3.1.** Five audio models, API prices cut up to 95 percent. No open weights on Hugging Face, GitHub or Qwen's post. Local voice stays June's Qwen3-ASR-1.7B. 📖 The Decoder · the local option.
- **llama.cpp v0.5.0.** Twelve builds and a version tag: CUDA conv2d speedup, Metal MoE fusion, MiMo-V2.6 support, no breaking changes. We stay on v0.4.0 until it's re-gated on both rigs. 📖 Release notes · our dataset.
- **Opus 5.5 and GPT-6.** Both closed, both cheaper: Opus 5.5 with stricter cyber safeguards, GPT-6 pitched on cost and fewer mistakes. The API column of our local-vs-API break-even moved. 📖 The Verge · OpenAI · our break-even.
- **Buried injections.** 629 AgentDojo injections buried in tool output: regex catches 0 percent, Prompt Guard 2 catches 1 percent out of the box, 99 percent once its threshold tuned. 📖 The test set · ours.
- **Dettmers's 1.5-bit claim.** Qwen 3.6 35B-A3B at 450 tok/s at 1.5 bits per weight on Metal. His lab's first release is a private beta of bitsandbytes2, so there's still nothing public to download; when the code ships, we'll run it on a 3090. 📖 The post · the beta.
- **Open Jev-style decision models.** Jev-Style-Qwen3.5-2B picks among 2 to 26 options with a probability each, Apache 2.0, on plain llama.cpp. Take the 2 GB Q8: the Q4 agrees with full precision on only 91 percent of choices. Ollaya serves the same class on ONNX Runtime. 📖 The GGUF · Ollaya · our explainer.
- **2.2x from one setting on an Intel Arc iGPU.** Qwen3.6-35B on a laptop's integrated Arc B390 went from 14.7 to 32.5 tok/s by moving the MoE experts off the CPU. An old README priced that at 10 percent — it cost 55 percent of decode. If your model fits, check you aren't offloading experts by habit. 📖 The post · our 35B guide.

A 27B on a 12 GB Card, Minus One Field

If you have a 12 GB card and want a 27B for chat, try Bonsai 2 27B. It's PrismML's ternary Qwen3.8-27B — every weight stored as −1, 0 or +1 — and the whole model lands in a 7.2 GB file. On my 3090 it peaked at 8,118 MiB with 8k of context, which leaves a 12 GB card room to spare. It decodes at 77 tok/s against 42 for the Q4 it came from, and still holds 1.8x at 8k deep. PrismML's table claims parity on code; I haven't measured that. An 8 GB card will need a short context, and I haven't measured how short either.

Then the bill. I ran both through the 47-item routing split this site has scored since August. The Q4 got 19 right on all three fields. Bonsai got 12, and the whole gap is one field. On tool, they tied at 28. On mode, Bonsai won, 27 to 24. On scope — the field that asks whether a message points back at something said earlier — it fell from 28 to 18, and 23 of its wrong scopes were the same call: `facts` where the answer was `session` or `all`.

PrismML says it keeps 98.2 percent of the full model's benchmark score. Both numbers can stand. Theirs is a 14-benchmark average in thinking mode against FP16. Mine is one task, thinking off, against the Q4 a consumer card can hold. An average like theirs can't show you one field breaking.

So if you route on it, measure first. With a 24 GB card and no VRAM pressure, the Q4 is still the safer file.

📖 **We wrote a full breakdown here.**

That's the week. If you have a 12 GB card, Bonsai 2 is worth a download for chat — just don't hand it your router without scoring it on your own data first. And if you spot a price on this site that looks wrong, tell me. The last one led to corrections on 24 pages.

New here, reading this on the web? Subscribe and the next one lands in your inbox.

— Mark, InsiderLLM

Tried Bonsai 2 on a 12 GB or 8 GB card? Got a context length that fits? Reply, or hit me at hello@insiderllm.com. I read everything.

A weekly email with every new guide and measured benchmark.

Subscribe — free, no spam
