DeepSeek V4 Flash vs Pro: What Actually Dropped and How to Run It

April 24, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** DeepSeek V4 preview shipped the evening of April 23, 2026 as two MoE checkpoints: V4-Pro at 1.6T total / 49B active, and V4-Flash at 284B total / 13B active. Both are MIT-licensed with a 1M-token context window. Pricing per DeepSeek's own API: Flash is $0.14 in / $0.28 out per million tokens, Pro is $1.74 in / $3.48 out. Flash is the practical story — frontier-adjacent capability at Haiku-tier pricing, open weights. Pro is impressive but it's workstation and server hardware, not a laptop. Independent benchmarks will land over the next week or two; current claims are DeepSeek's own plus community anecdote.

Related: DeepSeek V4 Preview (what we knew before) · DeepSeek V3.2 Guide · VRAM Requirements · llama.cpp vs Ollama vs vLLM · Qwen 3.5 Setup Guide · Run 31B Models on a Laptop

Contents

DeepSeek V4 preview went live the evening of April 23, 2026. Two MoE checkpoints, both MIT, both 1M context. r/LocalLLaMA has been in steady eruption since, Hacker News has multiple front-page threads, and Simon Willison has his pelican-on-a-bicycle post up. This is the time-sensitive read — here's what's real, what's claimed, and what's still waiting on independent testing.

Image: DeepSeek V4 Pro and Flash logos side by side with parameter counts

What actually dropped

Four checkpoints on the Hub under `deepseek-ai`:

- **DeepSeek-V4-Pro** — 1.6T total, 49B active, instruct
- **DeepSeek-V4-Pro-Base** — same params, base (no instruction tuning)
- **DeepSeek-V4-Flash** — 284B total, 13B active, instruct
- **DeepSeek-V4-Flash-Base** — same params, base

Instruct checkpoints ship as FP4 for MoE experts and FP8 for everything else. Base checkpoints are FP8 throughout. Both sizes carry a 1M-token context window. License is MIT on every file. That last point matters — no "community license" nonsense, no usage restriction clauses. Download, run, ship, done.

Architecture is where DeepSeek earned the headlines. Per DeepSeek's V4 technical report and the Hugging Face blog post, V4 keeps the MoE direction but replaces V3.2's attention stack with a hybrid of Compressed Sparse Attention and Heavily Compressed Attention, alternating across 61 layers. KV cache is stored in FP8 with BF16 RoPE dimensions.

The efficiency numbers DeepSeek claims for this are substantial:

- **V4-Pro**: 27% of single-token FLOPs, 10% of KV cache vs V3.2 at 1M tokens
- **V4-Flash**: 10% of FLOPs, 7% of KV cache vs V3.2

If these hold up under independent testing, million-token context stops being a theoretical feature and starts being something you can actually serve. We'll see.

Three reasoning modes ship on both models: Non-think (fast, no chain of thought), Think High (explicit `<think>` blocks), and Think Max (maximum reasoning, requires 384K+ context window allocated). Recommended sampling is temperature 1.0, top-p 1.0.

V4-Flash vs V4-Pro: the real tradeoff

Spec	V4-Flash	V4-Pro
Total params	284B	1.6T
Active params	13B	49B
Context window	1M tokens	1M tokens

Spec	V4-Flash	V4-Pro
License	MIT	MIT
Weights format	FP4 (MoE) + FP8	FP4 (MoE) + FP8
FLOPs vs V3.2 @ 1M	~10%	~27%
KV cache vs V3.2 @ 1M	~7%	~10%
API input (cache miss)	$0.14/M	$1.74/M
API input (cache hit)	$0.028/M	$0.145/M
API output	$0.28/M	$3.48/M
Reasoning modes	Non-think, Think High, Think Max	same
Best for	Agents, tool calling, high-volume	Research, max-quality long context

All pricing per DeepSeek's official API pricing page as of April 24, 2026.

The gap between these two isn't what you'd expect from "small vs large." Flash is a 12x cheaper per output token. It's also ~5.6x smaller on disk. And per DeepSeek's technical report, Flash approaches Pro on reasoning-heavy tasks when Think High or Think Max is enabled. That claim deserves real independent verification before anyone bets a pipeline on it — but if it holds even partially, Flash is the model most teams should be testing first.

Pro is what it sounds like: the frontier-adjacent model. DeepSeek's own numbers put V4-Pro-Base at 90.1% MMLU, 76.8% HumanEval, 92.6% GSM8K. On agent benchmarks V4-Pro-Max scores 80.6 on SWE-bench Verified (roughly parity with Claude Opus 4.6) and 67.9 on Terminal Bench 2.0 (behind GPT-5.4-xHigh at 75.1). Those are DeepSeek's own reported numbers from the tech report. They aren't independent evaluations.

Can you actually run this locally?

Honest answer: one of these, maybe. The other, no.

V4-Pro. 1.6T total params. Even at Q4, you're looking at ~800GB just for weights, plus KV cache for however much context you load, plus activation memory. This is not a homelab story. It's a workstation with a terabyte of fast RAM, a high-end GPU for hybrid offload, and patience. An 8-GPU H100 or H200 box can serve it comfortably. A pair of 5090s and a ThreadRipper cannot. No independent benchmarks on home hardware yet — if you see a tok/s number for V4-Pro on consumer gear in the next week, treat it skeptically until someone reproduces it.

V4-Flash. 284B total, 13B active. At FP4 weights plus FP8 everywhere else, the checkpoint is in the ~150GB range. That's still not laptop territory, but it's in range for serious homelabs: two RTX 6000 Ada at 48GB each, or a Mac Studio M3 Ultra 512GB with unified memory, or a Threadripper with enough DDR5 channels to feed llama.cpp-style hybrid offload. Community reports in the first 24 hours suggest it runs — we don't have reproducible tok/s numbers from independent testers yet. Flash at 13B active is structurally the same scale of "hot path" as Mixtral 8x7B or GLM-4.5 — if you have a rig that runs those, Flash is at least in the conversation.

For comparison: a 1T-class model like Kimi K2 has previously required similar hardware to V4-Pro — dedicated server or deep-pocket workstation. V4-Pro isn't materially harder than that class. V4-Flash is meaningfully easier to run than V4-Pro, but still harder than a 70B dense model at Q4.

Quantization status as of April 24. Unsloth has not yet released Dynamic 2.0 GGUFs for V4. No community GGUF at the time of writing. vLLM supports the native FP4/FP8 checkpoints out of the box — see the vLLM DeepSeek V4 blog post for deployment notes. Expect `ubergarm`, Unsloth, and bartowski GGUFs over the next few days. If you're planning a local deployment and you need llama.cpp compatibility, the honest answer is wait a few days.

Image: Diagram showing V4-Flash and V4-Pro hardware fit across laptop, homelab, workstation, and server tiers

---

Early community reports

Everything in this section is community anecdote from the first 24 hours. Treat accordingly.

Vibe Code Benchmark. Vals AI reported V4 "overwhelmingly" topped open-source models on the Vibe Code Benchmark, with a roughly 10x jump from V3.2. That's a Vals AI claim, not an independent re-run — useful signal, not a settled number. Expect the LM Arena and Aider Polyglot boards to update over the next few days.

Flash as a Haiku / Gemma replacement. Multiple r/LocalLLaMA threads note Flash is fast enough and cheap enough via the API to stand in for Claude Haiku or Gemma 4 in tool-calling pipelines. One benefit people keep pointing out: Flash's tool-call schema uses interleaved thinking that survives across tool boundaries, which matters if you're running multi-step agents. This is the claim to test first if you're running agent workloads on closed APIs today.

Ollama cloud availability. `deepseek-v4-flash:cloud` went live on Ollama's cloud catalog within hours of release. r/ollama threads on April 24 report intermittent timeouts on long

requests — possibly just load from the release spike. Local Ollama pulls for V4 weights are not available yet; the cloud tag proxies through Ollama's hosted inference.

Simon Willison's pelican test. Simon's post has the pelican-on-a-bicycle output for both models. Short version: Flash drew a better bicycle than Pro did, and Pro's pelican came out with "a VERY large body, only one wing" (Simon's words). He positions V4-Pro as roughly 3 to 6 months behind the American frontier labs while costing a fraction of their API rates — a fair take at this stage, though one weekend's results aren't a verdict.

Where to use which

V4-Flash. Pick this when:

- You're running agents or tool-calling pipelines and currently paying Anthropic or OpenAI per token
- You need long-context document work (contracts, codebases, transcripts) and don't want to chunk
- You were running Qwen 3.5 27B or Gemma 4 for cost reasons and want to try a bigger active-param footprint
- You want MIT-licensed open weights as a hedge against closed-API policy changes

V4-Pro. Pick this when:

- You're running research-grade reasoning tasks and can afford $1.74 / $3.48 per million tokens
- You have a 1M-token use case that actually needs maximum quality — multi-document analysis, very long code review, full-book translation
- You already run Kimi K2 or similar 1T-class models and have the hardware to host them
- Budget isn't the constraint; model quality is

Either. Pick V4 at all when you want open weights under a permissive license and don't want your production workload depending on closed-API pricing or availability decisions.

How to try it today

DeepSeek's own API. OpenAI-compatible, cheapest direct path. See the DeepSeek API docs. Register, get a key, switch your base URL. Works as a drop-in for most OpenAI SDK code.

Vercel AI Gateway. Hacker News commenters on the V4 release thread noted Vercel's gateway has the cheapest effective rate for V4 via prompt caching — their cache layer survives longer than DeepSeek's native cache window. Verify against your own traffic before switching.

Ollama cloud. `ollama run deepseek-v4-flash:cloud` if you have cloud access set up. Currently seeing some stability reports on long runs — fine for testing, check your error rates before putting it in front of users.

vLLM self-hosted. The vLLM recipe for V4-Flash is published. Native FP4/FP8 serving. You need the hardware described in the hardware section above — this is not a weekend project on a single 3090.

llama.cpp + GGUF. Wait. No production-ready GGUFs yet. Unsloth historically ships V3/R1 quants within a week of release, expect similar timing here.

Bottom line

V4-Flash is the actual news. Frontier-adjacent capability, MIT license, Haiku-tier API pricing, and architecture that's legitimately easier to serve at 1M context than anything in its class. If you're running agentic workloads on Claude or GPT today, test Flash this week. The pricing alone justifies a half-day of effort.

V4-Pro is impressive but it's not a consumer-hardware story. Research groups, enterprise teams, and well-funded homelabs only. If you're running Kimi K2 locally, Pro is in the same league. If you're running a 3090, it isn't.

Independent benchmarks will land over the next one to two weeks. The numbers we have now are DeepSeek's own plus community first-impressions. Good enough to act on, not good enough to treat as settled. Pull Flash, try your actual workload against it, and form your own read. That's always the rule with a release this fresh.

Last updated April 24, 2026 — the day after the preview landed. This page will be refreshed as independent benchmarks, Unsloth GGUFs, and longer-running community reports come in.

Free guides for running AI locally