04: Open weight vs closed weight

What you get when you can download the model

Why this matters

- Weights are the model. Closed labs keep them; open labs let you download them.
- Open models trail the closed frontier by months
- Open wins on control, privacy, cost at scale, and customization
- Closed wins on peak capability and convenience

Where the weights came from isn't where your data goes.

Part 1

What "open" means

Weights are just big files

Format	What it is
`.safetensors`	Standard on Hugging Face; holds only tensors, so loading can't run hidden code
`.gguf`	Single file for llama.cpp, Ollama, and LM Studio; usually already quantized
`.bin`, `.pt` (pickle)	Older PyTorch formats that can run arbitrary code when loaded

Having the weights lets you run, fine-tune, and inspect a model. It doesn't tell you its training data or how to rebuild it.

The openness spectrum (as of November 2026)

Category	What you get	Examples
Closed / API-only	Access, no weights	Claude, GPT-6, Gemini, Grok, Muse Wonder
Open weight, restrictive license	Weights, with limits on use, scale, or attribution	Kimi K3, Qwen3.8-Max, GLM-5.3, Muse Spark Open, Llama 4
Open weight, permissive license	Weights under Apache 2.0 or MIT	DeepSeek V4, gpt-oss, Gemma 4, Qwen3.8-27B
Open source AI (OSI definition)	Weights, training code, and data information	Few models qualify
Fully open	All that, plus the data, checkpoints, and logs	AI2's OLMo 3

"Open weight" isn't "open source"

Under the OSI Open Source AI Definition 1.0 (October 28, 2024), anyone may use, study, modify, and share it, for any purpose.

It requires	Most "open" models publish it?
Data information, enough to build a substantially equivalent system	No
Code, the complete source to train and run it	No
Parameters, the weights	Yes

Licenses vary a lot

License family	Terms	Examples
Apache 2.0	Use, modify, sell; keep notices; patent grant	gpt-oss, Gemma 4, Qwen3.8-27B, Muse Glimmer 30B
MIT	Use, modify, sell; keep the notice	DeepSeek V4 and V4.1
Revenue-tiered custom (the 2026 trend)	Free, except hosting it as a service above a revenue line	Kimi K3, Qwen3.8-Max, GLM-5.3, Muse Spark Open
Default non-commercial	Commercial use needs notice or authorization	MiniMax M3
Llama Community	Free under 700M monthly users; "Built with Llama"	Llama 3.x, Llama 4

Read the license, every upgrade

Family	Before	Now
MiniMax	M2, M2.5: modified MIT	M3: default non-commercial
GLM	GLM-5: MIT	GLM-5.3: custom license
Kimi	K2: modified MIT	K3: revenue-tiered

Check commercial use, thresholds, attribution, acceptable use, geography, outputs, patents, and your legal team's list.

Why labs give weights away

Reason	Example
Ecosystem and standards	Qwen derivatives: over 40% of new models on Hugging Face by mid-2025
Commoditize the complement	Zuckerberg's 2024 case for open models; Meta kept its best Muse closed
Sell the hosted version	Revenue-tiered licenses stop big hosts undercutting the lab
Talent and reputation	Researchers want to publish
Geopolitics and sovereignty	US 2025 AI Action Plan; Mistral's "AI sovereignty" pitch
Price pressure on rivals	A good-enough open model sets a price floor

The safety debate

For open weights	Against
Defenders can use them without permission	Release can't be undone; safeguards can be removed
Data never leaves your machines	LoRA fine-tuning cheaply strips safety training
You can study the internals	An "abliterated" Muse Spark Open appeared within two days
No single point of control	Dangerous capability arrives in open models months later

Most current policy treats today's open models as acceptable and watches the frontier.

Part 2

Running and choosing open models

Memory ≈ parameters × bytes per parameter

Precision	Bytes per parameter	8B	30B	120B
BF16 / FP16 (as released)	2	16 GB	60 GB	240 GB
8-bit	~1	~8.5 GB	~32 GB	~125 GB
4-bit (the common local default)	~0.55–0.6	~4.7 GB	**~17 GB**	~65 GB
~3-bit	~0.4	~3.5 GB	~13 GB	~50 GB

8-bit is close to lossless. Below about 3 bits, quality drops noticeably.

Then add room for context

KV bytes per token ≈ 2 × layers × KV heads × head dimension × bytes per value

For 48 layers, 8 KV heads, head dimension 128, 16-bit values:

- 2 × 48 × 8 × 128 × 2 ≈ 0.2 MB per token
- A 32K-token context ≈ 6 GB on top of the weights

Budget 20–50% extra for real coding contexts.

MoE changes speed, not memory

- Total parameters decide memory, because every expert must be loaded
- Active parameters decide speed, because only these are read per token

Model	Total	Active	What that means
Gemma 4 26B-A4B	26B	~4B	Memory of a 26B model, speed of a 4B
Kimi K3	2.8T	104B	~1.56 TB at 4-bit, so data-center hardware

Speed is mostly memory bandwidth

tokens per second ≲ memory bandwidth ÷ bytes of active weights

- A dense 30B at 4-bit (~17 GB) on ~270 GB/s tops out near 16 tokens/s
- A 3B-active MoE at 4-bit (~1.7 GB per token) is several times faster
- GPU VRAM is fast, but the model must fit
- Apple unified memory fits bigger models, with lower bandwidth

What fits where (approximate, 4-bit)

Machine	Comfortable
18 GB Mac	4B–14B dense: Gemma 4 E4B or 12B, small Qwen models
24 GB GPU	27B–32B dense: Qwen3.8-27B, Gemma 4 31B, Muse Glimmer 30B
36 GB Mac	The same 27B–32B models; Gemma 4 26B-A4B (fast)
64 GB Mac	70B-class dense (slow)
128 GB Mac or workstation	gpt-oss-120b; 120B-A12B MoE such as Nemotron 3 Super
8× 80 GB data-center GPUs	300B–800B MoE: GLM-5.3 at 4-bit (~420 GB)
Multi-node clusters	Kimi K3 / K3.1 Pro, DeepSeek V4-Pro, Qwen3.8-2.4T

Tools for running open models

Tool	Best for
Ollama	The easiest start: `ollama run gpt-oss:20b`
LM Studio	A GUI; Mac users; GGUF and Apple MLX models
llama.cpp	The engine under many local tools; `llama-server`
MLX / mlx-lm	Fast inference and LoRA fine-tuning on Apple silicon
vLLM	Serving a team or product
SGLang	Agent-heavy serving

Check free disk space first. Each tool keeps its own copy of the weights.

You don't have to own the hardware

Option	Examples
Router and marketplace	OpenRouter: one API key, hundreds of models
GPU clouds for open models	Together AI, Fireworks, Baseten, DeepInfra
Custom chips, very fast output	Groq, Cerebras
The labs' own APIs	DeepSeek, Moonshot, Z.ai, MiniMax, Alibaba Cloud, Mistral
Hyperscalers	AWS Bedrock, Google Vertex AI, Microsoft Foundry

The "same" open model can behave differently across hosts. Pin a provider when you evaluate.

Open models in coding harnesses

Harness	How
OpenCode	MIT-licensed; 200+ providers; detects local Ollama, LM Studio, vLLM
Claude Code	Point it at an Anthropic-compatible endpoint (Z.ai, Kimi, DeepSeek)
Codex CLI	`codex --oss` with Ollama or LM Studio

Anthropic doesn't support routing Claude Code to non-Claude models. It often works; test first.

Where do your prompts and code go?

Setup	Your data goes to
Self-hosted open weights	Nowhere outside your infrastructure
US or EU host serving Chinese-origin weights	That host, under its terms and jurisdiction
A Chinese lab's own API	That company's servers; Chinese law applies
A closed US lab's API	That lab's servers, under retention terms you can often set

Self-hosting fixes data flow, not model behavior.

When to choose open vs closed

Situation	Lean toward
Hardest agentic coding, large refactors	Closed frontier
Code or data can't leave your network	Open, self-hosted
Millions of bulk calls	Open, or a cheap closed tier
Fine-tuning on proprietary data	Open
Offline use, or a frozen model for audits	Open
Small team, no ML ops, wants it to just work	Closed

Where open models sit

> **Bryan's take:** For coding, Claude and Codex are the best in my opinion. Open models like Kimi sit at a different place in the chain.

Try it

Run a model and measure it (45 minutes)

- Check free disk and memory; pick a model that leaves headroom
- Run it in Ollama or LM Studio; have it explain and test a function
- Record tokens per second, memory used, and whether the test passes
- Compare with a hosted frontier model, then delete the local model

As a team, do a license review for a hypothetical product.

Discussion

1. Should "open source AI" require training data, or are open weights enough?
2. Is self-hosting a Chinese lab's weights meaningfully different from an American lab's API?
3. Revenue-tiered licenses restrict only large hosts. Is that fair? Is it "open"?
4. What share of your team's work could a local 30B model handle well?

Read more

- Module: 04 Open weight vs closed weight
- OSI Open Source AI Definition 1.0 (October 2024)
- Epoch AI, Open-weight models trail closed models in cyber capabilities by 6–7 months (2026)
- NIST, CAISI evaluation of DeepSeek AI models (2025)