GPT-OSS Guide: OpenAI's First Open Model for Local AI

February 16, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** GPT-OSS 20B is OpenAI's first open-weight model — a 20.9B MoE with only 3.6B active params per token. It ships in MXFP4 quantization at ~13GB, fits on a 16GB GPU, runs at 35-42 tok/s, and has 128K context. Apache 2.0 licensed. Strong at coding (60.7% SWE-Bench) and reasoning, weaker at creative writing. The sweet spot is 16-24GB VRAM. It's a genuine competitor at its size, but Qwen3-14B is still more versatile for general use. Start with: ollama run gpt-oss:20b

📚 **More on this topic:** Llama 4 vs Qwen3 vs DeepSeek · Qwen3 Guide · DeepSeek V3.2 Guide · What Can You Run on 16GB VRAM? · Planning Tool

OpenAI released open-weight models. Read that sentence again.

The company that spent years arguing against open weights dropped GPT-OSS in August 2025 — a 20.9B MoE model under Apache 2.0. You can download it, run it on your own hardware, modify it, and use it commercially. No API keys. No usage limits. No data leaving your machine.

The model is genuinely good. It hits 60.7% on SWE-Bench Verified, runs at 35-42 tok/s on a 16GB GPU, and packs 128K context into ~13GB of VRAM. But the question for budget builders isn't whether it's good — it's whether it's better than what you're already running.

What GPT-OSS Actually Is

Spec	Value
Total parameters	20.9B
Active parameters	3.6B per token
Architecture	MoE — 32 experts, top-4 routing
Layers	24
Context	128K tokens

Spec	Value
Quantization	MXFP4 (4.25 bits/param, quantization-aware)
Model size on disk	~13GB
License	Apache 2.0
Release	August 2025

The architecture is what makes GPT-OSS work on consumer hardware. It's a Mixture of Experts model — 32 experts per MoE layer, 4 selected per token. Only 3.6B of the 20.9B parameters activate on any given token. That's why a 21B model runs faster than a dense 14B.

The quantization matters too. MXFP4 isn't standard GGUF Q4 — OpenAI designed it specifically for this model. Only the MoE weights (90%+ of total params) get quantized to 4.25 bits. Attention and other components stay at higher precision. The model was post-trained with quantization baked in, so quality loss is minimal compared to post-hoc quantization.

Performance

Benchmarks

Benchmark	GPT-OSS 20B	Notes
MMLU	85.3%	Strong general knowledge
SWE-Bench Verified	60.7%	Agentic coding — competitive with much larger models
AIME 2024 (no tools)	42.1%	Math reasoning
AIME 2024 (with tools)	61.2%	Jumps with tool use
GPQA-Diamond	56.8%	PhD-level science
CodeForces Elo	2,230	Competitive programming

The SWE-Bench score is the headline number. 60.7% puts GPT-OSS 20B in the same conversation as models 5-10x its size for agentic coding tasks. The AIME scores jump significantly with tool use enabled, suggesting OpenAI optimized heavily for agentic workflows.

Speed

This is where the MoE architecture pays off. Only 3.6B active params means fast inference:

GPU	Context	Speed	Notes
RTX 3090 (24GB)	2K	~114 tok/s	Linux, no Flash Attention
RTX 3090 (24GB)	4K	~148 tok/s	Linux + Flash Attention
RTX 3090 (24GB)	50K	~55 tok/s	Speed drops with context
RTX 4080 (16GB)	Short	35-42 tok/s	Sweet spot for 16GB cards
RTX 3090 (24GB)	Windows	22-36 tok/s	Windows is significantly slower

On a 3090 with Flash Attention on Linux, it hits 148 tok/s at 4K context. That's absurdly fast. Even on a 16GB card, 35-42 tok/s is faster than any dense model at comparable quality.

The catch: speed drops as context grows. MoE models have higher prefill cost from expert routing, so throughput declines steeper than dense models at long context. On a 3090, expect ~62 tok/s at 128K — still usable, but a far cry from the 148 tok/s at short context.

VRAM Requirements

Setup	VRAM Usage	Practical?
Short context (1-4K)	~12-13GB	Fits on 16GB comfortably
Medium context (8-32K)	~13-14GB	Fine on 16GB
Long context (60-120K)	~14-15GB	Tight on 16GB, needs 24GB
Full 128K context	~23GB	Needs 24GB GPU + Flash Attention

By GPU Tier

8GB VRAM — Won't fit. The MXFP4 model alone is 13GB. Skip this tier.

12GB VRAM — Technically loads with short context, but no headroom. You'll hit OOM errors quickly. Not recommended.

16GB VRAM — The sweet spot for GPT-OSS. The model loads at ~12-13GB, leaving 3-4GB for KV cache. Practical context up to ~32K before things get tight. At 35-42 tok/s, it's the fastest model at this quality tier.

24GB VRAM — Full 128K context with Flash Attention. Speeds of 55-148 tok/s depending on context length. This is where GPT-OSS really opens up.

CPU only — Possible but slow. Expect 3-5 tok/s. Dense models like Qwen3-4B are a better fit for CPU inference.

Getting Started

```
# Install and run
ollama run gpt-oss:20b

# Set a practical context window
/set parameter num_ctx 32768

# For 24GB GPUs — full context
/set parameter num_ctx 128000
```

The default Ollama pull gives you the MXFP4 quantization. No need to hunt for specific quant files — the default is the good one.

If you're using LM Studio, look for the GGUF conversions from Unsloth on Hugging Face. Q4_K_M works well if MXFP4 isn't supported in your setup.

GPT-OSS vs the Competition at 16GB

The real question isn't whether GPT-OSS is good — it's whether it's better than what already exists at this VRAM tier.

	GPT-OSS 20B	Qwen3-14B	DeepSeek R1-14B
Architecture	MoE (3.6B active)	Dense (14B)	Dense (14B)
VRAM	~13GB	~9GB	~9GB
Speed (16GB GPU)	35-42 tok/s	18-25 tok/s	15-22 tok/s
Context	128K	32K	128K
SWE-Bench	60.7%	—	53.1%
AIME 2024	42.1%	76.3%*	69.7%
Chat quality	Functional, dry	Excellent	Good

	GPT-OSS 20B	Qwen3-14B	DeepSeek R1-14B
License	Apache 2.0	Apache 2.0	MIT

*Qwen3-14B AIME score is in `/think` mode.

GPT-OSS wins on speed and coding benchmarks. Qwen3-14B wins on reasoning, instruction following, and conversational quality. DeepSeek R1-14B wins on chain-of-thought reasoning depth.

The honest take: GPT-OSS outputs are accurate but dry. It prioritizes correctness over polish. Qwen3 produces more natural, conversational responses. If you're building an agentic coding pipeline, GPT-OSS is the pick. If you want a general-purpose assistant, Qwen3-14B at 9GB gives you better quality with 7GB of VRAM to spare.

Strengths and Weaknesses

GPT-OSS is good at:

- **Speed** — Fastest model at this quality level on 16GB hardware
- **Coding** — 60.7% SWE-Bench, strong agentic tool use
- **Long context** — 128K native, usable to the full length on 24GB
- **Accuracy** — Prioritizes correct answers over stylistic responses

GPT-OSS is weak at:

- **Creative writing** — Outputs are clinical and dry. Not the model for prose, storytelling, or conversational warmth
- **VRAM floor** — 13GB minimum locks out 8GB and 12GB users. Qwen3 serves those tiers, GPT-OSS doesn't
- **Math reasoning** — 42.1% AIME without tools is below Qwen3 and DeepSeek R1 at similar sizes
- **Windows performance** — Speeds drop to 22-36 tok/s on Windows. Linux is strongly recommended

---

Why GPT-OSS Matters

The model itself is solid but not category-defining at its size. Qwen3 and DeepSeek offer equal or better quality in most use cases. So why does GPT-OSS matter?

OpenAI releasing open weights validates the entire movement. The company that lobbied hardest against open models now ships one under Apache 2.0. That's not a technical achievement — it's a political one. It means the "open models are dangerous" argument lost. Every other lab now faces pressure to follow.

Competition drives quality. GPT-OSS forced Alibaba and DeepSeek to respond. Qwen3-Coder dropped weeks later. The local AI ecosystem gets better when every major lab is competing for open-weight mindshare.

It's a strong baseline. Even if it's not the best at any single task, GPT-OSS is competitive across coding, reasoning, and general knowledge in a 13GB package. For users who want one model that does everything reasonably well at high speed, it's a legitimate choice.

When to Choose GPT-OSS

Choose GPT-OSS 20B if:

- You have 16-24GB VRAM and want maximum inference speed
- You're building agentic coding workflows (SWE-Bench 60.7%)
- You need long context (128K native) on consumer hardware
- Accuracy matters more than prose style
- You want Apache 2.0 licensing from a major lab

Choose something else if:

- You have 8-12GB VRAM — GPT-OSS won't fit; use Qwen3-8B or 14B
- You need conversational warmth or creative writing — Qwen3 is better here
- Pure math/reasoning is your focus — DeepSeek R1 distills are the specialist tool
- You run Windows — performance drops significantly; consider dense models instead

```
ollama run gpt-oss:20b       # The speed king at 16GB
ollama run qwen3:14b         # The versatile alternative at 9GB
ollama run deepseek-r1:14b   # The reasoning specialist at 9GB
```

The bottom line: GPT-OSS 20B is a fast, accurate, well-licensed model that earned its spot in the local AI conversation. It's not the best at everything, but it's the fastest model at its quality tier — and it comes from the last company anyone expected to go open.
