Best Local LLMs for Mac in 2026 — M1, M2, M3, M4 Tested

February 5, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** On 8GB: Llama 3.2 3B or Phi-4 Mini via Ollama. On 16GB: Qwen 3 8B (Q4) is the best all-rounder. On 24GB: Qwen 3 14B for general use, DeepSeek-R1-Distill-14B for reasoning. On 48GB: Qwen 3 32B (Q4) is the standout — expert-level quality at 15-22 tok/s. On 64GB+: Llama 3.3 70B (Q4) matches GPT-3.5 quality. Use Ollama for the easiest setup, LM Studio for a GUI with MLX backend, or MLX-LM directly for 20-30% faster inference. Memory bandwidth is what determines your speed — an M3 Max generates tokens faster than an M4 Pro because it has more bandwidth, even though the M4 Pro is newer.

📚 **More on this topic:** Running LLMs on Mac M-Series · Qwen 3.5 on Mac: MLX vs Ollama · Qwen 3.5 Complete Guide · Ollama vs LM Studio · llama.cpp vs Ollama vs vLLM · VRAM Requirements · Planning Tool

Every Mac with Apple Silicon can run local LLMs. The question isn't whether — it's which model, and whether it'll be fast enough to actually use. A model that "fits" in memory but generates 3 tokens per second isn't useful. A smaller model at 40 tok/s is.

This guide gives you specific model recommendations for every Mac tier, with real performance numbers. No "it depends" — concrete picks you can install right now.

For setup instructions and general architecture details, see our complete M-series guide. This article focuses on which models to run.

Why Mac Is Different

Unified Memory Changes the Math

On a PC, your GPU has its own dedicated VRAM (typically 8-24GB). Models that don't fit in VRAM either won't run or crawl at 2-3 tok/s via offloading.

On Mac, there's no separate GPU memory. Your entire RAM pool — 8GB to 128GB — is shared between CPU and GPU. A Mac Mini with 48GB can load a 32B model that would need a $700+

used RTX 3090 on PC. A Mac Studio with 128GB runs 70B models that require $3,000+ in dual GPUs.

The tradeoff: Mac's memory bandwidth is lower than a discrete GPU's. An RTX 3090 pushes 936 GB/s. The M4 Pro pushes 273 GB/s. Token generation speed is directly proportional to memory bandwidth, so Mac is 30-60% slower per token for models that fit in a GPU's VRAM. But for models that don't fit — Mac wins by running them at all.

Memory Bandwidth Matters More Than Chip Generation

This is the counterintuitive part. An M3 Max (400 GB/s) generates tokens faster than an M4 Pro (273 GB/s), despite the M4 Pro being a newer chip. For LLM inference, bandwidth is the bottleneck, not compute.

Chip	Memory Bandwidth	Relative Speed
M1 / M2 / M3 / M4 (base)	68-120 GB/s	1x
M1 Pro / M2 Pro / M3 Pro / M4 Pro	150-273 GB/s	2-2.5x
M1 Max / M2 Max / M3 Max / M4 Max	300-546 GB/s	3-5x
M1 Ultra / M2 Ultra / M3 Ultra	400-800 GB/s	4-7x

Before buying: check the bandwidth of your specific chip, not just the generation. A $1,799 Mac Mini M4 Pro 48GB will be slower per token than a $2,700 Mac Studio M4 Max 64GB, even if you're running the same model.

Best Models by Mac Tier

8GB Macs (M1 / M2 / M3 / M4 base)

macOS needs 2-3GB for itself. You have about 5-6GB for a model. This limits you to 3B models comfortably or 7B-8B models with aggressive quantization and short context.

Model	Size	Speed	Best For
Llama 3.2 3B	~2 GB	25-35 tok/s	General chat, basic Q&A
Phi-4 Mini 3.8B	~2.3 GB	25-40 tok/s	Reasoning-heavy tasks for its size
Qwen 3 4B	~2.5 GB	20-35 tok/s	Multilingual, good instruction following

Model	Size	Speed	Best For
Gemma 3 4B	~2.5 GB	20-35 tok/s	Google's entry-level, decent at summarization
Llama 3.1 8B Q3	~4 GB	8-12 tok/s	Tight fit, quality tradeoff, short context only

The pick: Llama 3.2 3B for general use. It's fast, fits easily, and leaves room for context. If you need better reasoning and can tolerate shorter conversations, try Phi-4 Mini.

Skip: Any 7B+ model at Q4 or higher. It'll technically load but you'll have 1-2GB for context and system, which means frequent crashes and 4K token limits.

Honest take: 8GB Macs are for casual use with small models. If you're serious about local AI, the memory upgrade is worth it. An M4 MacBook Air starts at 16GB now — that's the minimum to buy in 2026.

16GB Macs (M1 Pro 16GB / M2 16GB / M3 16GB / M4 16GB)

The sweet spot for 7B-8B models. You have ~12-13GB available for the model and context.

Model	Size	Speed	Best For
Qwen 3 8B Q4	~5 GB	20-40 tok/s	Best all-rounder at this tier
Llama 3.1 8B Q4	~4.5 GB	25-40 tok/s	General assistant, well-tested
DeepSeek-R1-Distill-Qwen-8B	~4.5 GB	20-35 tok/s	Reasoning and chain-of-thought
Mistral Nemo 12B Q3	~6 GB	12-18 tok/s	128K context window, tight fit
Qwen 2.5 Coder 7B Q4	~4.5 GB	25-40 tok/s	Code generation and completion

The pick: Qwen 3 8B (Q4_K_M). It's the best 8B model available in early 2026 — strong instruction following, good at code, solid reasoning, and the /think mode gives you chain-of-thought when you need it. Run it at Q4_K_M for the best quality-to-size ratio. **Update:** Qwen 3.5 9B has since taken this spot — it fits in 6.6GB via Ollama and beats models 3x its size on reasoning benchmarks.

Runner-up for coding: Qwen 2.5 Coder 7B. If your primary use case is code completion and generation, this dedicated coding model outperforms general-purpose 8B models on programming tasks.

For reasoning: DeepSeek-R1-Distill-Qwen-8B. When you need step-by-step problem solving, the R1 distill's chain-of-thought training shows. It's slower because it generates reasoning tokens, but the answer quality is noticeably better on math and logic.

24GB Macs (M2 Pro 24GB / M4 16GB / M4 Pro 24GB)

The 14B tier opens up. This is where models start getting genuinely good.

Model	Size	Speed	Best For
Qwen 3 14B Q4	~9 GB	15-30 tok/s	Best general model at this tier
DeepSeek-R1-Distill-14B Q4	~8.5 GB	15-25 tok/s	Complex reasoning, math, analysis
Llama 3.1 8B Q8	~8.5 GB	25-45 tok/s	Maximum quality at 8B size
Mistral Nemo 12B Q4	~7.5 GB	18-30 tok/s	128K context for long documents
Qwen 3 32B Q3	~15 GB	6-10 tok/s	Possible but slow, minimal context

The pick: Qwen 3 14B (Q4_K_M). The jump from 8B to 14B is significant — better reasoning, more coherent long-form output, and noticeably fewer hallucinations. This is the Mac Mini M4 Pro 24GB sweet spot.

For reasoning: DeepSeek-R1-Distill-14B. The best reasoning model at this memory tier. Chain-of-thought responses are slower but more accurate than any general-purpose model of the same size.

Don't bother with: Qwen 3 32B at Q3. It technically fits in 24GB but you'll have ~6GB for context and overhead. The quality at Q3 quantization is degraded enough that a 14B at Q4 is usually better. Wait for more memory.

36-48GB Macs (M3 Pro 36GB / M4 Pro 48GB / M2 Max 32-48GB)

This is where it gets exciting. 32B models run comfortably and the quality jump is dramatic.

Model	Size	Speed	Best For
Qwen 3 32B Q4	~20 GB	12-22 tok/s	Best all-round pick, period
DeepSeek-R1-Distill-32B Q4	~20 GB	12-22 tok/s	Reasoning, math, complex analysis
Qwen 2.5 Coder 32B Q4	~20 GB	12-22 tok/s	Best local coding model
Qwen 3 14B Q8	~15 GB	18-35 tok/s	Maximum quality at 14B
Llama 3.3 70B Q2	~30 GB	3-6 tok/s	Only if you need 70B and have patience

The pick: Qwen 3 32B (Q4_K_M). This is the model that makes Mac local AI worthwhile. Expert-level responses on complex topics. Strong coding ability. Good creative writing. The /think mode

handles multi-step reasoning that 14B models fumble. At 48GB you have room for 16K+ context alongside the model.

For coding: Qwen 2.5 Coder 32B. If you write code all day, this is your model. It understands complex codebases, generates better functions, and catches more bugs than general-purpose models. 48GB gives you comfortable room for this plus context.

For reasoning: DeepSeek-R1-Distill-32B. The strongest reasoning model you can run locally on consumer hardware. Chain-of-thought on math, logic, and analysis tasks rivals much larger models.

The Mac Mini M4 Pro 48GB at $1,799 is the best-value setup in this tier. Silent, low power, runs 32B models all day. It's genuinely the sweet spot for local AI on Mac.

64-96GB Macs (M3 Max 64-96GB / M4 Max 64GB)

70B models become practical. You're matching what cloud APIs deliver.

Model	Size	Speed	Best For
Llama 3.3 70B Q4	~40 GB	8-15 tok/s	Best general-purpose large model
Qwen 2.5 72B Q4	~42 GB	8-14 tok/s	Strong on Chinese/multilingual tasks
Qwen 3 32B Q6	~26 GB	15-28 tok/s	32B at near-full quality
DeepSeek-R1-Distill-70B Q4	~40 GB	8-14 tok/s	Reasoning at scale
Mixtral 8x7B Q4	~26 GB	15-25 tok/s	Fast MoE option with 32K context

The pick: Llama 3.3 70B (Q4_K_M). The 70B tier is a significant leap. These models match GPT-3.5 and approach GPT-4 on many tasks. At 8-15 tok/s on M4 Max, it's slower than reading speed but perfectly usable for interactive chat. You get 128K context support, strong multilingual performance, and excellent instruction following.

Alternative: Qwen 3 32B at Q6/Q8. If you prefer speed over model size, running 32B at higher quantization gives you better quality than Q4 and 15-28 tok/s. You won't miss 70B on most everyday tasks.

128GB+ Macs (M4 Max 128GB / M3 Ultra 192GB)

The "no compromises" tier. Run whatever you want.

Model	Size	Speed	Best For
Llama 3.1 70B Q6	~55 GB	8-15 tok/s	Maximum 70B quality
Qwen 2.5 72B Q8	~75 GB	8-12 tok/s	Near-lossless 72B
Qwen3 235B-A22B Q4	~88 GB	5-10 tok/s	Largest MoE you can run locally
DeepSeek V3 Q3	~110 GB	3-5 tok/s	Frontier model, slow but impressive

The pick: Llama 3.1 70B at Q6 or Q8. Higher quantization means less quality loss. At 128GB you have room for the model at Q6 (~55GB) plus generous context. The quality improvement over Q4 is subtle but real, especially on complex reasoning and coding.

If you have 192GB (M3 Ultra): Qwen3 235B-A22B is the most capable model you can run locally. It's a Mixture of Experts model (22B active per token out of 235B total) that delivers frontier-class quality. Expect 5-10 tok/s — slow, but nothing else matches it locally.

MLX vs Ollama vs LM Studio: Which to Use

All three work on Apple Silicon. The difference is speed, ease, and interface.

Tool	Backend	Speed (8B Q4, M4 Max)	Setup	Best For
MLX-LM	Apple MLX	~95-110 tok/s	Python CLI	Maximum speed
Ollama	llama.cpp	~75-85 tok/s	One command	Simplest setup, API server
LM Studio	llama.cpp + MLX	~75-95 tok/s	GUI app	Visual interface, model browsing

When to Use MLX

MLX is Apple's native machine learning framework, built from the ground up for unified memory. It's consistently 20-30% faster than llama.cpp across model sizes, with the gap widening on larger models.

Use MLX when:

- You want maximum tok/s and are comfortable with Python
- You're building applications that need fast local inference

- You're running larger models where the 20-30% speed gap matters more

```
pip install mlx-lm
mlx_lm.generate --model mlx-community/Qwen3-8B-4bit --prompt "Hello"
```

The `mlx-community` organization on HuggingFace maintains hundreds of pre-converted models. If it exists in GGUF, it probably exists in MLX format too. For a head-to-head speed comparison on Qwen 3.5 specifically, see our MLX vs Ollama benchmark on Apple Silicon.

When to Use Ollama

Ollama wraps llama.cpp with dead-simple model management. One command to install, one command to run.

Use Ollama when:

- You want the fastest setup possible
- You need an API server for other apps (Open WebUI, Continue, etc.)
- You're new to local LLMs
- You want to switch between models quickly

```
ollama run qwen3:8b
```

For a detailed comparison of inference engines, see our llama.cpp vs Ollama vs vLLM guide.

When to Use LM Studio

LM Studio gives you a ChatGPT-like interface with model browsing, parameter controls, and conversation management. Recent versions use MLX as the backend on Mac, closing the speed gap with MLX-LM.

Use LM Studio when:

- You prefer a GUI over terminal
- You want to browse and compare models visually
- You need fine control over temperature, top-p, and other sampling parameters

---

Models That Technically Fit But Actually Crawl

This is the trap. A model can load into memory and still be useless.

Scenario	What Happens	Speed
70B Q4 on 64GB M4 Pro	Model loads, but only 4GB for context. M4 Pro's 273 GB/s bandwidth makes it slow.	4-7 tok/s
32B Q4 on 24GB	Model loads (~20GB), but 4GB left for context + OS. Frequent memory pressure.	6-10 tok/s with crashes
8B Q4 on 8GB M1	Model fits (~4.5GB) but context limited to ~2K tokens before swapping starts.	10-15 tok/s, drops under 5 when swapping
Mixtral 8x7B on 32GB	All 46.7B params load (~26GB Q4). Runs but barely any context headroom.	8-12 tok/s, 4K context max

The rule of thumb: the model file should be no more than 60-70% of your total memory. That leaves room for macOS, the KV cache (context), and framework overhead. A 20GB model on a 48GB Mac is comfortable. A 20GB model on a 24GB Mac is a knife's edge.

If you're right at the limit, drop to a lower quantization or a smaller model. A snappy 14B model is more useful than a sluggish 32B.

The Best Mac for Local AI in 2026

Budget	Buy This	Best Model It Runs	Why
$599	Mac Mini M4 16GB	Qwen 3 8B	Cheapest usable entry point
$1,399	Mac Mini M4 Pro 24GB	Qwen 3 14B	Great balance of cost and capability
$1,799	**Mac Mini M4 Pro 48GB**	**Qwen 3 32B**	**Best value for local AI**
$2,700	Mac Studio M4 Max 64GB	Llama 3.3 70B	Fastest bandwidth for big models
$3,500	Mac Studio M4 Max 128GB	Llama 3.1 70B Q8	No compromises

The Mac Mini M4 Pro 48GB at $1,799 is the sweet spot. It runs 32B models comfortably, sits silently on your desk, draws 30W under AI load, and costs less per year in electricity than a single month of ChatGPT Plus.

The Bottom Line

The model matters more than the tool. Pick the right model for your memory tier, use whichever app you're comfortable with, and don't try to squeeze a model that's too big. A fast small model beats a slow big one every time.

Quick decision tree:

- **8GB:** Llama 3.2 3B via Ollama. Accept the limitations.
- **16GB:** Qwen 3 8B via Ollama or LM Studio. This is where it gets useful.
- **24GB:** Qwen 3 14B. The Mac Mini M4 Pro entry config.
- **48GB:** Qwen 3 32B. The sweet spot — expert-quality responses on consumer hardware.
- **64GB+:** Llama 3.3 70B. Cloud-API quality, running on your desk.
- **128GB+:** Whatever you want. You've won the local AI hardware lottery.

Install Ollama (`curl -fsSL https://ollama.com/install.sh | sh`), pull your model, and start chatting. Your Mac is already a capable AI workstation — you just need to pick the right model for it.