What Can You Run on 8GB Apple Silicon? Local AI on a Budget Mac

February 26, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** On 8GB Apple Silicon, you have about 5-6GB available after macOS takes its share. Llama 3.2 3B (Q4) is the best all-rounder at 25-35 tok/s. Phi-4 Mini 3.8B fits comfortably and handles reasoning well. 7B models at Q4_K_M technically load but leave almost no room for context, leading to swapping and crashes. Stable Diffusion 1.5 works. Whisper transcription works great. The honest ceiling is 3-4B models for daily use.

More on this topic: Best Local LLMs for Mac | Running LLMs on Mac M-Series | Best Models Under 3B | VRAM Requirements | Ollama vs LM Studio

The base MacBook Air ships with 8GB. So does the base Mac Mini and the iPad Pro. Millions of these machines are out there, and most local AI guides skip right past them with a "you'll need at least 16GB" disclaimer.

That's not entirely wrong. But it's not the whole picture either. An 8GB Mac can run local AI. It just can't run everything, and the line between "works fine" and "unusable swapping mess" is thinner than you'd think. This guide shows you where it is.

How 8GB actually breaks down

What macOS takes

macOS needs 2-3GB for itself. Finder, WindowServer, kernel tasks, Spotlight indexing – it adds up. Open Activity Monitor on a fresh boot with nothing running and you'll see 3-4GB already "used" (Apple counts cached files as used, which is misleading, but the memory is still occupied).

That leaves you with about 5-6GB for your model, its context window, and whatever framework overhead Ollama or LM Studio adds.

The Metal GPU allocation cap

Apple's Metal framework limits GPU memory allocation to roughly 75% of unified memory. On an 8GB machine, that's about 6GB the GPU can address. This is a hard ceiling – no setting changes it, no hack bypasses it.

In practice, this means a model file over 5GB will start competing with macOS for the remaining memory. That's when the swapping begins.

Memory bandwidth: your real speed limit

Token generation speed on Apple Silicon is bottlenecked by memory bandwidth, not compute. The base M1 through M4 chips push 68-120 GB/s. Compare that to an M4 Pro at 273 GB/s or an M4 Max at 546 GB/s. This is why a base M1 generates tokens at roughly half the speed of an M1 Pro running the same model.

You can't change this. It's baked into the chip.

What actually works on 8GB

I tested each model using Ollama with default settings on a base M1 MacBook Air. Close everything else first – browsers especially. More on that later.

The sweet spot: 3-4B models

Model	Size on disk	Memory used	Speed	Verdict
Llama 3.2 3B Q4	~2 GB	~2.5 GB	25-35 tok/s	Best all-rounder
Phi-4 Mini 3.8B Q4	~2.3 GB	~3 GB	25-40 tok/s	Best for reasoning
Qwen 3 4B Q4	~2.5 GB	~3.2 GB	20-35 tok/s	Good multilingual
Gemma 3 4B Q4	~2.5 GB	~3.2 GB	20-35 tok/s	Solid summarization
SmolLM2 1.7B Q4	~1 GB	~1.5 GB	40-55 tok/s	Fast but limited

These models load in seconds, leave 2-3GB of headroom for context and system, and generate text fast enough to feel conversational. Activity Monitor stays green the entire time.

Llama 3.2 3B is the pick for general use. It handles chat, basic Q&A, simple summarization, and light creative writing at a speed that feels responsive. It's the model I'd tell anyone with 8GB to install first.

Phi-4 Mini is better than a 3.8B model has any right to be on reasoning tasks. Microsoft trained it specifically for chain-of-thought, and it shows on math and logic problems. Slightly slower on pure chat but noticeably smarter on structured problems.

SmolLM2 is interesting as a "background" model. At 1.7B parameters it's not going to write your novel, but for autocomplete, quick classification, or embedding into an app, it barely touches your memory.

The danger zone: 7-8B models

Model	Size on disk	Memory used	Speed	Verdict
Mistral 7B Q4_K_M	~4.4 GB	~5.1 GB	10-14 tok/s	Tight. Short context only
Llama 3.1 8B Q3_K_S	~3.6 GB	~4.3 GB	8-12 tok/s	Degraded quality, barely fits
Qwen 2.5 7B Q4_K_M	~4.5 GB	~5.3 GB	8-12 tok/s	Memory pressure within minutes

Here's where 8GB starts to hurt. A 7B model at Q4 quantization uses about 5GB of actual memory once you account for the KV cache and framework overhead. On an 8GB machine, that leaves 3GB for macOS – which is right at the edge.

Mistral 7B Q4_K_M will load and run. For the first few exchanges, it feels fine – 10-14 tok/s is readable. But watch Activity Monitor. Memory pressure creeps from green to yellow within a few minutes of conversation. Extend the context past 2-4K tokens and you'll see swap usage climb. At that point, generation speed drops to 3-5 tok/s and the whole system starts lagging.

Qwen 2.5 7B Q4_K_M has the same problem. Slightly larger memory footprint than Mistral, slightly faster to start swapping. It's the better model when it's running clean, but "running clean" doesn't last long on 8GB.

Llama 3.1 8B at Q3 is the desperation option. Dropping to Q3 quantization saves about a gigabyte but the quality hit is real – you'll notice more hallucinations, worse instruction following, and garbled output on complex prompts. Not worth it when Llama 3.2 3B at Q4 gives you better output at triple the speed.

The honest take: 7B models on 8GB are a demo, not a daily driver. They prove the hardware can technically run them. They don't prove it can run them well.

What to avoid entirely

- **13B+ models at any quantization.** They won’t fit. Don’t try.
- **Any model at Q8 or FP16.** Even a 3B model at FP16 uses 6GB. You need quantized models on 8GB.
- **70B “offloaded” to CPU.** Some guides suggest this. It generates 0.5-1 tok/s. That’s not inference, that’s a slideshow.
- **Multiple models loaded simultaneously.** Ollama can technically keep two models in memory. On 8GB, the second model will push the first into swap.

Beyond text: what else works

Image generation

Stable Diffusion 1.5 works on 8GB Apple Silicon. Draw Things (free Mac app) handles it well with Core ML optimizations. Expect 30-60 seconds per 512x512 image on an M1 base. Not fast, but functional.

SDXL is possible but tight. The base model alone needs ~4.5GB of memory. With macOS overhead, you’re right at the limit. It works if you close literally everything else, but I’ve had Safari silently reopen a tab in the background and kill a generation mid-image.

FLUX at full precision needs 22GB+. Even NF4 quantized FLUX needs 6-8GB of GPU memory, which exceeds what an 8GB Mac can allocate to the GPU. Skip it.

Model	Memory needed	Works on 8GB?	Speed
SD 1.5	~2-3 GB	Yes	30-60s per image
SDXL	~4.5 GB	Barely	60-120s, crash-prone
FLUX NF4	~6-8 GB	No	N/A
FLUX full	~22 GB	No	N/A

Whisper transcription

Whisper is the one thing that runs great on 8GB without any caveats.

The Whisper Small model (244M parameters) uses about 500MB of memory and transcribes faster than real-time. A 10-minute audio file finishes in about 2-3 minutes on an M1 base. The

Medium model (769M parameters) is tighter on memory but still usable, with better accuracy on difficult audio.

Use whisper.cpp for the best Apple Silicon performance – it's optimized for Metal and is 5-10x faster than the Python implementation. MacWhisper is a good GUI option if you don't want terminal.

For 8GB, stick with the Small or Base models for transcription. They're accurate enough for meeting notes, podcast transcription, and personal use.

Embedding models for RAG

Lightweight embedding models run fine on 8GB. Models like `nomic-embed-text` (137M parameters, ~275MB) or `all-minilm` (33M parameters, ~67MB) barely register on memory usage. You can run an embedding model alongside a 3B chat model without issues.

This means basic RAG pipelines work. Load documents into a local vector store, embed them, and query with Llama 3.2 3B. It's not going to match a cloud setup with GPT-4 and a hosted vector database, but for personal document search over a few hundred files, it's functional and completely private.

Making 8GB work: optimization tricks

Close your browser first

I know, I know. But it matters more than you think on 8GB. Safari with 10 tabs uses 1-2GB. Chrome with 10 tabs uses 2-4GB. That's the difference between a model running in memory and a model swapping to disk.

Before launching Ollama or LM Studio:

1. Close Safari/Chrome/Firefox entirely (not minimize, quit)
2. Close Slack, Discord, Teams, Zoom
3. Close VS Code if you don't need it during inference
4. Open Activity Monitor > Memory tab and check the pressure graph

If the graph is green and "Memory Used" is under 4GB, you're in good shape. If it's yellow, close more apps.

Ollama vs LM Studio: memory management

They handle memory differently, and it matters on 8GB.

Ollama pre-allocates memory at model load. A 2.5GB model file might reserve 3.2GB of unified memory upfront, accounting for KV cache buffers and safety margins. This means you see the full memory cost immediately but get consistent performance. If it fits at load time, it'll keep working.

LM Studio allocates more dynamically, loading on-demand and releasing memory when idle. This makes it more resilient when memory pressure spikes – it's less likely to hard-crash your system. But it can also be slightly slower to start generating.

On 8GB, I'd use Ollama for 3B models where you know it fits. Use LM Studio for 7B experiments where you want the safety net of dynamic memory management.

The Activity Monitor workflow

Open Activity Monitor and keep it visible while running models. The Memory tab tells you everything:

- **Memory Pressure graph**: Green = fine. Yellow = starting to swap. Red = stop, your model is too big.
- **Swap Used**: Under 500MB is normal macOS behavior. Over 1GB while running a model means the model doesn't actually fit. Over 2GB means you're getting noticeably slower responses.
- **Memory Used vs Cached Files**: macOS shows "Used" as a large number but "Cached Files" is memory that can be reclaimed. The real question is whether pressure is green.

The swap death threshold on 8GB is roughly this: if your model plus macOS exceeds 7.5GB of actual memory demand, the system starts writing to SSD. NVMe swap on Apple Silicon is fast enough that you won't notice for the first few hundred tokens. Then generation speed drops from 25 tok/s to 5 tok/s and your fans spin up. When that happens, don't wait it out. Close the model and start a smaller one.

Use Q4_K_M or smaller

On 8GB, Q4_K_M is the maximum quantization level worth using for any model over 3B parameters. Q5 and Q6 give marginal quality improvements but push memory usage up by 20-30%, which is the difference between "fits" and "swaps" on this hardware.

For 3B models, you can actually afford Q5_K_M or even Q6_K – the models are small enough that the extra memory is still within budget. If you're using Llama 3.2 3B primarily, try the Q5_K_M variant for slightly better output quality.

The honest take

8GB Apple Silicon can run local AI. I don't mean "technically possible" – I mean a 3B model at 30 tok/s is fast enough for real conversations, and the quality of 3-4B models in 2026 would have been impressive for a 13B model two years ago. Llama 3.2 3B and Phi-4 Mini are useful for everyday questions, writing drafts, and simple coding tasks. Not "useful for a small model" – actually useful.

But know the ceiling. 7B models are a stretch. 13B is impossible. You won't run coding assistants that understand large codebases. You won't run image generation beyond SD 1.5 without babysitting memory. And every time macOS decides to re-index Spotlight or update in the background, your model's context window is the first thing that gets squeezed.

If you haven't bought yet

The M4 MacBook Air starts at 16GB in 2026. Apple quietly killed the 8GB base configuration for the newest generation. If you're shopping now, 16GB is the minimum to buy for local AI. The $200 upgrade from 16GB to 24GB is worth it if you think you'll use models regularly – it opens up the 7-8B tier completely and even lets you experiment with 14B models.

If you already own an 8GB Mac

You're not locked out. Install Ollama, pull `llama3.2:3b`, and start using it today. Add Whisper for transcription. Set up a basic RAG pipeline with `nomic-embed-text` for searching your notes. These workflows run clean on 8GB and they're useful right now, today.

The 8GB Mac won't replace a 24GB GPU setup. But it'll tell you whether local AI fits your workflow before you spend real money on hardware. And for a lot of people, a 3B model running locally is all they actually needed.

Related guides

- Best Local LLMs for Mac in 2026

- Running LLMs on Mac M-Series
- Best Models Under 3B Parameters
- VRAM Requirements for Local LLMs
- Ollama vs LM Studio
- Ollama Troubleshooting Guide

Get notified when we publish new guides.

Subscribe — free, no spam

Source: https://insiderllm.com/guides/8gb-apple-silicon-local-ai/

Free guides for running AI locally