What Can You Actually Run on 12GB VRAM?

January 28, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** 12GB is where local AI gets comfortable. You can run 7B-8B models at near-lossless quality (Q6-Q8), 13B-14B models at the Q4 sweet spot with 25-32 tok/s, and SDXL image generation without workarounds. Start with Qwen 2.5 14B at Q4 for the best overall experience, or Llama 3.1 8B at Q6 for maximum quality on a smaller model. When you outgrow 12GB, a used RTX 3090 (24GB, ~$700-850) doubles your capacity.

📚 **More on this topic:** VRAM Requirements · What Can You Run on 8GB · What Can You Run on 16GB · Quantization Explained

If 8GB is the floor for local AI, 12GB is where you stop fighting your hardware and start actually using it.

The jump from 8GB to 12GB sounds like 50% more VRAM. In practice, it's a different experience entirely. You go from squeezing 7B models at minimum quantization to running 13B-14B models comfortably. You go from managing every megabyte to having actual headroom. You go from "can I run this?" to "which model should I choose?"

This guide covers exactly what fits on 12GB, what doesn't, and how to get the most out of the most popular VRAM tier for local AI.

Who This Is For

If you own any of these cards, this guide is for you:

GPU	VRAM	Architecture	Notes
RTX 3060 12GB	12GB	Ampere	The budget AI champion. ~$200 used.
RTX 4070	12GB	Ada Lovelace	Faster than 3060, same VRAM
RTX 4070 Super	12GB	Ada Lovelace	Slightly faster still
AMD RX 6700 XT	12GB	RDNA 2	Works with ROCm on Linux
Intel Arc A770	16GB	Alchemist	More VRAM but less mature software

The RTX 3060 12GB is the most common card in this tier — and the most recommended entry point for local AI on a budget. The RTX 4070 is significantly faster (roughly 30-50% more tok/s) but costs more than double. Both have the same 12GB of VRAM, which is what determines what models you can run.

Why 12GB Is the Sweet Spot

The gap between 8GB and 12GB is bigger than the numbers suggest.

On 8GB, a 7B model at Q4 takes ~5GB, leaving 3GB for context and overhead. You're always on the edge. One too-long conversation and you're out of memory.

On 12GB, that same model takes the same 5GB — but now you have 7GB of headroom. That's enough for longer contexts, higher quantization, or jumping to a 13B-14B model entirely. The math shifts from "what can I squeeze in?" to "what quality level do I want?"

Here's the practical difference:

Capability	8GB	12GB
7B-8B models	Q4 only, tight context	Q6-Q8, comfortable context
13B-14B models	Q2-Q3, painful	Q4-Q5, fast and usable
30B+ models	Won't fit	Tight but possible (Q3)
SDXL image gen	Needs hacks	Works out of the box
Context window (7B)	2-4K tokens	8-16K tokens

That extra 4GB transforms local AI from a demo into a daily tool.

What Runs Well on 12GB

7B-8B Models at High Quantization

With 12GB, you can stop using minimum quantization and start running 7B models the way they were meant to be run — at Q6 or Q8, where quality is near-lossless.

Model	Quant	VRAM	Speed (RTX 3060)	Quality
Llama 3.1 8B	Q8_0	~8 GB	~28 tok/s	Near-lossless
Llama 3.1 8B	Q6_K	~6.5 GB	~35 tok/s	Excellent
Mistral 7B	Q6_K	~5.5 GB	~38 tok/s	Excellent
Qwen 2.5 7B	Q8_0	~7.5 GB	~30 tok/s	Near-lossless

At Q6_K, you retain ~97% of the original model's quality — barely measurable on perplexity benchmarks. At Q8, you're at ~99%. On 8GB, these quantizations didn't fit or left no room for context. On 12GB, they run fast with headroom to spare.

The RTX 4070 is roughly 30-50% faster than the RTX 3060 at the same model and quantization — expect ~45-58 tok/s for 7B models at Q4-Q6.

13B-14B Models at Q4-Q5: The Real Unlock

This is the tier that makes 12GB worth it. A 14B model is noticeably smarter than a 7B — better reasoning, better instruction following, better code, longer coherent output. On 8GB, you couldn't run these. On 12GB, they're your daily driver.

Model	Quant	VRAM	Speed (RTX 3060)	Best For
Qwen 2.5 14B	Q4_K_M	~9 GB	~30 tok/s	Best overall at this tier
Mistral Nemo 12B	Q4_K_M	~8 GB	~32 tok/s	Strong reasoning, 128K context
Llama 2 13B	Q4_K_M	~8.5 GB	~28 tok/s	Solid general use
DeepSeek Coder V2 Lite	Q4_K_M	~5 GB	~35 tok/s	Coding (MoE, 2.4B active)

Qwen 2.5 14B at Q4_K_M is the standout. At ~9GB, it leaves ~3GB for context and overhead — enough for 4-8K tokens of context comfortably. It outperforms CodeStral-22B and DeepSeek Coder 33B on coding benchmarks, and it's a strong general-purpose model.

Mistral Nemo 12B is the runner-up. Co-developed by NVIDIA and Mistral AI, it was trained with quantization awareness, meaning FP8 inference works without quality loss. Apache 2.0 licensed with a 128K context window.

```
ollama pull qwen2.5:14b
ollama pull mistral-nemo
```

Where 12GB Beats 8GB Most

The biggest quality jump isn't just bigger models — it's the same models at better quantization. A Llama 3.1 8B at Q6_K on 12GB is measurably better than the same model at Q4_K_S on 8GB, and you'll notice it on complex reasoning, precise coding, and long-form writing.

And with 14B models available, you get a genuine capability step-up. The bigger model at lower quant beats the smaller model at higher quant rule means Qwen 2.5 14B at Q4 outperforms Qwen 2.5 7B at Q8 on most tasks.

What's Possible But Tight

30B+ Models at Low Quantization

Can you squeeze a 30B model into 12GB? Not really. A 32B model at Q3_K_M still needs ~15-17GB for weights alone — well beyond 12GB. Even Q2 won't fit.

What you can do is partial offloading: load some layers on GPU and the rest in system RAM. But the speed penalty is brutal. Models that run at 30 tok/s fully on GPU drop to 3-8 tok/s with partial offloading. PCIe bandwidth becomes the bottleneck, and inference feels like watching paint dry.

The verdict: If you need 30B+ model capability, you need more VRAM, not more optimization. A Qwen 2.5 14B at Q4 on 12GB will outperform a forced 32B at Q2 with partial offloading — it's faster, more coherent, and actually pleasant to use.

Longer Context Windows

Context windows eat VRAM. With a 14B model at Q4_K_M (~9GB), you have ~3GB left for KV cache and overhead. That gives you:

- **4096 tokens**: Comfortable, fast, no issues
- **8192 tokens**: Usable, some pressure
- **16K tokens**: Tight. Works with KV cache quantization (q8_0 or q4_0)
- **32K+**: Probably spilling to CPU RAM. Expect slowdowns.

If you need long contexts regularly — processing documents, maintaining extended chat histories — you have two options: use a smaller model (7B at Q4 gives much more context headroom) or upgrade to 24GB.

Tip: Ollama and llama.cpp support KV cache quantization. Switching from FP16 to q8_0 KV cache roughly doubles your available context length with minimal quality impact.

What Won't Work

Save yourself the troubleshooting:

- **70B models**: Need 24GB+ even at Q4. A 70B at Q4 requires ~40GB. Not happening on 12GB.
- **32B models at usable quality**: Q3 is the lowest you'd want, and it still needs ~16GB. 12GB isn't enough.
- **Fine-tuning 14B+ models**: LoRA training on a 14B model needs 16-24GB minimum. Fine-tuning a 7B is possible on 12GB with aggressive settings, but slow.
- **Multiple models simultaneously**: One at a time. Loading two 7B models would eat your entire VRAM.

If 70B models or fine-tuning are priorities, you need the 24GB tier.

Image Generation on 12GB

12GB is the comfortable tier for image generation. Where 8GB users need hacks and workarounds, 12GB users just generate.

Stable Diffusion 1.5: Runs Great

SD 1.5 uses ~4GB VRAM, leaving 8GB of headroom. Generation is fast, ControlNet works, and the massive community ecosystem of LoRAs and checkpoints is fully accessible.

Resolution	Time (RTX 3060)	Notes
512x512	~4 seconds	Fast, tons of headroom
768x768	~8 seconds	Comfortable
1024x1024	~15 seconds	Still easy

SDXL: Comfortable

This is the big upgrade from 8GB. SDXL uses ~7-8GB for the base model, which was a tight squeeze on 8GB but leaves room on 12GB. The refiner can run sequentially without tricks.

Resolution	Time (RTX 3060)	Notes
1024x1024	~20 seconds	Comfortable, no optimizations needed
1024x1024 + refiner	~35 seconds	Sequential, works smoothly

No need for `--medvram` hacks. No need for specialized VAEs. SDXL just works on 12GB.

Flux: Doable

Flux is the newer, higher-quality model. The NF4 quantized version fits on 12GB and produces excellent results.

Resolution	Time (RTX 3060)	Notes
1024x1024 (NF4, 20 steps)	~80 seconds	Slower, but usable

Flux is noticeably slower than SDXL on the same hardware. For rapid iteration and experimentation, SDXL is still the better choice on 12GB. Flux is worth the wait when you need photorealistic output or better text rendering.

Best Models for 12GB GPUs (Ranked)

Here's what to install, in order:

1. Qwen 2.5 14B — The best model that fits comfortably. Smarter than any 7B, excellent at coding, reasoning, and general tasks. Your daily driver.

```
ollama pull qwen2.5:14b
```

2. Llama 3.1 8B at Q6_K — When you want headroom for longer context or prefer near-lossless quality on a smaller model. Fast and dependable.

```
ollama pull llama3.1:8b
```

(Ollama defaults to Q4; for Q6, download a GGUF from Hugging Face and import it, or use LM Studio.)

3. Mistral Nemo 12B — Strong reasoning, 128K context window, quantization-aware training. A great all-rounder that was designed for this VRAM tier.

```
ollama pull mistral-nemo
```

4. DeepSeek Coder V2 Lite — The coding specialist. MoE architecture means only 2.4B parameters are active per inference, making it fast and memory-efficient despite 16B total params.

```
ollama pull deepseek-coder-v2:16b
```

5. Qwen 2.5 7B at Q8 — Maximum quality on a smaller model. Leaves tons of headroom for context and runs fast. Great for tasks where you want the highest fidelity.

```
ollama pull qwen2.5:7b
```

New to local AI? Start with our Ollama setup guide — one command to install, one command to run.

→ Check what fits your hardware with our Planning Tool.

Tips to Maximize 12GB

1. Use Q5_K_M as Your Default

On 8GB, Q4_K_S is the standard. On 12GB, you can afford to step up. Q5_K_M gives ~95% quality (vs. Q4_K_M's ~92%) with only 15-20% more VRAM. For 7B-8B models, Q5 or Q6 is the sweet spot on this tier.

2. Set Context to 8192

The default context in many tools is 2048 or 4096. On 12GB with a 14B model, you can comfortably push to 8192:

```
ollama run qwen2.5:14b --num-ctx 8192
```

For 7B models at Q4-Q5, you can go even higher — 16K is feasible.

3. Quantize Your KV Cache

If you want longer contexts without upgrading, quantize the KV cache:

```
# In llama.cpp
./llama-cli -m model.gguf -c 16384 --cache-type-k q8_0 --cache-type-v q8_0
```

This roughly halves KV cache VRAM usage with minimal quality impact. A 14B model at Q4 with q8_0 KV cache can handle 16K context on 12GB.

4. Monitor with nvidia-smi

```
nvidia-smi -l 1
```

Watch for VRAM usage creeping above 11GB — that's when you're on the edge. If it happens during long conversations, reduce context or switch to a smaller model.

5. Close GPU-Hungry Background Apps

Same as with 8GB: Chrome with hardware acceleration, game launchers, video players — all eat VRAM. On 12GB this is less critical than on 8GB, but it still matters when running 14B models near the VRAM ceiling.

When to Upgrade (And What To)

You've outgrown 12GB when:

- You need 30B+ models at usable quality
- You want to fine-tune models larger than 7B

- You need 32K+ context windows on 14B models
- You want to run 70B quantized for the smartest local model available

Here's the upgrade path:

GPU	VRAM	Street Price (Jan 2026)	What It Unlocks
RTX 5060 Ti 16GB	16GB	~$429-500	14B at Q5-Q6, 30B at Q3 (tight)
Used RTX 3090	24GB	~$700-850	32B at Q4, 70B quantized, fine-tuning
RTX 4090	24GB	~$2,000+	Same as 3090, faster inference

The used RTX 3090 is the clear winner for value. At $700-850, it doubles your VRAM from 12GB to 24GB — unlocking an entirely different tier of models. The 5060 Ti 16GB is a smaller step (only 4GB more) and harder to find at MSRP due to GDDR7 shortages.

If your budget allows, skip the 16GB tier and go straight to 24GB. The jump from 12GB to 24GB is transformative. The jump from 12GB to 16GB is incremental.

The Bottom Line

12GB is where local AI stops being a compromise and starts being a tool. You can run 14B models at interactive speeds, generate SDXL images without workarounds, and choose quality levels instead of praying things fit.

The practical advice:

1. Install Ollama and pull `qwen2.5:14b`. That's your main model.
2. Keep `llama3.1:8b` around for when you need longer context or faster responses.
3. Use Q5_K_M as your default quantization — you have the VRAM for it.
4. When 12GB isn't enough, a used RTX 3090 is the move. Skip 16GB and go straight to 24GB.

You have a genuinely capable local AI machine. Use it.

Related Guides

- What Can You Run on 8GB VRAM?
- How Much VRAM Do You Need for Local LLMs?

- GPU Buying Guide for Local AI

Sources: Hardware Corner RTX 3060 LLM Guide, Hardware Corner RTX 4070 Guide, LocalLLM.in Ollama VRAM Guide, NVIDIA Mistral NeMo Blog, Qwen2.5 Speed Benchmark, PropelRC Best GPU for SD/Flux

---

Source: https://insiderllm.com/guides/what-can-you-run-12gb-vram/

Free guides for running AI locally