AMD vs NVIDIA for Local AI: Is ROCm Finally Ready?

January 27, 2025 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** NVIDIA is still easier. But AMD is now genuinely viable—especially on Linux. The RX 7900 XTX offers 24GB VRAM for $700-950, delivering 85-95% of RTX 4090 performance at 60-70% of the price. If you're on Linux and comfortable with occasional troubleshooting, AMD makes sense. If you're on Windows or want zero friction, stick with NVIDIA.

📚 **More on this topic:** GPU Buying Guide · Used RTX 3090 Guide · VRAM Requirements

Every few months, someone asks: "Can I use AMD for local AI yet?"

For years, the answer was "technically yes, but don't." ROCm was a mess. Driver support was spotty. Half the tools didn't work. NVIDIA's CUDA ecosystem was so dominant that choosing AMD meant signing up for endless troubleshooting.

That's changing. ROCm 6.x and 7.x have brought real improvements. PyTorch now officially supports AMD on Windows. Ollama, LM Studio, and llama.cpp all work with AMD GPUs. The RX 7900 XTX offers 24GB of VRAM—matching the RTX 4090—for hundreds less.

So is ROCm finally ready? The honest answer: it depends on who you are and what you're willing to tolerate.

This guide gives you the real picture—no cheerleading, no AMD bashing. Just practical advice for making the right choice.

The Short Answer

NVIDIA is still the safer choice. Install drivers, install Ollama, run models. It works on Windows and Linux with minimal friction. If your time is valuable and you don't enjoy debugging, NVIDIA saves headaches.

AMD is now a legitimate option. The RX 7900 XTX delivers 85-95% of RTX 4090 inference performance at 60-70% of the cost. Software support has improved dramatically. If you're on Linux, comfortable with command lines, and willing to occasionally troubleshoot, AMD offers excellent value.

The deciding factors:

- **Windows user?** Stick with NVIDIA.
- **Linux user who doesn't mind tinkering?** AMD is worth considering.
- **Beginner?** NVIDIA. No question.
- **Budget-constrained and want maximum VRAM?** AMD wins on price-per-GB.

Why NVIDIA Has Dominated Local AI

The CUDA Ecosystem

NVIDIA's CUDA platform has a 15+ year head start. Every major AI framework—PyTorch, TensorFlow, vLLM, llama.cpp—was built on CUDA first. The documentation is comprehensive. The community is massive. When something breaks, someone has already solved it.

This isn't just marketing. It's a real advantage:

- **More tutorials** — Search any AI problem, the solution assumes CUDA
- **Faster updates** — New models and tools support CUDA on day one
- **Better optimization** — NVIDIA's Tensor Cores and TensorRT are deeply integrated
- **Wider compatibility** — Virtually every AI application works out of the box

"It Just Works"

On NVIDIA hardware:

```
# Install Ollama
curl -fsSL https://ollama.com/install.sh | sh

# Run a model
ollama run llama3.1:8b
```

That's it. GPU detected automatically. Full acceleration enabled. No environment variables to set, no drivers to manually configure, no GitHub issues to hunt through.

This simplicity has real value—especially if you're new to local AI or just want to use models without becoming a systems administrator.

What’s Changed with AMD

ROCm 6.x and 7.x Improvements

AMD’s ROCm (Radeon Open Compute) platform has matured significantly:

ROCm 7.2 (2025):

- First release with official Windows support alongside Linux
- PyTorch now available as public preview on Windows
- Support for Radeon RX 7000 and 9000 series consumer cards
- Integration into major Linux distributions (Ubuntu, Red Hat, OpenSUSE)

vLLM Integration:

- As of January 2026, 93% of vLLM test groups pass on AMD CI pipeline (up from 37% in November 2025)
- Pre-built Docker images available—no more building from source
- Validated on Instinct MI300/MI350 datacenter GPUs

Consumer GPU Support:

- RX 7900 XTX, 7900 XT, 7800 XT now officially supported
- Older RDNA2 cards (6800 XT, 6900 XT) work with workarounds
- PyTorch and ONNX-EP available for Radeon GPUs

llama.cpp and the HIP Backend

The llama.cpp project—which powers Ollama and LM Studio—has a mature HIP backend for AMD GPUs. This means the core inference engine works well on AMD, even if some higher-level tools have quirks.

For direct llama.cpp usage:

```
# Build with HIP support
cmake -B build -DGGML_HIP=ON
cmake --build build

# Run inference
./build/bin/llama-cli -m model.gguf -p "Hello" -ngl 99
```

Vulkan as a Universal Fallback

If ROCm doesn't work for your specific card, Vulkan provides a cross-platform alternative. It's slower than native ROCm but works on virtually any GPU:

- LM Studio uses Vulkan when ROCm fails
- llama.cpp has a Vulkan backend
- Useful for older or unsupported AMD cards

In some cases, Vulkan actually outperforms ROCm due to driver issues—one benchmark showed Vulkan at 24 tok/s versus ROCm at 17 tok/s on the same hardware.

Which AMD Cards Work for Local AI

The Top Tier: RX 7900 XTX (24GB)

The RX 7900 XTX is AMD's best option for local AI. Period.

Spec	RX 7900 XTX
VRAM	24GB GDDR6
Memory Bandwidth	960 GB/s
Stream Processors	6,144
TDP	355W
New Price	~$950
Used Price	~$750

Why it matters: 24GB of VRAM matches the RTX 3090 and RTX 4090. You can run 70B models at Q4 quantization, 13B models at high precision, and have headroom for longer contexts.

Performance reality: In head-to-head benchmarks with the RTX 4090:

- DeepSeek R1 7B: 7900 XTX wins by 13%
- DeepSeek R1 14B: 7900 XTX wins by 2%
- Llama 3 8B (llama.cpp): 4090 at 142 tok/s vs 7900 XTX at 89 tok/s
- Llama 3 70B Q4: 4090 at 38 tok/s vs 7900 XTX at 23 tok/s

The pattern: AMD competes well on smaller models and specific optimized workloads, but NVIDIA pulls ahead on larger models and general-purpose inference.

The Value Pick: RX 7900 XT (20GB)

Spec	RX 7900 XT
VRAM	20GB GDDR6
Memory Bandwidth	800 GB/s
Stream Processors	5,376
TDP	315W
New Price	~$675
Used Price	~$530

The case for it: 20GB is still a lot—more than any RTX 40-series card except the 4090. At $530-675, it's the cheapest way to get this much VRAM from a current-generation card.

Tradeoffs: ~15-20% slower than the 7900 XTX. The 4GB VRAM difference rarely matters for 7B-13B models but limits headroom for 30B+ models.

Older Options: RX 6800 XT / 6900 XT (16GB)

Card	VRAM	Used Price	Notes
RX 6900 XT	16GB	~$350-400	Best older AMD option
RX 6800 XT	16GB	~$280-350	Good budget entry

Reality check: 16GB is workable but limiting. You can run 7B models comfortably and 13B models at Q4 quantization. 30B+ models require aggressive compression or won't fit.

ROCm support: RDNA2 cards work but aren't officially supported in newer ROCm versions. You may need `HSA_OVERRIDE_GFX_VERSION` workarounds. Expect more troubleshooting than RDNA3 cards.

What to Avoid

Radeon VII (16GB HBM2): Deprecated in ROCm. Was interesting for its HBM2 bandwidth, but software support is ending. Not recommended for new setups.

RX 7600/7700 series (8-12GB): Too little VRAM for serious LLM work. You're limited to 7B models at best. At this tier, an RTX 3060 12GB is a better choice for the software compatibility.

RX 9070/9070 XT: Too new. ROCm support is still being developed. Wait 6+ months for drivers to mature.

AMD GPU Comparison Table

GPU	VRAM	New Price	Used Price	Best For
RX 7900 XTX	24GB	$950	$750	Maximum AMD capability
RX 7900 XT	20GB	$675	$530	Best value high-VRAM
RX 6900 XT	16GB	—	$350-400	Budget option, older
RX 6800 XT	16GB	—	$280-350	Entry-level AMD

Software Compatibility: The Real Picture

What Works Well

Ollama — Works with ROCm on supported cards. Official documentation available from AMD. Some cards require the community fork with extended GPU support.

```
# Check if your GPU is detected
ollama list
# Should show your model and use GPU acceleration
```

llama.cpp — HIP backend is mature and well-maintained. Build with `-DGGML_HIP=ON` and you're running native AMD acceleration.

LM Studio — Supports ROCm on Linux, with Vulkan fallback on Windows. The experience is improving but occasionally requires manual backend selection.

PyTorch — ROCm builds available for Linux; Windows now in public preview. Most training and inference code works, though some CUDA-specific operations may need adjustment.

What's Still Rough

Stable Diffusion (Automatic1111/ComfyUI): Mixed results. Some users report success, others fight configuration issues for hours. NVIDIA remains far easier for image generation.

Anything requiring cuDNN: CUDA-specific libraries don't translate. If a tool explicitly requires cuDNN, it won't work on AMD.

Cutting-edge models on day one: New model architectures often launch with CUDA-only support. AMD compatibility follows weeks or months later.

Fine-tuning: The ROCm variant of xformers doesn't support consumer GPUs like the 7900 XTX. This blocks tools like Unsloth for efficient fine-tuning. Inference works; training is limited.

Software Compatibility Matrix

Software	AMD Support	Notes
Ollama	Good	ROCm on Linux, improving on Windows
LM Studio	Good	ROCm + Vulkan fallback
llama.cpp	Good	HIP backend mature
PyTorch	Good	ROCm builds available
vLLM	Good	93% tests passing, Docker images available
Automatic1111	Partial	Works but requires effort
ComfyUI	Partial	ROCm integration improving
Text-generation-webui	Good	AMD support documented

Performance: AMD vs NVIDIA Head-to-Head

Inference Benchmarks

Real-world performance depends heavily on the specific model, quantization, and software stack. Here's what benchmarks show:

Test	RTX 4090	RX 7900 XTX	Winner
Llama 3 8B Q4_K_M (llama.cpp)	142 tok/s	89 tok/s	NVIDIA (+60%)

Test	RTX 4090	RX 7900 XTX	Winner
Llama 3 70B Q4_K_M	38 tok/s	23 tok/s	NVIDIA (+65%)
DeepSeek R1 7B	100%	113%	AMD (+13%)
DeepSeek R1 14B	100%	102%	AMD (+2%)
DeepSeek R1 32B	100%	96%	NVIDIA (+4%)

The pattern: NVIDIA generally leads in raw tok/s with llama.cpp. AMD can match or beat NVIDIA on specific optimized workloads (like DeepSeek with AMD-optimized kernels). For general-purpose inference, expect the 7900 XTX to deliver roughly 60-70% of RTX 4090 speed.

The VRAM Value Proposition

Where AMD shines is price-per-GB of VRAM:

GPU	VRAM	Street Price	$/GB
RTX 4090	24GB	$1,800+	$75/GB
RTX 4080 Super	16GB	$1,000	$62/GB
RTX 3090 (used)	24GB	$750-900	$31-37/GB
RX 7900 XTX	**24GB**	**$750-950**	**$31-40/GB**
RX 7900 XT	**20GB**	**$530-675**	**$26-34/GB**

For LLM inference, VRAM capacity often matters more than raw compute speed. A 7900 XT with 20GB can run models that a faster RTX 4080 (16GB) cannot fit at all.

→ Not sure what fits? Try our Planning Tool.

Power Consumption

Both AMD flagships run hot:

- **RX 7900 XTX:** 355W TDP
- **RX 7900 XT:** 315W TDP
- **RTX 4090:** 450W TDP
- **RTX 4080 Super:** 320W TDP

AMD actually has a slight efficiency advantage at the high end. Plan for a 750W+ PSU regardless.

The Linux Factor

ROCm Is Linux-First

This is the most important thing to understand: **ROCm works best on Linux.**

On Linux:

- Full ROCm stack available
- All math libraries supported
- Mature driver integration
- Community documentation and support

On Windows:

- PyTorch in public preview (not full ROCm)
- Limited to HIP SDK components
- Many tools fall back to slower Vulkan
- Not recommended for serious work

If you're considering AMD for local AI, you should be comfortable with Linux—or willing to learn.

Which Linux Distros Work Best

Ubuntu 22.04/24.04 — Official support, best documentation, recommended for beginners.

Fedora — Good community support, works well with recent ROCm versions.

Arch Linux — Works but rolling releases can break ROCm compatibility. For experienced users only.

Pop!_OS — Ubuntu-based, generally works, popular with AMD users.

Windows Reality

On Windows, your options are:

1. **Vulkan backend** — Works but slower than native ROCm
2. **PyTorch ROCm preview** — Limited to PyTorch workloads
3. **WSL2 with ROCm** — Adds complexity, mixed results

If you must use Windows, NVIDIA is the clear choice. AMD on Windows is possible but not recommended.

Who Should Consider AMD

Linux users comfortable with troubleshooting. If you already run Linux and don't mind reading GitHub issues occasionally, AMD works well and saves money.

Budget-conscious buyers who want maximum VRAM. The 7900 XT at $530-675 for 20GB is unmatched. Nothing from NVIDIA at that price comes close in VRAM.

Privacy-focused users. AMD's driver stack is more open-source than NVIDIA's proprietary CUDA. If that matters to you, AMD aligns better philosophically.

People with existing AMD systems. Already have an AMD CPU and motherboard? An AMD GPU keeps your system consistent and may have better power management integration.

Tinkerers who enjoy the process. If configuring software and occasionally debugging is part of the fun for you, AMD delivers great value.

Who Should Stick with NVIDIA

Windows users. Full stop. NVIDIA's Windows support is years ahead. Don't fight this battle.

Beginners. Your first local AI experience should be smooth. NVIDIA removes variables. Start there, consider AMD later if you want to optimize costs.

Anyone who needs Stable Diffusion / image generation. NVIDIA's ecosystem for image gen is far more mature. AMD works but requires more effort.

Users who value their time. If an hour of troubleshooting costs you more than the price difference, buy NVIDIA and move on.

People running production workloads. If reliability matters more than cost, NVIDIA's track record and support are worth the premium.

Practical Setup Tips for AMD

Pre-Purchase Checklist

Before buying an AMD GPU for local AI:

1. **Verify ROCm support** — Check AMD's official compatibility matrix for your specific card
2. **Plan for Linux** — Windows support is limited; Linux is the real platform
3. **Check your distro** — Ubuntu 22.04/24.04 has the best support
4. **Research your target software** — Verify Ollama/LM Studio/your tools support AMD

Basic Installation (Ubuntu)

```
# Add ROCm repository
wget https://repo.radeon.com/amdgpu-install/latest/ubuntu/focal/amdgpu-install_6.0.60000-1_all.
sudo apt install ./amdgpu-install_6.0.60000-1_all.deb

# Install ROCm
sudo amdgpu-install --usecase=rocm

# Add user to render group
sudo usermod -a -G render $USER
sudo usermod -a -G video $USER

# Reboot
sudo reboot

# Verify installation
rocminfo
```

Installing Ollama with ROCm

```
# Install Ollama
curl -fsSL https://ollama.com/install.sh | sh

# Ollama should detect ROCm automatically
# Verify GPU is being used
ollama run llama3.2 "Hello"
```

```
# Check GPU utilization
watch -n 1 rocm-smi
```

Common Issues and Fixes

“No GPU detected”

- Verify ROCm installation: `rocminfo` should show your GPU
- Check group membership: user must be in `render` and `video` groups
- Try: `HSA_OVERRIDE_GFX_VERSION=11.0.0` for unsupported cards

Slow performance

- Ensure ROCm is being used, not Vulkan fallback
- Check `rocm-smi` for GPU utilization during inference
- Try different llama.cpp builds or Ollama versions

Memory errors

- Reduce model size or quantization level
- Close other GPU-using applications
- Check `rocm-smi` for actual VRAM usage

The Verdict

Recommendation Matrix

You Are…	Recommendation
Windows user	NVIDIA
Linux beginner	NVIDIA (easier start)
Linux power user, budget-conscious	AMD 7900 XTX/XT
Need maximum VRAM under $700	AMD 7900 XT
Want zero friction	NVIDIA
Enjoy tinkering	AMD

You Are…	Recommendation
Image generation focus	NVIDIA
LLM inference only	Either (AMD good value)

The Honest Tradeoff

AMD gives you: More VRAM per dollar, open-source friendly stack, competitive performance on optimized workloads, 85-95% of NVIDIA speed at 60-70% of the price.

AMD costs you: Linux requirement for best experience, occasional troubleshooting, slower support for new models/tools, some software incompatibility.

The bottom line: ROCm is finally ready—for the right user. If you're on Linux and don't mind occasional friction, the RX 7900 XTX and 7900 XT are excellent values. If you want things to just work, NVIDIA remains the safer choice.

The gap is closing. But it hasn't closed yet.

Related Guides

- GPU Buying Guide for Local AI
- Used RTX 3090 Buying Guide for Local AI
- How Much VRAM Do You Need for Local LLMs?

Source: https://insiderllm.com/guides/amd-vs-nvidia-local-ai-rocm/

Free guides for running AI locally