RTX 5090 vs DGX Spark vs AMD: The Ultimate Local LLM Benchmark (2026)

March 25, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** The RTX 5090 (32GB GDDR7, 1,792 GB/s) is the fastest single GPU for local LLM inference in 2026. It generates 186 tok/s on Qwen3 8B Q4_K_M and 124 tok/s on 14B -- roughly 30-50% faster than the RTX 4090 depending on context length. The DGX Spark (128GB, $3,000+) runs 70B-120B models that no single GPU can fit, but only manages 38 tok/s on a 120B model because its 273 GB/s bandwidth is a bottleneck. AMD's Strix Halo with 128GB unified memory handles Qwen3.5 122B-A10B at 9.5 tok/s -- slow, but it runs. For most people: a used RTX 3090 ($800-900) remains the best value. The 5090 ($2,000+) is the speed king if you can find one at MSRP.

More on this topic: GPU Buying Guide | VRAM Requirements | ROCm vs CUDA | Mac M-Series Guide | GB10 Boxes Compared

The RTX 5090 has been out long enough for the llama.cpp community to get real numbers. Not marketing slides. Not synthetic benchmarks. Actual tok/s from llama-bench running real models at real context lengths.

The verdict? 32GB of GDDR7 at 1,792 GB/s changes the game for single-GPU inference. But the 5090 isn’t the only new hardware worth benchmarking. NVIDIA’s DGX Spark brings 128GB unified memory to a desktop box. AMD’s Strix Halo puts 128GB unified memory in a mini PC. And the Radeon AI PRO R9700 is an oddball 32GB card running Vulkan that nobody expected to be competitive.

Here’s every number that matters, from every platform, tested with the same llama.cpp builds.

The hardware

Hardware	Memory	Bandwidth	Bus	Price (2026)
RTX 5090	32 GB GDDR7	1,792 GB/s	512-bit	$2,000 MSRP (~$2,600 street)
RTX 4090	24 GB GDDR6X	1,008 GB/s		$1,600-1,800 used

Hardware	Memory	Bandwidth	Bus	Price (2026)
			384-bit	
RTX 3090	24 GB GDDR6X	936 GB/s	384-bit	$800-900 used
DGX Spark	128 GB LPDDR5x (unified)	273 GB/s	–	~$3,000
Strix Halo (AI Max+ 395)	128 GB LPDDR5x (unified)	256 GB/s (~212 measured)	256-bit	$2,000-3,000 (mini PCs)
Radeon AI PRO R9700	32 GB GDDR6	645 GB/s	256-bit	~$1,100

The bandwidth column tells most of the story. LLM token generation is bandwidth-bound – more GB/s generally means more tok/s. The RTX 5090's 1,792 GB/s is 1.78x the 4090's 1,008 GB/s. That advantage shows up in every single benchmark below.

But bandwidth isn't everything. The DGX Spark and Strix Halo trade raw speed for capacity: 128GB lets you run 70B+ models without quantizing them into oblivion. Different tools for different jobs.

RTX 5090: the single-GPU king

Token generation benchmarks

Tested with llama.cpp (Q4_K_M quantization unless noted):

Model	VRAM Used	4K ctx (tok/s)	8K ctx (tok/s)	32K ctx (tok/s)
Qwen3 8B	4.78 GB	185.9	169.8	111.9
Qwen3 14B	8.53 GB	123.8	115.5	82.4
Qwen3 MoE 30B-A3B	16.47 GB	234.3	170.5	110.7
Qwen3 32B	18.64 GB	61.4	55.5	43.8
gpt-oss 20B	–	–	–	–

The MoE number stands out: 234 tok/s on a 30B-parameter model because only 3B parameters are active per token. That's faster than the dense 8B model. MoE architectures are the RTX 5090's best friend – you get big-model quality at small-model speeds.

Prompt processing (prefill)

This is where the 5090's 21,760 CUDA cores flex:

Model	4K ctx (tok/s)	8K ctx (tok/s)	32K ctx (tok/s)	65K ctx (tok/s)
Qwen3 8B	10,407	8,745	3,688	2,212
Qwen3 14B	6,498	5,594	2,908	1,707
Qwen3 MoE 30B-A3B	6,630	5,799	2,878	1,512
Qwen3 32B	2,931	2,530	1,451	–

10,000+ tok/s prompt processing on an 8B model. That means a 4,000-token system prompt processes in under half a second. RAG workflows and long-context applications see massive gains from Blackwell's compute.

Extreme context (131K+ tokens)

Push the 5090 to its 32GB limit:

Model	VRAM	Context	PP 2048 (tok/s)	TG 128 (tok/s)
Qwen3 8B	23 GB	131K	948	49.4
Qwen3 14B	31 GB	131K	908	37.2
Qwen3 MoE 30B-A3B	31 GB	147K	666	52.3

Generation speed drops to 49 tok/s at 131K context on the 8B model – down from 186 at 4K. That's the KV cache eating VRAM and bandwidth. Still usable. Still faster than reading speed. But context length has a real cost.

RTX 5090 vs RTX 4090: is the upgrade worth it?

Real-world comparison using Ollama and LM Studio:

Model	Quant	RTX 4090	RTX 5090	Speedup
Llama 3.1 70B	Q4_K_M, 2K ctx	28.3 tok/s	36.7 tok/s	+30%
Llama 3.1 70B	Q4_K_M, 32K ctx	11.2 tok/s	16.8 tok/s	+50%
Mixtral 8x7B	Q5_K_M, 2K ctx	47.1 tok/s	58.4 tok/s	+24%
Qwen 2.5 32B	Q6_K, 8K ctx	39.4 tok/s	51.8 tok/s	+31%
Qwen 2.5 32B	Q6_K, 32K ctx	18.7 tok/s	26.3 tok/s	+41%
DeepSeek Coder 33B	Q5_K_M, 2K ctx	42.8 tok/s	54.1 tok/s	+26%

The pattern: 24-50% faster, with the gap widening at longer context lengths. This makes sense – at short contexts, you're compute-bound during generation and the bandwidth advantage matters less. At 32K context, the KV cache is large enough that bandwidth dominance kicks in.

The 5090 also gets 8GB more VRAM (32 vs 24). That's the difference between running Qwen 3.5 27B at Q4 with a comfortable context window versus running it tight on VRAM.

Is the upgrade from a 4090 worth it? Not at $2,600 street price. The 30% average speedup doesn't justify paying $1,000+ more than a used 4090. If you're buying fresh, the 5090 is the obvious pick. If you already have a 4090, keep it.

DGX Spark: 128GB for desktop inference

The DGX Spark is NVIDIA's GB10 Grace Blackwell chip in a small form factor. Its pitch: run models that won't fit in any single GPU's VRAM.

DGX Spark benchmarks

gpt-oss 120B (MXFP4 quantization):

Context Depth	PP 2048 (tok/s)	TG 32 (tok/s)
Baseline (0)	1,956	60.6
4K	1,637	54.1
8K	1,512	51.5
16K	1,307	47.5
32K	1,027	40.6

Cross-model comparison:

Model	PP (tok/s)	TG (tok/s)
gpt-oss 20B MXFP4	3,622	59.0
gpt-oss 120B MXFP4	1,723	38.6
Qwen3 Coder 30B Q8_0	2,916	47.1

The prefill numbers are strong – nearly 2,000 tok/s on a 120B model. That's the Blackwell tensor cores doing work. But generation at 38.6 tok/s reveals the bottleneck: 273 GB/s bandwidth. For comparison, the RTX 5090 has 6.6x more bandwidth.

DGX Spark vs the field (gpt-oss 120B)

Hardware	PP (tok/s)	TG (tok/s)
DGX Spark (128GB)	1,723	38.6
Strix Halo 128GB	340	34.1
Apple M3 Ultra 256GB	864	70.8
3x RTX 3090 (72GB total)	1,642	124.0

The 3x RTX 3090 setup destroys everything on generation speed – 124 tok/s versus the Spark's 38.6. Three used 3090s costs roughly $2,700, comparable to the DGX Spark's price. The tradeoff: three GPUs need a big case, a 1200W+ PSU, and motherboard with enough PCIe slots. The Spark fits on your desk.

The Apple M3 Ultra is interesting too – 70.8 tok/s generation from 819 GB/s unified bandwidth. But a 256GB M3 Ultra runs $7,000+.

Should you buy a DGX Spark? Only if you specifically need to run 70B-120B models in a quiet, compact form factor and you're OK with 40-60 tok/s generation. If raw speed matters, multi-GPU discrete setups are faster and comparably priced. If you want unified memory without the noise, a Mac Studio M4 Max 128GB gives better bandwidth per dollar.

AMD: ROCm, Vulkan, and the unified memory play

AMD has two stories in 2026: discrete GPUs with ROCm/Vulkan, and Strix Halo's unified memory.

Radeon AI PRO R9700 vs RTX 5090

The R9700 is a 32GB GDDR6 card with 645 GB/s bandwidth. Head-to-head on Qwen3.5 35B-A3B (Q4_K_XL):

Prompt processing:

Context	RTX 5090 (CUDA)	R9700 (Vulkan)	5090 Advantage
512	7,026 tok/s	2,713 tok/s	2.6x
2,048	6,960 tok/s	2,610 tok/s	2.7x
8,192	6,835 tok/s	2,413 tok/s	2.8x
32,768	6,461 tok/s	1,877 tok/s	3.4x

Token generation: RTX 5090 gets 194 tok/s, R9700 gets 127 tok/s (1.53x gap).

The generation gap (1.5x) is much smaller than the prefill gap (2.6-3.4x). That's because generation is bandwidth-bound, and the bandwidth ratio (1,792 / 645 = 2.8x) gets partially offset by software optimization differences. For prompt processing, the 5090's Blackwell tensor cores and CUDA compute kernels create a wider gap.

At $1,100 vs $2,000+, the R9700 delivers 65% of the 5090's generation speed at 55% of the price. Not bad for an AMD card running Vulkan.

Strix Halo: 128GB in a laptop chip

The Ryzen AI Max+ 395 puts 128GB LPDDR5x unified memory in a chip that draws under 120W. The backend story is messy:

Backend	Llama 2 7B Q4_0 pp512	Llama 2 7B Q4_0 tg128
CPU only	295 tok/s	29.0 tok/s
HIP (ROCm)	349 tok/s	48.7 tok/s
HIP + WMMA + FA	344 tok/s	50.9 tok/s
Vulkan	882 tok/s	52.2 tok/s
Vulkan + FA	884 tok/s	52.7 tok/s

Vulkan beats ROCm HIP by 2.5x on prompt processing and edges it out on generation too. On Strix Halo, Vulkan is the backend you want – not HIP.

Larger models on Strix Halo (Vulkan):

Model	pp512 (tok/s)	tg128 (tok/s)
Qwen3 MoE 30B-A3B	119	75.3
Llama 4 Scout 109B (17B active)	103	20.2
70B Q4_K_M (HIP)	95	4.5

75 tok/s on a MoE model is genuinely usable. 4.5 tok/s on a dense 70B is not – that's a hardware limitation from 256 GB/s bandwidth trying to move a ~40GB model through memory every token.

The Strix Halo's value prop is running models that physically don't fit on any 24GB or 32GB GPU. If you need Qwen3.5 122B-A10B (reported at 9.5 tok/s in community benchmarks) and you're not buying a Mac, this is how you do it on a budget.

The full picture: generation speed comparison

Token generation at 4K context, Q4_K_M where applicable:

Hardware	7-8B Model	14B Model	30B MoE	32B Dense	70B+
RTX 5090	186 tok/s	124 tok/s	234 tok/s	61 tok/s	37* tok/s
RTX 4090	~143 tok/s	~95 tok/s	~180 tok/s	~47 tok/s	28 tok/s
RTX 3090	~112 tok/s	~75 tok/s	~140 tok/s	~37 tok/s	– (24GB limit)
DGX Spark	~59 tok/s	~47 tok/s	–	~38 tok/s	39 tok/s
Strix Halo	53 tok/s	~35 tok/s	75 tok/s	–	4.5 tok/s
R9700 (Vulkan)	~85 tok/s	~60 tok/s	127 tok/s	–	–

*70B on RTX 5090 requires heavy quantization or partial CPU offload; listed number is from Q4_K_M Ollama benchmarks.

The RTX 5090 wins everywhere a model fits in 32GB. The DGX Spark and Strix Halo win on capacity – they run models that physically won't fit on the other cards.

Who should buy what

Budget king: used RTX 3090 ($800-900)

24GB GDDR6X at 936 GB/s. Still runs 32B models at Q4_K_M. Gets 112 tok/s on 8B models. Nothing has dethroned this card at the price point. The main limitation is 24GB – Qwen 3.5 27B at Q4 with 32K context won't fit, and forget about 70B without multi-GPU.

Read: Used RTX 3090 Buying Guide

Speed king: RTX 5090 ($2,000 MSRP)

32GB GDDR7 at 1,792 GB/s. The fastest single GPU for local inference by a wide margin. 8GB more VRAM than the 4090 means you can run larger models or use longer context windows. The problem is availability – street prices are $2,600+ and stock is inconsistent. At MSRP, it's the clear 2026 pick. At $2,600, it's harder to justify over a used 4090 + pocketing $1,000.

Maximum capacity: DGX Spark or Mac Studio

If you need to run 70B-120B models without multi-GPU complexity, unified memory is the path. The DGX Spark ($3,000) gets you 128GB and Blackwell tensor cores. A Mac Studio M4 Max 128GB gets you 128GB with higher bandwidth (~546 GB/s on M4 Max) and a mature software stack. The Mac is the better value for pure inference if you don't need CUDA.

Read: GB10 Boxes Compared | Mac M-Series Guide

AMD wildcard: Strix Halo

128GB unified memory without the Apple tax. Vulkan backend has gotten surprisingly competitive in llama.cpp. The catch: 256 GB/s bandwidth means you're getting laptop-class generation speeds. You're paying for capacity, not speed. Makes sense for people who want to run huge models for research or experimentation and don't need fast interactive speeds.

Read: ROCm vs CUDA in 2026

The R9700 surprise

At $1,100, 32GB GDDR6 and 127 tok/s on MoE models via Vulkan is genuinely competitive. If you're on a budget, don't need NVIDIA's ecosystem, and primarily run MoE models (which are increasingly the best architectures for local AI), the R9700 is worth a look. Vulkan support in llama.cpp has improved dramatically.

What about NVFP4?

The RTX 5090's Blackwell tensor cores natively support NVFP4 (4-bit floating point). This matters more for TensorRT-LLM and vLLM than for llama.cpp, which uses its own GGML quantization formats (Q4_K_M, Q5_K_S, etc.).

In practice: llama.cpp users won't see NVFP4-specific gains. The existing Q4 and Q5 formats already achieve similar compression ratios with good quality. Where NVFP4 matters is in production serving frameworks where you want maximum throughput on large batches – not the single-user, single-request workflow most local LLM users care about.

The bandwidth rule of thumb

Every hardware platform roughly follows this formula for token generation:

tok/s = (bandwidth in GB/s) x (efficiency factor) / (model size in GB)

NVIDIA CUDA gets an efficiency factor around 0.12-0.14. AMD Vulkan gets about 0.08-0.10. Unified memory platforms (DGX Spark, Strix Halo, Apple Silicon) get 0.06-0.10 depending on the memory controller.

This means you can roughly predict any platform's performance:

- RTX 5090 on a 5GB model: 1,792 x 0.13 / 5 = ~46.6 tok/s per GB… which at Q4_K_M Qwen3 8B (~4.8GB) gives ~48.5 x 4 = ~194 tok/s. Close to the measured 186.
- DGX Spark on a 60GB model: 273 x 0.08 / 60 = ~0.36 tok/s per GB… x 60 = ~22 tok/s. Real-world is higher due to compute optimizations, but the ballpark holds.

Bandwidth is destiny for local LLM inference. Everything else is optimization on top.

Bottom line

The RTX 5090 is the best single GPU for local AI in 2026. Period. 32GB GDDR7 at 1,792 GB/s gives you both the capacity and the speed. But "best" and "best value" aren't the same thing.

A used RTX 3090 at $800 gives you 62% of the 5090's bandwidth for 40% of the price. Two of them give you 48GB VRAM and more total bandwidth than a single 5090.

The DGX Spark and Strix Halo are capacity plays, not speed plays. Buy them when the model won't fit anywhere else and you don't want to manage multi-GPU.

The actual answer for most people hasn't changed: figure out which models you want to run, check how much VRAM they need, and buy the cheapest card that fits. The benchmarks above tell you exactly how fast each option will be.

Get notified when we publish new guides.

Subscribe — free, no spam

Source: https://insiderllm.com/guides/rtx-5090-local-ai-benchmarks/

Free guides for running AI locally