Llama 3 Guide: Every Size from 1B to 405B

February 2, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** Llama 3.3 70B is the flagship — it matches Llama 3.1 405B performance at a fraction of the hardware cost, with major gains in math (+9.2 on MATH) and instruction following (92.1 on IFEval). But you need ~43 GB VRAM to run it at Q4, so it's a dual-GPU or CPU-offload setup for most people. For 8 GB GPUs, Llama 3.1 8B at Q4_K_M is still solid but Qwen 3 8B now beats it on most benchmarks. The 1B and 3B models are for edge devices and quick tasks only. The Llama 3.2 Vision 11B is the easiest way to run a multimodal model locally — pull it in Ollama and start describing images. Overall, Llama 3 has the largest fine-tune ecosystem of any open model family, but Qwen 3 has overtaken it on raw benchmarks at most sizes.

📚 **More on this topic:** Qwen Models Guide · DeepSeek Models Guide · Mistral & Mixtral Guide · VRAM Requirements

Meta's Llama 3 is the most recognizable name in open-weight AI. It's the model most people start with, the base for thousands of community fine-tunes, and the reason "run your own LLM" became a mainstream idea.

But the naming is a mess. Llama 3.1, 3.2, 3.3 — they're not sequential upgrades. They're different model families released for different purposes, and picking the wrong one wastes your VRAM on a worse model. This guide cuts through it.

The Llama 3 Family at a Glance

Model	Parameters	Release	Context	Type	Ollama Command
Llama 3.2 1B	1.24B	Sep 2024	128K	Text only	`ollama pull llama3.2:1b`
Llama 3.2 3B	3.21B	Sep 2024	128K	Text only	`ollama pull llama3.2:3b`
Llama 3.1 8B	8.03B	Jul 2024	128K	Text only	`ollama pull llama3.1:8b`
Llama 3.2 Vision 11B	10.67B	Sep 2024	128K	Text + Images	`ollama pull llama3.2-vision:11b`

Model	Parameters	Release	Context	Type	Ollama Command
Llama 3.3 70B	70.6B	Dec 2024	128K	Text only	`ollama pull llama3.3:70b`
Llama 3.2 Vision 90B	88.8B	Sep 2024	128K	Text + Images	`ollama pull llama3.2-vision:90b`
Llama 3.1 405B	405B	Jul 2024	128K	Text only	Too large for most local setups

All models support 128K context. All have instruct-tuned versions (the ones you want for chat). The base/pretrained versions exist too but are only useful for fine-tuning.

Which Version? 3.1 vs 3.2 vs 3.3

This is where people get confused. The version numbers don't mean "3.3 is always better than 3.2."

Llama 3.1 (July 2024) — The big launch. 8B, 70B, and 405B text models. Extended context to 128K tokens. Multilingual support for 8 languages. The 8B is still the one to use at that size.

Llama 3.2 (September 2024) — Two separate things packaged under one name:

- **Small text models** (1B, 3B) built for edge devices and mobile. Not downsized 8B models — they're distinct architectures using Grouped-Query Attention, optimized to be tiny.
- **Vision models** (11B, 90B) that can process images. Built by adding vision adapters to the Llama 3.1 text backbone. The 11B runs on 8 GB VRAM.

Llama 3.3 (December 2024) — One model only: 70B text. Trained to match 3.1 405B performance at 70B size. Massive improvements over 3.1 70B in math (+9.2 on MATH), instruction following (92.1 IFEval, beating 405B), and multilingual (+4.2 on MGSM). **This is the 70B you should use** — 3.1 70B is obsolete.

The version you actually want

Size	Use This	Not This
1B or 3B	Llama 3.2	(only option)
8B	Llama 3.1	(only option)
11B Vision	Llama 3.2 Vision	(only option)

Size	Use This	Not This
70B	**Llama 3.3**	~~Llama 3.1 70B~~
405B	Llama 3.1	(only option, but consider the API instead)

Every Size, Honestly

Llama 3.2 1B — Edge Device Territory

VRAM: ~1-2 GB (Q4), ~3 GB (FP16) **Ollama:** `ollama pull llama3.2:1b`

This is for phones, tablets, and embedded systems. It can summarize short text, do simple instruction following, and handle basic classification. It cannot hold a real conversation or do anything requiring reasoning.

Competitive with Gemma 2 at this size. Useful if you need something that runs on almost anything, including a Raspberry Pi with enough RAM. Not useful for much else.

Llama 3.2 3B — Lightweight but Limited

VRAM: ~2-3 GB (Q4_K_M), ~6-7 GB (FP16) **Ollama:** `ollama pull llama3.2:3b`

The 3B can handle short conversations, simple Q&A, summarization, and basic tool calling. Meta designed it for on-device use and it shows — it outperforms Gemma 2 2.6B and Phi 3.5-mini on instruction following and summarization.

Supports 8 languages (English, German, French, Italian, Portuguese, Hindi, Spanish, Thai). The 128K context window is supported at full precision, though quantized versions typically cap at 8K.

Good for: a fast assistant on your laptop that doesn't eat your GPU, preprocessing/routing tasks, or situations where you need an answer in milliseconds and don't care if it's occasionally wrong.

Not good for: coding, complex reasoning, creative writing, or anything you'd trust with an important decision.

Llama 3.1 8B — The Workhorse

VRAM: ~5 GB (Q4_K_M), ~8.5 GB (Q8_0), ~16 GB (FP16) **Ollama:** `ollama pull llama3.1:8b`

This is where Llama gets genuinely useful. The 8B handles conversation, coding assistance, writing, summarization, and light reasoning well enough for daily use. It runs comfortably on any modern GPU with 8 GB VRAM at Q4_K_M.

The benchmarks tell the story: it outperforms every open model from 2023 regardless of size, including the original Llama 2 70B. For a model that fits on a single consumer GPU, that's remarkable.

The honest comparison: Qwen 3 8B now beats Llama 3.1 8B on most benchmarks, especially math, reasoning, and multilingual tasks. Qwen 3 also has a thinking mode toggle that Llama lacks. If you're choosing purely on capability, Qwen 3 8B wins in early 2026.

So why use Llama 3.1 8B at all? Three reasons:

1. **Fine-tune ecosystem.** Thousands of Llama 3 fine-tunes exist. Dolphin, Hermes, WizardLM — the most popular community models are Llama-based. Qwen fine-tunes exist but the selection is smaller.
2. **Speed.** Llama 3.1 8B is slightly faster at inference than Qwen 3 8B, particularly when Qwen's thinking mode is active.
3. **Compatibility.** Every tool, framework, and tutorial supports Llama. It's the default model in almost every getting-started guide.

Llama 3.3 70B — The Flagship

VRAM: ~43 GB (Q4_K_M), ~70 GB (Q8_0), ~140 GB (FP16) **Ollama:** `ollama pull llama3.3:70b`

This is the model that matters. Llama 3.3 70B was trained to match the 405B's performance, and on several benchmarks it actually exceeds it:

Benchmark	Llama 3.1 70B	Llama 3.3 70B	Llama 3.1 405B
MATH (0-shot)	67.8	**77.0**	73.8
IFEval	—	**92.1**	88.6
HumanEval	—	88.4	89.0
MMLU (0-shot)	86.0	86.0	88.6
MGSM (multilingual)	86.9	**91.1**	—
GPQA Diamond	48.0	50.5	—

The math improvement (+9.2 over 3.1 70B) is the biggest jump. Instruction following at 92.1 beats every model in its class including 405B. Multilingual went from good to excellent.

The hardware reality: 43 GB at Q4_K_M means no single consumer GPU can run it comfortably. Your options:

- **Dual RTX 3090s** (48 GB total) — tight but works with short context
- **Single RTX A6000** (48 GB) or similar workstation card
- **CPU offload** with 64+ GB system RAM — slow (2-5 tok/s) but functional. Pull the model in Ollama and it handles offloading automatically
- **Pure CPU** with 64 GB+ RAM — expect 1-3 tok/s depending on your processor

For more on running large models, see our 24 GB VRAM guide.

Llama 3.1 405B — Impractical Locally

VRAM: ~230+ GB (Q4_K_M), ~405 GB (Q8_0) **Ollama:** Technically `ollama pull llama3.1:405b` but you almost certainly can't run it

Unless you have a rack of GPUs or a machine with 512+ GB of RAM (for pure CPU inference at glacial speeds), this isn't a local model. Use it via API providers like Together AI, Fireworks, or Groq.

And honestly, Llama 3.3 70B makes 405B mostly redundant. The 70B matches or beats it on several benchmarks at 1/6th the hardware cost. The 405B's advantage shows only on narrow edge cases and the hardest reasoning tasks.

VRAM Requirements

Text Models

Model	Q4_K_M	Q5_K_M	Q8_0	FP16
1B	~1 GB	~1.2 GB	~1.5 GB	~2.5 GB
3B	~2.5 GB	~3 GB	~4 GB	~6.5 GB
8B	~5 GB	~5.5 GB	~8.5 GB	~16 GB
70B	~43 GB	~48 GB	~70 GB	~140 GB
405B	~230 GB	~260 GB	~405 GB	~810 GB

These are weights only. Add 1-2 GB for KV cache at default context lengths, more for longer context. A 70B model at 32K context adds ~14 GB for KV cache alone. At 128K, the KV cache exceeds 40 GB.

Tip: Set `OLLAMA_KV_CACHE_TYPE=q8_0` to halve KV cache memory with minimal quality loss. This is especially valuable for the 70B model.

Vision Models

Model	Q4_K_M	FP16
11B Vision	~6-7 GB	~22 GB
90B Vision	~50+ GB	~180 GB

The 11B Vision model is the practical choice. It fits on 8 GB GPUs at Q4 and handles image understanding, OCR, chart reading, and visual Q&A.

Which GPU for Which Model

GPU	VRAM	Best Llama Model
GTX 1660 / RTX 3060	6 GB	3B (Q4_K_M), 8B (Q3-Q4 tight)
RTX 3060 12GB / 4060 Ti 16GB	12-16 GB	8B (Q8_0), 11B Vision (Q4)
RTX 3090 / 4090	24 GB	8B (FP16), 11B Vision (Q8)
2x RTX 3090 / RTX A6000	48 GB	70B (Q4_K_M, short context)
4x RTX 3090 / 2x A6000	96 GB	70B (Q8_0 with headroom)

→ Use our Planning Tool to check exact VRAM for your setup.

Llama 3 vs the Competition (Early 2026)

The open model landscape has shifted since Llama 3's releases. Here's where things stand.

At 8B: Llama 3.1 vs Qwen 3 vs Gemma 3

	Llama 3.1 8B	Qwen 3 8B	Gemma 3 12B
Math	Good	**Best**	Good

	Llama 3.1 8B	Qwen 3 8B	Gemma 3 12B
Coding	Good	**Best**	Good
Reasoning	Good	**Best** (thinking mode)	Better
Multilingual	8 languages	**29+ languages**	Good
Speed	**Fastest**	Slightly slower (thinking mode)	Moderate
Fine-tune ecosystem	**Largest**	Growing	Smaller
Vision	No (separate 11B)	Qwen 3 VL available	Yes (built-in)

Verdict at 8B: Qwen 3 8B wins on benchmarks. Llama 3.1 8B wins on ecosystem and compatibility. If you want the best model out of the box, pull Qwen 3. If you want the most fine-tune options or need a specific community model, Llama is still the base to build on.

At 70B: Llama 3.3 vs Qwen 2.5 72B vs DeepSeek R1

	Llama 3.3 70B	Qwen 2.5 72B	DeepSeek R1 Distill 70B
General knowledge	**Strong**	**Strong**	Good
Math / Reasoning	Good (77.0 MATH)	Good	**Best** (94.5 MATH-500)
Coding	Strong (88.4 HumanEval)	Strong	**Best** (57.5 LiveCodeBench)
Instruction following	**Best** (92.1 IFEval)	Good	Weaker
Speed	**Fastest**	Comparable	Slower (thinking tokens)
Multilingual	Strong (91.1 MGSM)	**Best**	Moderate

Verdict at 70B: It depends on your task. Llama 3.3 is the best general-purpose 70B — fast, great at following instructions, strong across the board. DeepSeek R1 Distill 70B crushes it on math and reasoning but is slower and narrower. Qwen 2.5 72B is comparable to Llama 3.3 overall. For pure reasoning, see our DeepSeek guide.

March 2026 Update: Enter Qwen 3.5

Qwen 3.5 landed in February-March 2026 with a new Gated DeltaNet architecture, native multimodal (text + images + video from the same weights), and 262K context. It widens the gap that Qwen 3 already opened.

At 8GB: Llama 3.1 8B vs Qwen 3.5 9B. The Qwen 3.5 9B scores 81.7 on GPQA Diamond versus roughly 49 for Llama 3.1 8B. It's natively multimodal (paste a screenshot, it reads it), has 262K

context versus 128K, and fits in ~5GB at Q4. Llama 3.1 8B still loads faster, has more fine-tunes, and doesn't need Ollama 0.17.4+. But on raw capability, the gap is now large.

At 24GB: Llama 3.3 70B (offloaded) vs Qwen 3.5 27B/35B-A3B. Llama 3.3 70B can't fit on a single 24GB card without CPU offload, which drops you to 2-5 tok/s. Qwen 3.5 gives you two options that run entirely on-GPU:

	Llama 3.3 70B (offloaded)	Qwen 3.5 27B	Qwen 3.5 35B-A3B
VRAM needed	~43 GB (spills to RAM)	~16 GB Q4	~20 GB Q4
Speed on 24GB card	2-5 tok/s (offload)	25-40 tok/s	~112 tok/s (3090)
SWE-bench Verified	—	72.4%	69.2%
Multimodal	No	Yes (native)	Yes (native)
Context	128K	262K	262K

If you have a single 24GB card, Qwen 3.5 27B or 35B-A3B gives you dramatically better throughput than an offloaded Llama 70B. The Llama 70B is still the stronger model on general knowledge and instruction following if you have the hardware to run it properly (dual 24GB GPUs or 48GB+).

Where Llama still wins:

- Fine-tune ecosystem (Dolphin, Hermes, WizardLM, thousands of community models)
- Tool compatibility (every framework tests Llama first)
- Meta's continued investment and Llama 4's MoE direction
- If you need an uncensored or specialized fine-tune, it's almost certainly Llama-based

Llama isn't losing relevance — the options just got wider. A year ago, the 8GB choice was "which 8B model." Now it's "do I want the model with better benchmarks (Qwen 3.5 9B) or the model with 10x more fine-tunes (Llama 3.1 8B)?" Both are valid.

The Bigger Picture

Qwen 3 and now Qwen 3.5 (from Alibaba) have overtaken Llama as the leading open model family on raw benchmarks. But Llama retains three major advantages:

1. **The fine-tune ecosystem.** More community fine-tunes exist for Llama than all other families combined. Dolphin, Hermes, WizardLM, Dark Champion — if you want an uncensored, creative, or specialized model, it's probably Llama-based.

2. **Licensing simplicity.** Meta's Llama license is permissive for commercial use (under 700M monthly users). Qwen uses Apache 2.0, which is also fine, but Llama's ecosystem maturity matters.
3. **Tool compatibility.** Every local AI tool is tested against Llama first. When something breaks, Llama gets fixed first.

Multimodal: Llama 3.2 Vision

Llama 3.2 Vision is the easiest way to run a multimodal model locally. The 11B version handles images alongside text — describe photos, read charts, extract text from documents, answer questions about diagrams.

Setup

```
ollama pull llama3.2-vision:11b
```

Using It

Command line:

```
ollama run llama3.2-vision:11b "Describe this image" --image ./photo.jpg
```

From the API:

```
import ollama

response = ollama.chat(
    model='llama3.2-vision:11b',
    messages=[{
        'role': 'user',
        'content': 'What does this chart show?',
        'images': ['./chart.png']
    }]
```

```
)
print(response['message']['content'])
```

What It Can Do

- **Image captioning** — describe what's in a photo
- **OCR** — read text from images, receipts, screenshots
- **Chart/graph analysis** — interpret data visualizations
- **Visual Q&A** — answer specific questions about image content
- **Document understanding** — extract information from scanned pages

Limitations

- Single image per prompt (no multi-image comparison)
- No video support
- The 11B is noticeably less capable than the 90B on complex visual reasoning
- Can hallucinate details in images, especially small text or crowded scenes
- Knowledge cutoff means it won't recognize very recent products or people

VRAM

The 11B Vision fits on 8 GB at Q4 quantization. At default precision, expect ~11 GB. The 90B requires 64+ GB — impractical for most local setups.

The Fine-Tune Ecosystem

This is Llama's killer advantage. No other model family has as many community fine-tunes available in Ollama and on Hugging Face.

Worth Knowing About

Fine-Tune	Base	Best For	Ollama
Dolphin 3.0	Llama 3	General purpose, uncensored, function calling	`ollama pull dolphin-llama3`
Nous Hermes 3	Llama 3.2 8B	Creative writing, roleplay, long-form	`ollama pull hermes3`

Fine-Tune	Base	Best For	Ollama
Nemotron	Llama 3.1 70B	NVIDIA-tuned, strong benchmarks	`ollama pull nemotron`

Dolphin (by Eric Hartford) is the go-to uncensored Llama model. Trained on diverse datasets, it removes refusals while maintaining quality. Dolphin 3.0 emphasizes precision and function calling.

Nous Hermes 3 from NousResearch is consistently praised as the best creative writing model in the open-source space. Uses ChatML format for structured conversations. If you want fiction, roleplay, or imaginative brainstorming, this is the model.

Nemotron is NVIDIA's fine-tune of Llama 3.1 70B, optimized for instruction following. It benchmarks well but requires the same 70B hardware.

Fine-Tuning Your Own

If you want to create a custom Llama 3 fine-tune, the most popular tools in 2026:

- **Unsloth** — 2x faster, 70% less VRAM than standard fine-tuning. Supports LoRA and QLoRA. The easiest starting point.
- **Axolotl** — More configurable, supports RLHF/RLVR, good for advanced users.
- **Hugging Face TRL + PEFT** — The standard library approach, most tutorials available.

You can fine-tune Llama 3.1 8B on a single 16 GB GPU with QLoRA. The 70B needs at least 48 GB for LoRA fine-tuning. For a deep dive, our fine-tuning guide is coming soon.

Setup Guide

Basic: Pull and Run

```
# Install Ollama if you haven't
# https://ollama.com/download

# The 8B — fits on any modern GPU
ollama pull llama3.1:8b
ollama run llama3.1:8b

# The 3B — for lightweight tasks
```

```
ollama pull llama3.2:3b

# The 70B — needs 48+ GB VRAM or CPU offload with 64 GB RAM
ollama pull llama3.3:70b

# Vision — image understanding
ollama pull llama3.2-vision:11b
```

Custom Modelfile

Create a file called `Modelfile`:

```
FROM llama3.1:8b

PARAMETER temperature 0.7
PARAMETER num_ctx 8192

SYSTEM """You are a helpful coding assistant. Be concise. Show code examples when relevant. If yo
```

Then:

```
ollama create my-coding-assistant -f Modelfile
ollama run my-coding-assistant
```

Recommended Quantizations

Your VRAM	Model	Quantization
4 GB	3B	Q4_K_M
6 GB	8B	Q4_K_M (tight)
8 GB	8B	Q4_K_M (comfortable)
12 GB	8B	Q8_0 or 11B Vision Q4
16 GB	8B	FP16 (full quality)
24 GB	8B FP16 + 11B Vision Q8	Switch between them
48 GB+	70B	Q4_K_M

For how quantization affects quality, see our quantization guide.

Common Problems

"The 70B is too slow." If you don't have 48+ GB of VRAM, the model spills to system RAM. That's a 10-30x slowdown. Check with `ollama ps` — if "GPU%" is below 100%, you're offloading. Options: use Q3_K_M instead of Q4_K_M, reduce context with `num_ctx 4096`, or use the 8B instead.

"Which Llama 3 should I pull?"

- 8B text: `llama3.1:8b`
- 70B text: `llama3.3:70b` (not llama3.1:70b)
- Small/edge: `llama3.2:3b` or `llama3.2:1b`
- Vision: `llama3.2-vision:11b`

"Context is too short." Default Ollama context is 2048 tokens. Override it: `ollama run llama3.1:8b --num-ctx 8192`. Or set it in your Modelfile with `PARAMETER num_ctx 8192`. Going above 32K on the 8B starts eating significant VRAM — each doubling of context roughly doubles the KV cache.

"My Llama 3 model refuses everything." The official instruct models have safety guardrails. For less restrictive behavior, use community fine-tunes like Dolphin (`ollama pull dolphin-llama3`) or Nous Hermes (`ollama pull hermes3`). See our uncensored models guide (coming soon) for details.

"Llama 3 keeps repeating itself." Lower the temperature (try 0.6-0.7) and increase `repeat_penalty` (try 1.1-1.2). This is more common at lower quantizations and with longer conversations. Using Q5_K_M instead of Q4_K_M can help.

"Should I use Llama 3 or Qwen 3?" For raw capability in 2026: Qwen 3 wins at most sizes. For the fine-tune ecosystem, compatibility, and community support: Llama 3. If you need a specific community model (Dolphin, Hermes, etc.), those are Llama-based. If you want the best out-of-the-box experience, try Qwen 3 first.

Bottom Line

Llama 3 isn't the benchmark leader it was in 2024. Qwen 3.5 has widened the gap that Qwen 3 opened, and DeepSeek R1 distills dominate for reasoning. But Llama remains the most

important open model family for three reasons: the fine-tune ecosystem is unmatched, tool compatibility is universal, and Meta keeps shipping (Llama 4 Scout brought MoE to the family).

The practical recommendations:

- **Tight on VRAM (4-8 GB)?** Pull `llama3.1:8b` at Q4_K_M. It's the reliable workhorse.
- **Want the best small model?** Try `qwen3:8b` first, fall back to `llama3.1:8b` if you need a specific Llama fine-tune.
- **Have 48+ GB VRAM?** `llama3.3:70b` is excellent for general use. Still one of the best 70B models available.
- **Need vision?** `llama3.2-vision:11b` is the easiest multimodal setup in local AI.
- **Need a specialized model?** Check if a Llama fine-tune exists for your use case — it probably does.

For choosing between all the model families, see our best models for chat guide and DeepSeek models guide.
