Qwen 3.5 for Local AI: Which Model, Which Quant, Which GPU

February 25, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** For most local users, Qwen3.5-35B-A3B is the pick. It's a 35B MoE with only 3B active parameters, so it runs at near-small-model speeds with large-model quality. On a 24GB GPU (RTX 4090/3090), run it at Q4_K_M. On 16GB (RTX 5060 Ti/5080), use Q4_K_L. The 27B dense model is the better coder (72.4 SWE-bench, matching GPT-5 mini) but needs Q4 on 24GB cards. If you have 48GB+ unified memory, the 122B-A10B matches GPT-5 mini across the board while beating it by 30% on tool use.

Qwen 3.5 dropped on February 24, 2026, and it changes the local AI math. Four models spanning 27B to 397B parameters, all Apache 2.0 licensed, all natively multimodal (text + image + video), and the MoE variants run faster than models a fraction of their size.

The 35B-A3B hits 194 tok/s on an RTX 5090. The 27B dense model scores 72.4 on SWE-bench Verified, matching GPT-5 mini. The 122B-A10B beats GPT-5 mini by 30% on tool-use benchmarks while running on a Mac Studio.

This is the complete guide to running them locally: which model fits your GPU, which quantization to pick, and where each one actually excels.

The lineup

Model	Total Params	Active Params	Architecture	Context	GGUF Q4 Size
Qwen3.5-27B	27B	27B (dense)	Dense + Hybrid Attention	262K	~17 GB
Qwen3.5-35B-A3B	35B	3B	MoE + Hybrid Attention	262K	~22 GB
Qwen3.5-122B-A10B	122B	10B	MoE + Hybrid Attention	262K	~70 GB
Qwen3.5-397B-A17B	397B	17B	MoE + Hybrid Attention	262K	~214 GB

All four models support 262K context natively (1M via YaRN extension), 201 languages, thinking/non-thinking modes, and multi-token prediction for speculative decoding. FP8 weights are available for every size.

The architecture is new. Qwen 3.5 uses a hybrid of Gated DeltaNet (linear attention) and full attention in a 3:1 ratio: three DeltaNet layers for every one full attention layer. The linear attention layers scale near-linearly with sequence length, which is why these models handle 256K context without the speed cliff you'd expect.

The 35B-A3B: the one most people should run

The 35B-A3B is the successor to the community favorite Qwen3-30B-A3B. It's a Mixture of Experts model with 35 billion total parameters but only 3 billion active per token. That means it runs at small-model speeds while drawing from large-model knowledge.

Real-world speeds:

GPU	Quantization	Token Generation	Prompt Processing
RTX 5090 (CUDA)	Q4_K_XL	194 tok/s	7,026 tok/s
AMD R9700 (Vulkan)	Q4_K_XL	127 tok/s	2,713 tok/s
DGX Spark	Q5 (UD-Q5_K_XL)	58.6 tok/s	1,861 tok/s
DGX Spark	Q8 (UD-Q8_K_XL)	35.9 tok/s	1,733 tok/s
Tesla V100 32GB	GGUF	38.4 tok/s	570 tok/s

194 tok/s on an RTX 5090. That's faster than most 7B models ran a year ago. Even the V100 (a card you can find used for $300-400) manages 38 tok/s.

The benchmarks are strong for a model this fast:

Benchmark	35B-A3B	GPT-5 mini	Claude Sonnet 4.5
MMLU-Pro	85.3	83.7	80.8
GPQA Diamond	84.2	82.8	80.1
SWE-bench Verified	69.2	72.0	62.0
BFCL-V4 (Tool Use)	67.3	55.5	54.8
BrowseComp	61.0	48.1	41.1

Benchmark	35B-A3B	GPT-5 mini	Claude Sonnet 4.5
TAU2-Bench (Agentic)	81.2	–	–

That BFCL-V4 score deserves attention. 67.3 vs GPT-5 mini's 55.5 on function calling and tool use. For anyone building local AI agents, this is the new default model to test.

The vision capabilities are also native. The 35B-A3B scores 81.4 on MMMU and 91.5 on MMBench, performing close to models 7x its size on visual benchmarks. You don't need a separate vision model anymore.

The caveat: the 35B-A3B scores lower than the 27B dense model on all coding benchmarks. SWE-bench 69.2 vs 72.4. LiveCodeBench 74.6 vs 80.7. Early community reports also flag that it produces broken diffs and hallucinates APIs on real repo work. If sustained coding is your primary use case, read the 27B section below.

The 27B dense: the coder's pick

The 27B is the dense model in the family, replacing the older Qwen3-32B. Every parameter is active on every token, which means slower generation but deeper reasoning per-token than the MoE variants.

Benchmark	27B	35B-A3B	GPT-5 mini
SWE-bench Verified	72.4	69.2	72.0
LiveCodeBench v6	80.7	74.6	80.5
Terminal-Bench 2	41.6	40.5	31.9
HMMT Feb 2025 (Math)	92.0	89.0	89.2

72.4 on SWE-bench Verified matches GPT-5 mini exactly. Terminal-Bench 2 at 41.6 crushes GPT-5 mini's 31.9. This is a competitive coding model at a size that fits on a single 24GB GPU at Q4.

At Q8 quantization, the 27B needs about 30GB. That puts it in A6000 (48GB) or Mac M-series territory for the highest quality quant. At Q4_K_M, it fits on a 4090 with room to spare at ~17GB.

After llama.cpp PR #19866 fixed multi-GPU graph splits, users report the 27B running across an RTX 3090 + RTX 3070 at over 700 tok/s prompt processing and 20+ tok/s generation using `-ts 85,15`. Multi-GPU setups are viable if you have the cards.

Pick the 27B over the 35B-A3B if your primary task is coding and you want the densest reasoning per token on a 24GB GPU. Pick the 35B-A3B if you need speed, tool use, or mixed workloads.

The 122B-A10B: the Mac Studio model

The 122B-A10B sits between the consumer models and the flagship. 122 billion total parameters, 10 billion active, built for machines with 48GB+ of unified memory.

Benchmark	122B-A10B	GPT-5 mini	Claude Sonnet 4.5
MMLU-Pro	86.7	83.7	80.8
SWE-bench Verified	72.0	72.0	62.0
Terminal-Bench 2	49.4	31.9	18.7
BFCL-V4 (Tool Use)	72.2	55.5	54.8
BrowseComp	63.8	48.1	41.1

Terminal-Bench 2 at 49.4 vs GPT-5 mini's 31.9 is not a close race. BFCL-V4 at 72.2 vs 55.5 is a 30% margin on tool use. If you have the hardware, this model outperforms GPT-5 mini on most tasks while running entirely on your machine.

At Q4 quantization, the 122B needs ~70GB. That means a Mac Studio with 96GB+ unified memory, a system with dual GPUs totaling 80GB+, or server hardware like the DGX Spark. Not consumer-friendly, but if you already have an M4 Max or Ultra Mac, this is the model that justifies the investment.

The 397B-A17B flagship

The flagship runs at 45 tok/s on 8xH100s with 8.6x faster decoding than Qwen3-Max. At Q4 quantization it needs ~214GB. That's Strix Halo 128GB territory (tight), Mac Ultra with 192GB+, or dedicated server hardware.

It beats GPT-5.2 on instruction following (IFBench 76.5 vs 75.4, the highest score of any model tested) and MultiChallenge (67.6 vs 57.9). It trails GPT-5.2 on AIME 2026 (91.3 vs 96.7) and SWE-bench (76.4 vs 80.0). Competitive with the best frontier models, but you need serious hardware.

For most local AI users, this is an aspirational model. The 122B or 35B-A3B cover 95% of use cases at a fraction of the hardware cost.

Which model on which GPU

Your Hardware	VRAM	Best Qwen 3.5 Model	Quantization	Expected Speed
RTX 3060 / 4060 (8GB)	8 GB	35B-A3B	Q2-Q3 (tight)	Usable but slow
RTX 3060 12GB	12 GB	35B-A3B	Q4_K_M	~30-40 tok/s
RTX 5060 Ti / 5080 (16GB)	16 GB	35B-A3B	Q4_K_L or Q4_K_XL	~40-60 tok/s
RTX 4090 / 3090 (24GB)	24 GB	27B at Q4 or 35B-A3B at Q8	Q4_K_M / Q8	20-60 tok/s
A6000 / dual GPU (48GB)	48 GB	27B at Q8 or 122B-A10B at Q4	Q8 / Q4_K_M	15-35 tok/s
Mac M4 Max (64GB)	64 GB	122B-A10B	Q4_K_M	Varies
Mac Ultra / Strix Halo (128GB+)	128 GB+	397B-A17B	Q4	Server-class

The 16GB tier is the sweet spot in 2026. The RTX 5060 Ti 16GB and RTX 5080 16GB both run the 35B-A3B at Q4 comfortably, and the MoE architecture means you're getting 35B-class knowledge from a model that only activates 3B parameters per token. Check exact VRAM figures with the VRAM Calculator and our VRAM requirements guide.

On a 4090 or 3090, you have a real choice. The 27B at Q4_K_M (~17GB) leaves room for other tools and gives you the strongest coding model in the family. The 35B-A3B at Q8 (~22GB, higher quality quant) gives you faster generation and better tool use. If you do both coding and agentic work, keep both in Ollama and swap between them.

Quantization: which quant matters

Unsloth's benchmarks on the 35B-A3B show real quality differences between quants:

Quantization	Top-1 Token Agreement	Notes
Q4_K_L	89%	Best quality retention at 4-bit

Quantization	Top-1 Token Agreement	Notes
MXFP4	Good (PPL +1.38)	New format, fast
UD-Q4_K_XL	79.4%	Lowest quality at 4-bit

Q4_K_L retains the highest quality. If your GPU can fit it, prefer Q4_K_L over Q4_K_XL.

For the 397B flagship, Unsloth reports UD-Q4_K_XL stays within 1 point of accuracy on most benchmarks despite reducing the file by ~500GB. At that scale, aggressive quantization hurts less.

Other findings worth knowing:

- 8-bit KV cache improves output quality when running 4-bit model quants
- Q3_K_XL reportedly beats Q4 on some benchmarks (Unsloth finding), though this needs broader validation
- FP8 weights are available officially for all sizes, giving you a clean middle ground between full precision and GGUF quants

How to run it

Ollama (simplest):

```
ollama run qwen3.5:35b
# or for the 27B:
ollama run qwen3.5:27b-q4_K_M
```

Requires Ollama v0.9.0 or higher. The multimodal support (images) works out of the box.

llama.cpp (most control):

```
./llama-server -hf Qwen/Qwen3.5-35B-A3B-GGUF:Q4_K_M \
  --jinja --reasoning-format deepseek -ngl 99
```

Build from source or use a release after b5092. If you need multi-GPU, make sure you have the PR #19866 fix (merged Feb 24) or a build from after that date.

Disable thinking by default (saves tokens on simple tasks):

```
./llama-server -hf Qwen/Qwen3.5-35B-A3B-GGUF:Q4_K_M \
  --jinja --chat-template-kwargs '{"enable_thinking": false}' -ngl 99
```

You can also add `/think` or `/nothink` to individual messages to toggle per-request.

Recommended sampling parameters (from Qwen):

- General thinking mode: temperature 1.0, top_p 0.95, top_k 20, presence_penalty 1.5
- Coding mode: temperature 0.6, top_p 0.95, top_k 20, presence_penalty 0.0
- Non-thinking instruct: temperature 0.7, top_p 0.8, top_k 20, presence_penalty 1.5

Known issues (day 2)

The models dropped 24 hours ago. Expect rough edges.

llama.cpp crashes on some multi-GPU configurations when running the 27B (issue #19860, illegal memory access on dual 3090s). PR #19866 (merged Feb 24) fixes most graph split issues, but build from latest main to be safe.

There's a 35% speed regression vs Qwen3 on CUDA. Issue #19894 shows the 35B-A3B at 38 tok/s on Tesla V100 vs the older 30B-A3B at 59 tok/s. CPU cores hit full load during generation, which suggests the DeltaNet architecture needs further CUDA optimization. This should improve in coming builds.

GGUF vision loading is broken (issue #19857, fails on vision projector weights). If you need multimodal now, use the HuggingFace weights with vLLM or SGLang.

The thinking model crashes llama-cli in some configurations (issue #19869). Workaround: `--chat-template-kwargs '{"enable_thinking": false}'`.

Some users report repetition/looping. Fix it with `--presence-penalty 1.5` (or up to 2.0).

Qwen 3.5 vs Qwen 3: what changed

	Qwen 3	Qwen 3.5
Dense model	32B	27B (denser, better benchmarks)
Small MoE	30B-A3B	35B-A3B (5B more total params)

	Qwen 3	Qwen 3.5
Medium MoE	–	122B-A10B (new tier)
Large MoE	235B-A22B	397B-A17B
Architecture	Standard attention	Hybrid DeltaNet + attention (3:1)
Multimodal	Separate VL models	Native in all models
Context	128K	262K (1M via YaRN)
FP8 weights	Community only	Official
Vocabulary	152K tokens	250K tokens

The 35B-A3B beats the previous flagship Qwen3-235B on language, vision, and agent benchmarks despite being about 7x smaller in total parameters. The architectural shift to DeltaNet is the reason: it scales better with context length and lets the MoE models pack more capability per active parameter.

The bottom line

The 35B-A3B is the model to start with. It runs on a 16GB GPU at Q4 and handles most workloads well. For coding work, the 27B dense model on a 24GB card is the stronger choice. For agent workflows with heavy tool use, the 122B-A10B is the best open-weight option available if you have the memory for it.

These models landed yesterday. The llama.cpp support is still stabilizing, the CUDA speed regression will get fixed, and the GGUF multimodal loading is broken. Give it a week before expecting a smooth experience on every backend. But the benchmarks are real, the architecture is new, and for a 16GB GPU owner running the 35B-A3B at Q4, this is the best local model available right now.

```
ollama run qwen3.5:35b
```

Try it.
