Qwen 3.5 Small Models: The 9B Beats Last-Gen 30B — Here's What Matters for Local AI

March 2, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** Qwen 3.5 just dropped 4 small models: 0.8B, 2B, 4B, and 9B. All natively multimodal (text, images, video from the same weights), all 262K context, all Apache 2.0. The 9B is the headline — it beats last-gen Qwen3-30B on GPQA Diamond (81.7 vs 77.2) and IFEval (91.5 vs 88.9) despite being one-third the size. It fits in 6.6GB on Ollama. If you have 8GB VRAM, run `ollama run qwen3.5:9b` right now. The 0.8B fits on a phone. The 4B is perfect for laptops. This is the most complete small model family anyone has shipped.

More on this topic: Qwen 3.5 Complete Local Guide | Qwen 3 Complete Guide | Best Local Coding Models 2026 | VRAM Requirements | Best Models Under 3B Parameters

Alibaba just completed the Qwen 3.5 family. Four new small models dropped today: 0.8B, 2B, 4B, and 9B. That brings the total to nine models from 0.8B to 397B, same Gated DeltaNet architecture across all of them, natively multimodal, Apache 2.0.

The 9B is the one that matters most for this audience. It beats Qwen3-30B on reasoning benchmarks despite being one-third the size. It fits in 6.6GB on Ollama. And it handles images and video from the same weights, no separate vision model needed.

If you've been running Qwen3-8B or Llama 3.1 8B as your default, those are last-gen now.

What dropped

Four dense models, all using the same architecture as the 397B flagship:

Model	Params	Layers	Ollama Size	Min RAM/VRAM	Architecture
Qwen 3.5-0.8B	0.8B	24	~1.0 GB	2 GB	Dense, Gated DeltaNet
Qwen 3.5-2B	2B	24	~2.7 GB	4 GB	Dense, Gated DeltaNet
Qwen 3.5-4B	4B	32	~3.4 GB	6 GB	Dense, Gated DeltaNet
Qwen 3.5-9B	9B	32	~6.6 GB	8 GB	Dense, Gated DeltaNet

Every model in this table handles text, images, and video natively. The vision encoder is in the base model, not a bolted-on adapter or a separate "-VL" variant you have to download separately. A 0.8B model that processes video. That didn't exist a month ago.

Other shared specs: 262K native context (extendable to ~1M via YaRN), 248K-token vocabulary covering 201 languages, multi-token prediction for faster inference, and Apache 2.0 licensing.

All four are dense models, meaning every parameter is active on every forward pass. No MoE routing, no expert selection. You get the same speed whether the query is simple or complex.

The 9B: why this is the new 8GB default

Look at the Qwen 3.5-9B model card. The numbers don't make sense until you factor in the architecture.

It beats models 3x its size

Benchmark	Qwen 3.5-9B	Qwen3-30B	Gap
GPQA Diamond	**81.7**	77.2	+4.5
IFEval	**91.5**	88.9	+2.6
LongBench v2	**55.2**	48.0	+7.2

A 9B model outscoring a 30B from the previous generation on graduate-level science questions, instruction following, and long-context comprehension. The LongBench gap is the most telling. 7.2 points is a large margin on a benchmark designed to stress long-context reasoning.

It destroys GPT-5-Nano on vision

Benchmark	Qwen 3.5-9B	GPT-5-Nano	Gap
MMMU-Pro	**70.1**	57.2	+12.9
MathVision	**78.9**	62.2	+16.7
OmniDocBench	**87.7**	55.9	+31.8

The OmniDocBench gap of 31.8 points is hard to overstate. OmniDocBench tests document understanding: tables, charts, forms, mixed layouts. A 9B open-source model on your GPU beating OpenAI's compact model by that margin on document tasks.

Full 9B benchmark sheet

For the benchmarks-first crowd, here's the complete picture:

Language:

Benchmark	Score
MMLU-Pro	82.5
MMLU-Redux	91.1
C-Eval	88.2
GPQA Diamond	81.7
IFEval	91.5
HMMT Feb 25 (math)	83.2
LongBench v2	55.2

Vision:

Benchmark	Score
MMMU	78.4
MMMU-Pro	70.1
MathVision	78.9
MathVista (mini)	85.7
OmniDocBench 1.5	87.7
VideoMME (w/ sub)	84.5
CountBench	97.2

97.2 on CountBench. The model can count objects in images better than most humans pay attention to.

The smaller three: where each one fits

Qwen 3.5-4B: the laptop model

6GB VRAM gets you here. That means laptops with a discrete GPU, Intel Arc cards, older NVIDIA cards with 6GB. Benchmark-wise, the 4B still scores 76.2 on GPQA Diamond and 66.3 on MMMU-Pro. Those are strong numbers for something that loads in 3.4GB on Ollama.

The 4B also gets native vision. You can point it at screenshots, documents, photos, and get structured responses without swapping models.

```
ollama run qwen3.5:4b
```

Qwen 3.5-2B: the integrated graphics tier

4GB of RAM or VRAM. This runs on machines without a dedicated GPU. The 2B model card shows 84.5 on OCRBench, which means it reads text in images well enough for practical document extraction. It scores 64.2 on MMMU, which is respectable for a model that fits in under 3GB.

Use cases: document OCR, receipt scanning, basic image understanding, structured data extraction from screenshots.

```
ollama run qwen3.5:2b
```

Qwen 3.5-0.8B: phone and Pi territory

Sub-billion parameters, 1GB on Ollama, runs on 2GB of RAM. The 0.8B model card shows 62.2 on MathVista and 49.0 on MMMU. Limited, but functional enough for structured tasks.

Where it gets interesting: intent classification, on-device query triage (decide locally whether to call a bigger model or handle it yourself), mobile assistants that need to understand what the camera sees, Raspberry Pi projects that need basic vision without cloud calls.

A 0.8B model with 262K context and native video understanding, running on a phone. That wasn’t possible before today.

```
ollama run qwen3.5:0.8b
```

Why the architecture matters

These small models score well above what you'd expect at their size, and Gated DeltaNet is why. It's the same hybrid architecture used in the 397B flagship.

The layout: three layers of DeltaNet (linear attention) followed by one layer of full softmax attention. A 3:1 ratio. DeltaNet uses constant memory. It doesn't grow with sequence length the way standard attention does. The periodic full attention layers preserve the reasoning quality that pure linear attention models lose.

In practice, three things matter:

First, 262K context on a 0.8B model. Standard transformers at 0.8B would choke on long context because the KV cache would exceed the model's own size. DeltaNet's constant-memory linear attention layers keep the KV cache manageable.

Second, multi-token prediction (MTP) is trained into all four models. Instead of predicting one token at a time, the model predicts multiple tokens per forward pass. Inference engines that support MTP (SGLang, vLLM) get real speed gains from this.

Third, the 0.8B and the 397B use the same building blocks. Optimizations at one scale transfer to the other. Fine-tuning results on the small models should translate more predictably to larger ones.

Getting started today

Ollama (fastest path)

All four models are already on Ollama:

```
# The new 8GB default
ollama run qwen3.5:9b
```

```
# Laptop with discrete GPU
ollama run qwen3.5:4b

# Integrated graphics or thin laptop
ollama run qwen3.5:2b

# Phone, Pi, edge device
ollama run qwen3.5:0.8b
```

GGUFs via Unsloth

Unsloth already has GGUF quantizations in 3-bit through 8-bit using their Dynamic 2.0 quantization. If you're running llama.cpp directly or want finer control over quantization, grab GGUFs from HuggingFace. Unsloth recommends at least the Q2_K_XL dynamic quant for the best size-to-accuracy tradeoff.

For production serving

SGLang and vLLM both support Qwen 3.5. If you need multi-user serving or want MTP acceleration, these are your backends. The 9B is small enough to serve multiple concurrent users on a single GPU.

How this stacks up against the competition

Nobody else has shipped a small model family this complete.

Google Gemma 3 has a 1B and 4B, but the smallest sizes don't have native vision. You need the larger models for multimodal. Llama 3.2 has 1B and 3B small models, but they're text-only; you need a separate pipeline for images. Microsoft Phi-4-mini is 14B (50% larger than the 9B) and still primarily text-focused.

Qwen 3.5 gives you four sizes from 0.8B to 9B with the same multimodal capabilities, same architecture, same 262K context. If you're building a pipeline where different hardware tiers run different model sizes, that consistency matters. The API and behavior are identical across sizes. Only the quality changes.

Apache 2.0 license. No usage restrictions, no commercial limitations, no "open" with asterisks.

VRAM and hardware matching

Model	Ollama Size	Min RAM/ VRAM	Best For	Run Command
0.8B	~1.0 GB	2 GB	Phone, Pi, edge, intent triage	`ollama run qwen3.5:0.8b`
2B	~2.7 GB	4 GB	Laptop integrated GPU, OCR, simple tasks	`ollama run qwen3.5:2b`
4B	~3.4 GB	6 GB	Laptop dGPU, balanced quality/ speed	`ollama run qwen3.5:4b`
9B	~6.6 GB	8 GB	Desktop, the new 8GB VRAM default	`ollama run qwen3.5:9b`

If you're on a Mac, check our Qwen 3.5 Mac: MLX vs Ollama guide. MLX runs Qwen 3.5 at roughly 2x Ollama speeds on Apple Silicon.

For the bigger Qwen 3.5 models (27B, 35B-A3B, 122B-A10B), see the Qwen 3.5 Complete Local Guide.

The bottom line

The "default 8GB model" just shifted. If you're running Qwen3-8B, Llama 3.1 8B, or Gemma 3 9B as your go-to local model, Qwen 3.5-9B replaces all of them. Higher reasoning scores, better instruction following, longer context, and native vision included. Same VRAM footprint.

On a Raspberry Pi or phone, the 0.8B gives you a multimodal model with 262K context in 1GB. That didn't exist before today.

Building something that needs to scale across hardware tiers? Same architecture, same API from 0.8B to 397B. Pick the size that fits the device and ship it.

```
ollama run qwen3.5:9b
```

The previous generation just became last generation.

Get notified when we publish new guides.

Subscribe — free, no spam

Source: https://insiderllm.com/guides/qwen-3-5-small-models-9b-beats-30b/

Free guides for running AI locally