Best Models Under 3B: Small LLMs That Work

January 29, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** Sub-3B models have gotten shockingly good. Qwen 3.5 2B (1.28GB Q4, native vision, 262K context) just leapfrogged last-gen models at this size. BitNet b1.58 2B4T fits in 0.4GB and runs on any CPU. For 3B, Qwen 2.5 3B and Llama 3.2 3B still lead. Start with `ollama run qwen3.5:2b` for the best quality-per-byte, or Llama 3.2 3B for instruction following. These aren't toys — they handle Q&A, summarization, tool calling, and classification well enough for real work.

📚 **More on this topic:** Run Your First Local LLM · CPU-Only LLMs · Quantization Explained · Qwen 3.5 9B Setup Guide

You don’t have a gaming GPU. Maybe you’re on a laptop with integrated graphics, a five-year-old desktop, a Raspberry Pi, or a phone. You’ve heard people running AI locally and you’re wondering: is that even possible on my hardware?

Yes. And not in a “technically it loads” way — in a “this is genuinely useful” way. The small model landscape changed dramatically in 2024-2025. A 3B model today outperforms a 7B model from 2023 on most benchmarks. A 1.5B model fits in under 2GB of RAM and generates faster than you can read.

This guide covers the best models under 3 billion parameters, what hardware you actually need, and what these models can and can’t do.

Who This Is For

If any of these describe your situation, this guide is for you:

Hardware	Typical RAM/VRAM	What You Can Run
Laptop (no dedicated GPU)	8-16GB RAM	3B models comfortably, multiple at Q4
Old GPU (GTX 1050 Ti, 1060)	4-6GB VRAM	3B models with room to spare
Raspberry Pi 5	8GB RAM	1B-3B models at usable speeds
Phone (recent Android/iPhone)	6-8GB RAM	0.5B-1.5B models, some 3B

Hardware	Typical RAM/VRAM	What You Can Run
Chromebook / thin laptop	4GB RAM	0.5B-1.5B models at Q4
Desktop with no GPU	8-32GB RAM	Any sub-3B model, fast

You're not the person with an RTX 4090 looking for the optimal model. You're the person wondering if local AI is even possible on what you've got. It is.

Why Small Models Matter Now

Two years ago, a 3B model was barely useful. It could complete sentences and sometimes follow instructions, but the output was rough. You needed at least 7B parameters for anything practical.

That changed fast. Three things happened:

Better training data. Model quality scales with data quality, not just size. Qwen 2.5 3B was trained on 18 trillion tokens of carefully curated data — more than many early 70B models saw.

Knowledge distillation. Smaller models now learn from larger ones during training. Llama 3.2 3B was distilled from Llama 3.1 70B, inheriting capabilities that would otherwise require far more parameters.

Architecture improvements. Grouped-query attention, better tokenizers, and improved positional encodings all help small models punch above their weight.

The result: Qwen 2.5 3B scores 65.6 on MMLU. The original Llama 2 7B scored 45.3. A model less than half the size, beating one twice as large.

And it keeps accelerating. Qwen 3.5 2B (March 2026) hits MMLU-Pro 55.3 in non-thinking mode, handles images and video natively, and supports 262K context — from a model that fits in 1.28GB at Q4. Microsoft's BitNet b1.58 2B4T takes a different approach: ternary weights ({-1, 0, +1}) that fit in 0.4GB and run on CPU at speeds that match or beat conventional 2B models. The floor keeps rising.

The Best Sub-3B Models, Ranked

1. Qwen 2.5 3B — Best All-Rounder

The strongest model at this size class, period. Qwen 2.5 3B matches or beats the previous-generation Qwen 2 7B on most benchmarks while using less than half the memory.

Metric	Score
MMLU	65.6
GSM8K (math)	79.1
HumanEval (coding)	42.1
HellaSwag	74.6
RAM at Q4_K_M	~2.5 GB
File size (Q4_K_M)	~2.0 GB

Strong at multilingual tasks, solid at coding, good instruction following. If you can run a 3B model, this is the default choice.

```
ollama pull qwen2.5:3b
```

2. Llama 3.2 3B — Best Instruction Following

Meta's small model, distilled from the 70B. Where Qwen 2.5 3B leads on raw benchmarks, Llama 3.2 3B excels at doing what you ask it to do. It scores 77.4 on IFEval (instruction following) — the highest in its class.

Metric	Score
MMLU	63.4
GSM8K (math)	77.7
ARC-C (reasoning)	78.6
IFEval	77.4
RAM at Q4_K_M	~2.5 GB

Metric	Score
File size (Q4_K_M)	~2.0 GB

Particularly good at tool use (BFCL V2: 67.0) and multilingual tasks (MGSM: 58.2). If you're building something that needs reliable instruction following — a chatbot, an assistant, a workflow tool — Llama 3.2 3B is the pick.

```
ollama pull llama3.2:3b
```

3. Phi-3.5 Mini (3.8B) — The Overachiever

Technically 3.8B parameters — slightly over the 3B line — but it earns its spot here. Phi-3.5 Mini punches absurdly above its weight. It beats Mixtral 8x7B (a 46.7B MoE model) on math benchmarks and nearly matches GPT-3.5 on MMLU.

Metric	Score
MMLU	69.0
GSM8K (math)	86.2
HumanEval (coding)	62.8
BBH (hard reasoning)	69.0
RAM at Q4_K_M	~3.0 GB
File size (Q4_K_M)	~2.3 GB

Best coding and math performance under 4B parameters by a wide margin. The tradeoff: weaker on factual recall (TriviaQA: 64.0 vs GPT-3.5's 85.8) and somewhat less natural in free-form conversation. If your tasks lean toward reasoning and code, Phi-3.5 Mini is the best you'll find anywhere near this size.

```
ollama pull phi3.5
```

4. Qwen 3.5 2B — Best Under 3B (New)

The new king of the sub-3B space. Qwen 3.5 2B is natively multimodal (text, images, video from the same weights), supports 262K context, and scores MMLU-Pro 55.3 in non-thinking mode — well above any previous 2B model. Turn on thinking mode and it hits 66.5. It also handles vision tasks that no other sub-3B model can touch: MMMU 64.2, MathVista 76.7.

Metric	Score
MMLU-Pro (non-thinking)	55.3
MMLU-Redux	69.2
IFEval	61.2
MMMU (vision)	64.2
RAM at Q4_K_M	~2.0 GB
File size (Q4_K_M)	1.28 GB

If you need vision, tool calling, or just want the best 2B model available, this is it. The 262K context window is absurd at this size.

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

5. Qwen 3.5 0.8B — Best Under 1B (New)

Replaces Qwen 2.5 0.5B as the tiniest model worth running. Same multimodal architecture as the 2B — images, video, 262K context — packed into 533MB at Q4. MMLU-Pro 29.7 in non-thinking mode, 42.3 in thinking mode. It handles OCR (74.5), basic vision tasks (MMStar 58.3), and simple text tasks at speeds that feel instant on a Pi 5 or phone.

Metric	Score
MMLU-Pro (non-thinking)	29.7
MMLU-Redux	48.5
IFEval	52.1
OCRBench	74.5
RAM at Q4_K_M	~1.0 GB

Metric	Score
File size (Q4_K_M)	533 MB

The Qwen 2.5 0.5B scored 47.5 on MMLU with no vision. This replaces it at every size except the absolute smallest deployments.

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

6. BitNet b1.58 2B4T — The 0.4GB Option (New)

Microsoft's ternary-weight model. Every weight is -1, 0, or +1 — no floating point, no quantization artifacts. The model was trained natively at 1.58-bit on 4 trillion tokens. Result: 0.4GB of memory for the weights, 29ms CPU decoding latency, and benchmark scores that compete with conventional 2B models.

Metric	Score
MMLU	53.17
GSM8K (math)	58.38
HumanEval+ (coding)	38.40
WinoGrande	71.90
RAM	~0.4 GB
Context	4096 tokens

The catch: you need bitnet.cpp to get the efficiency gains. No Ollama, no LM Studio, no standard GGUF. It's a separate runtime. If you're comfortable building from source, the performance-per-watt is unmatched. If you want one-command setup, stick with Qwen 3.5 0.8B above.

7. Qwen 2.5 1.5B — Previous Best Under 2B

Superseded by Qwen 3.5 2B for most tasks, but still a solid choice if you need a model with a well-tested ecosystem and wide compatibility. Scores 60.9 on MMLU — impressive for 1.5B — and runs at 8-15 tok/s on a Pi 5.

Metric	Score
MMLU	60.9

Metric	Score
GSM8K (math)	68.5
HumanEval (coding)	37.2
HellaSwag	67.9
RAM at Q4_K_M	~1.5 GB
File size (Q4_K_M)	~1.1 GB

```
ollama pull qwen2.5:1.5b
```

8. Gemma 2 2B — Google's Efficient Pick

Google's entry uses knowledge distillation from larger Gemma models to pack capability into 2B parameters. Its strength is language understanding — strong HellaSwag (72.9), BoolQ (72.7), and factual recall (TriviaQA: 60.4).

Metric	Score
MMLU	52.2
HellaSwag	72.9
Winogrande	71.3
TriviaQA	60.4
RAM at Q4_K_M	~1.8 GB
File size (Q4_K_M)	~1.1 GB

Weak on math (GSM8K: 24.3) and coding (HumanEval: 20.1). Don't pick Gemma 2 2B for those tasks. But for commonsense reasoning, entity extraction, and classification, it's solid. It also has excellent KV cache efficiency, making it a good choice for serving multiple users.

```
ollama pull gemma2:2b
```

9. Llama 3.2 1B — The Ultralight

Meta's smallest. At 1.24B parameters, it fits in under 1.5GB of RAM at Q4 and runs at 30-60+ tok/s on a desktop CPU. Not the smartest model on this list, but fast enough to feel instant.

Metric	Score
MMLU	49.3
GSM8K (math)	44.4
ARC-C (reasoning)	59.4
IFEval	59.5
RAM at Q4_K_M	~1.2 GB
File size (Q4_K_M)	~800 MB

Best for: quick answers, text classification, simple extraction tasks, and prototyping. At this size, you can run it alongside other applications without worry.

```
ollama pull llama3.2:1b
```

10. StableLM 2 1.6B — The Veteran

Released in early 2024 by Stability AI, StableLM 2 was state-of-the-art for sub-2B models at launch. It's since been surpassed by Qwen 2.5 1.5B and Llama 3.2 1B on most benchmarks, but it still has a niche: multilingual support across 7 languages and strong language understanding (HellaSwag: 70.5).

Metric	Score
MMLU	41.8 (Zephyr)
HellaSwag	70.5
Winogrande	64.6
RAM at Q4_K_M	~1.3 GB
File size (Q4_K_M)	~1.0 GB

Unless you specifically need its multilingual coverage, Qwen 3.5 2B or Qwen 2.5 1.5B are both better choices today.

```
ollama pull stablelm2:1.6b
```

Also worth knowing: Falcon3 1B and 3B

TII's Falcon3 family includes 1B and 3B models trained on 14 trillion tokens, available in standard GGUF and 1.58-bit variants. The 1.58-bit versions use the same bitnet.cpp runtime as BitNet b1.58 2B4T. Benchmark scores are lower than Qwen and Llama equivalents (Falcon3-3B MMLU-PRO: 29.7, MATH: 19.9), but if you're already running bitnet.cpp, they give you more size options in the ternary-weight ecosystem. For most users, Qwen 3.5 and Llama 3.2 are the better choices at these sizes.

11. Qwen 2.5 0.5B — Superseded

Qwen 3.5 0.8B replaces this for most use cases — it's only 133MB larger at Q4 but adds vision, 262K context, and better benchmarks. Qwen 2.5 0.5B is still the smallest option if you need to fit under 500MB or your runtime doesn't support Qwen 3.5 yet.

Metric	Score
MMLU	47.5
GSM8K (math)	41.6
HumanEval (coding)	30.5
RAM at Q4_K_M	~0.8 GB
File size (Q4_K_M)	~400 MB

```
ollama pull qwen2.5:0.5b
```

Head-to-Head Comparison

All models at Q4_K_M quantization (except BitNet, which is natively 1.58-bit):

Model	Params	RAM	File Size	MMLU-Pro	Best For
Phi-3.5 Mini	3.8B	~3.0 GB	~2.3 GB	— (MMLU 69.0)	Coding, math, reasoning
Qwen 2.5 3B	3B	~2.5 GB	~2.0 GB	— (MMLU 65.6)	All-around, multilingual
Llama 3.2 3B	3B	~2.5 GB	~2.0 GB	— (MMLU 63.4)	Instruction following, chat
Qwen 3.5 2B	**2B**	**~2.0 GB**	**1.28 GB**	**55.3**	**Vision, quality under 3B**
BitNet b1.58 2B4T	2B	~0.4 GB	~0.4 GB	— (MMLU 53.17)	CPU-only, minimal memory
Qwen 2.5 1.5B	1.5B	~1.5 GB	~1.1 GB	— (MMLU 60.9)	Tested ecosystem
Gemma 2 2B	2B	~1.8 GB	~1.1 GB	— (MMLU 52.2)	Classification, extraction
Llama 3.2 1B	1.24B	~1.2 GB	~800 MB	— (MMLU 49.3)	Speed, prototyping
Qwen 3.5 0.8B	**0.8B**	**~1.0 GB**	**533 MB**	**29.7**	**Vision, edge, under 1B**
Qwen 2.5 0.5B	0.5B	~0.8 GB	~400 MB	— (MMLU 47.5)	Absolute minimum
StableLM 2 1.6B	1.6B	~1.3 GB	~1.0 GB	— (MMLU 41.8)	Multilingual (7 languages)

Note: Qwen 3.5 models report MMLU-Pro (harder benchmark). Older models report classic MMLU. The scores aren't directly comparable — MMLU-Pro 55.3 is a strong result.

What Small Models Are Good At

Sub-3B models won't replace GPT-4. But for specific tasks, they're more than good enough — and they do it locally, privately, and for free.

Tasks where sub-3B models deliver:

- **Quick Q&A** — "What's the capital of France?" "How do I reverse a list in Python?" Fast answers, no API call needed.
- **Summarization** — Summarize a paragraph, an email, or a short document. Qwen 2.5 3B and Llama 3.2 3B handle this well.
- **Text classification** — Sentiment analysis, topic categorization, spam detection. Fine-tuned small models hit 90%+ accuracy on classification tasks.
- **Simple coding** — Generate a function, fix a syntax error, explain a code snippet. Phi-3.5 Mini scores 62.8 on HumanEval — that's real coding ability.
- **Translation** — Simple translations work well, especially with Qwen (strong multilingual training) and Llama 3.2 (trained on 8 languages).

- **Data extraction** — Pull names, dates, and structured fields from unstructured text. Gemma 2 2B is particularly good at this.
- **Autocomplete and suggestions** — Fast enough for real-time text completion in editors.
- **Tool calling and function use** — This is new. Qwen 3.5 2B scores 43.6 on BFCL-V4 (tool calling benchmark) in thinking mode. Llama 3.2 3B scores 67.0 on BFCL V2. Ministral 3B was designed specifically for structured JSON output and agentic workflows. Small models are getting surprisingly capable at calling functions, formatting JSON, and driving simple tool-use pipelines.

What Small Models Can't Do

Being honest about limits saves frustration.

Don't expect these:

- **Complex multi-step reasoning** — "Plan a two-week trip optimizing for budget and weather across five cities" will produce mediocre output. The model doesn't have the capacity to hold complex chains of logic.
- **Long-form writing** — Blog posts, essays, fiction beyond a few paragraphs. Coherence breaks down as output length increases.
- **Advanced math** — Multi-step proofs, calculus, competition-level problems. Even Phi-3.5 Mini's strong GSM8K score (86.2) drops hard on MATH (41.3) — the harder benchmark.
- **Nuanced analysis** — Comparing legal documents, analyzing research papers, weighing subtle tradeoffs. These tasks need more parameters.
- **Large context processing** — Most sub-3B models work best with 2048-4096 tokens of context. Qwen 3.5 models technically support 262K tokens, but a 2B model's ability to reason over long context is limited regardless of the window size. Feeding them 10-page documents produces unreliable results.
- **Code generation for complex projects** — Small models generate individual functions, not multi-file architectures.

The rule of thumb: if a task requires you to think hard about it, a sub-3B model will struggle with it too. For those tasks, step up to 7B-8B models — they only need 4-5GB of RAM at Q4.

Hardware Requirements

Sub-3B models run on almost anything. Here's exactly how much resources each tier needs:

RAM Requirements (Q4_K_M Quantization)

Model Size	Weights	Total with Context	Minimum RAM
0.5B (Qwen 2.5)	~400 MB	~0.8 GB	2 GB
0.8B (Qwen 3.5)	~533 MB	~1.0 GB	2 GB
1B	~600 MB	~1.2 GB	2 GB
1.5B	~900 MB	~1.5 GB	4 GB
2B (Qwen 3.5)	~1.28 GB	~2.0 GB	4 GB
2B (BitNet)	~0.4 GB	~0.7 GB	2 GB
3B	~1.7 GB	~2.5 GB	4 GB

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

"Total with Context" includes the KV cache at 2048-4096 tokens plus runtime overhead. "Minimum RAM" is total system RAM — you need room for the OS and runtime too.

Storage

Downloads are small. A 3B model at Q4 is about 2GB. A 0.5B model is 400MB. You can fit half a dozen sub-3B models in less space than a single 7B model.

Model	Q4_K_M File Size
BitNet b1.58 2B4T	~0.4 GB (native 1.58-bit)
Qwen 2.5 0.5B	~400 MB
Qwen 3.5 0.8B	533 MB
Llama 3.2 1B	~800 MB
Qwen 2.5 1.5B	~1.1 GB
Gemma 2 2B	~1.1 GB
Qwen 3.5 2B	1.28 GB

Model	Q4_K_M File Size
Llama 3.2 3B	~2.0 GB
Phi-3.5 Mini	~2.3 GB

Speed Expectations

Small models are fast. On most hardware, you'll be reading slower than the model generates.

Desktop and Laptop CPUs

CPU	1B Model (Q4)	3B Model (Q4)
Intel i5/Ryzen 5 (laptop)	~20-40 tok/s	~8-15 tok/s
Intel i7/Ryzen 7 (desktop)	~30-60 tok/s	~12-25 tok/s
Apple M1/M2	~35-70 tok/s	~15-30 tok/s
Apple M3 Pro+	~45-90 tok/s	~20-40 tok/s
AMD Ryzen AI 9 (laptop)	~50 tok/s	~18-28 tok/s

Memory bandwidth is the bottleneck, not CPU speed. Dual-channel DDR5 is noticeably faster than DDR4. Single-channel RAM can cut throughput by 50-70% — if your laptop has one RAM stick, that's your limit.

For more on CPU inference, see our CPU-only LLM guide.

Raspberry Pi 5

Model	tok/s	Usability
Qwen 2.5 0.5B	~20	Fast — real-time chat
Qwen 2.5 1.5B	~8-12	Usable — slight pauses
Llama 3.2 3B	~4-6	Slow but functional
7B models	~2-5	Painful — not recommended

Stick to 1B-1.5B on a Pi 5 for a good experience. 3B is possible but you'll feel the wait. Use active cooling — all four cores hit 100% during inference.

Old GPUs

If you have a dedicated GPU, even an old one, it helps:

GPU	VRAM	What Fits	Advantage
GTX 1050 Ti	4GB	3B at Q4 comfortably	2-3x faster than CPU-only
GTX 1060	6GB	3B at Q8, or 7B at Q4	Enough for 7B models
RX 580	8GB	3B at FP16	Full precision, no quantization needed

Even a 4GB GPU fully offloads a 3B Q4 model (weights are ~1.7GB), giving a significant speed boost over CPU inference.

Phones

Phone Tier	Model	Speed
Flagship (Snapdragon 8 Gen 3, A17 Pro)	1B-3B at Q4	8-17 tok/s
Mid-range (Snapdragon 7 Gen 1)	0.5B-1.5B at Q4	5-10 tok/s
Budget (6GB RAM or less)	0.5B at Q4	Barely loads

Apps like SmolChat (Android) and MLC Chat (iOS/Android) make this straightforward. Be warned: sustained inference drains battery fast — comparable to a graphics-intensive game.

How to Run Them

Ollama (Easiest)

Ollama is one command to install, one command to run:

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

# Pull and run a model
ollama pull qwen2.5:3b
ollama run qwen2.5:3b
```

That's it. Ollama auto-detects your hardware and optimizes accordingly. No GPU required.

LM Studio (GUI)

Prefer a visual interface? LM Studio gives you a ChatGPT-like UI for local models. Download, search for a model, click run. It handles GGUF quantization selection for you.

Raspberry Pi

On a Pi 5, Ollama works out of the box:

```
curl -fsSL https://ollama.com/install.sh | sh
ollama pull qwen2.5:1.5b
ollama run qwen2.5:1.5b
```

For better performance on a Pi, consider building llama.cpp with OpenBLAS — it's 10-20% faster than Ollama for sustained inference.

Phones

- **Android:** SmolChat, MLC Chat, or any app that supports GGUF models
- **iOS:** MLC Chat, or LLM Farm
- **Cross-platform apps:** llama.rn (React Native bindings for llama.cpp)

When to Stay Small vs. Upgrade to 7B

This is the real question. Here's the decision framework:

Stay with sub-3B if:

- Your hardware maxes out at 4GB RAM/VRAM
- You're running on a Raspberry Pi, phone, or edge device
- Your tasks are quick Q&A, classification, extraction, or simple code
- Speed matters more than depth (you need real-time responses)
- You want to run alongside other applications without memory pressure
- Privacy/offline is the priority and quality is secondary

Step up to 7B-8B if:

- You have 8GB+ RAM or any GPU with 6GB+ VRAM
- You need multi-step reasoning, longer outputs, or complex coding
- Quality per response matters more than speed
- You're hitting the limits of 3B output quality

The jump from 3B to 7B is the single biggest quality improvement in local AI. A Llama 3.1 8B at Q4 uses about 5GB of RAM and is dramatically more capable. If your hardware can handle it, it's worth the step — see our 8GB VRAM guide for details.

But if your hardware can't handle 7B, don't feel locked out. A Qwen 2.5 3B today is more useful than a 7B model from two years ago. The floor has risen.

Recommendations by Use Case

Use Case	Best Pick	Runner-Up	Why
General chat/Q&A	Qwen 2.5 3B	Llama 3.2 3B	Strongest overall quality
Coding assistance	Phi-3.5 Mini (3.8B)	Qwen 2.5 3B	62.8 HumanEval — real coding ability
Math/reasoning	Phi-3.5 Mini (3.8B)	Qwen 2.5 3B	86.2 GSM8K, untouchable at this size
Vision tasks (under 3B)	Qwen 3.5 2B	Qwen 3.5 0.8B	Only sub-3B models with native vision
Tool calling / agents	Llama 3.2 3B	Qwen 3.5 2B	BFCL V2 67.0 (Llama), 43.6 (Qwen)
Classification/extraction	Gemma 2 2B	Qwen 3.5 2B	Strong language understanding, efficient
Raspberry Pi 5	Qwen 3.5 2B	Qwen 3.5 0.8B	Vision + 262K context in 1.28GB
Phone	Qwen 3.5 0.8B	Llama 3.2 1B	533MB, native vision, fast
Edge/IoT	Qwen 3.5 0.8B	BitNet b1.58 2B4T	533MB (Qwen) or 0.4GB (BitNet)
Minimal memory (CPU-only)	BitNet b1.58 2B4T	Qwen 2.5 0.5B	0.4GB RAM, 29ms latency
Multilingual	Llama 3.2 3B	Qwen 3.5 2B	58.2 MGSM (Llama), 201 languages (Qwen)

Use Case	Best Pick	Runner-Up	Why
Absolute minimum hardware	BitNet b1.58 2B4T	Qwen 3.5 0.8B	0.4GB RAM, any CPU

The Bottom Line

Small models are no longer a consolation prize. They're a legitimate way to run AI locally on hardware you already own — no GPU required, no cloud dependency, no subscription.

The practical advice:

1. **Have 4GB+ RAM?** Start with `ollama pull qwen2.5:3b`. Still the strongest 3B all-rounder.
2. **Want vision + quality in under 2GB?** `ollama run qwen3.5:2b`. Native images/video, 262K context, 1.28GB download.
3. **Only 2GB RAM or a Pi?** `ollama run qwen3.5:0.8b`. 533MB download, handles vision, and runs fast on constrained hardware.
4. **Want the absolute smallest memory footprint?** Build bitnet.cpp and run BitNet b1.58 2B4T. 0.4GB, any CPU.
5. **Need coding or math?** `phi3.5` (3.8B) is still the strongest small model for technical tasks.
6. **When you outgrow 3B** — and you'll know when the answers aren't good enough — Qwen 3.5 9B on 8GB VRAM is the next step.

Your laptop is more capable than you think. Try it.

Related Guides

- Run Your First Local LLM in 15 Minutes
- CPU-Only LLMs: What Actually Works
- What Can You Actually Run on 8GB VRAM?
- What Quantization Actually Means

Sources: Qwen 2.5 Technical Report, Meta Llama 3.2 Model Card, Phi-3 Technical Report, Gemma 2 Technical Report, StableLM 2 Technical Report, Raspberry Pi 5 LLM Benchmarks, CPU vs GPU LLM Performance

Get notified when we publish new guides.

Subscribe — free, no spam

Source: https://insiderllm.com/guides/best-models-under-3b-parameters/