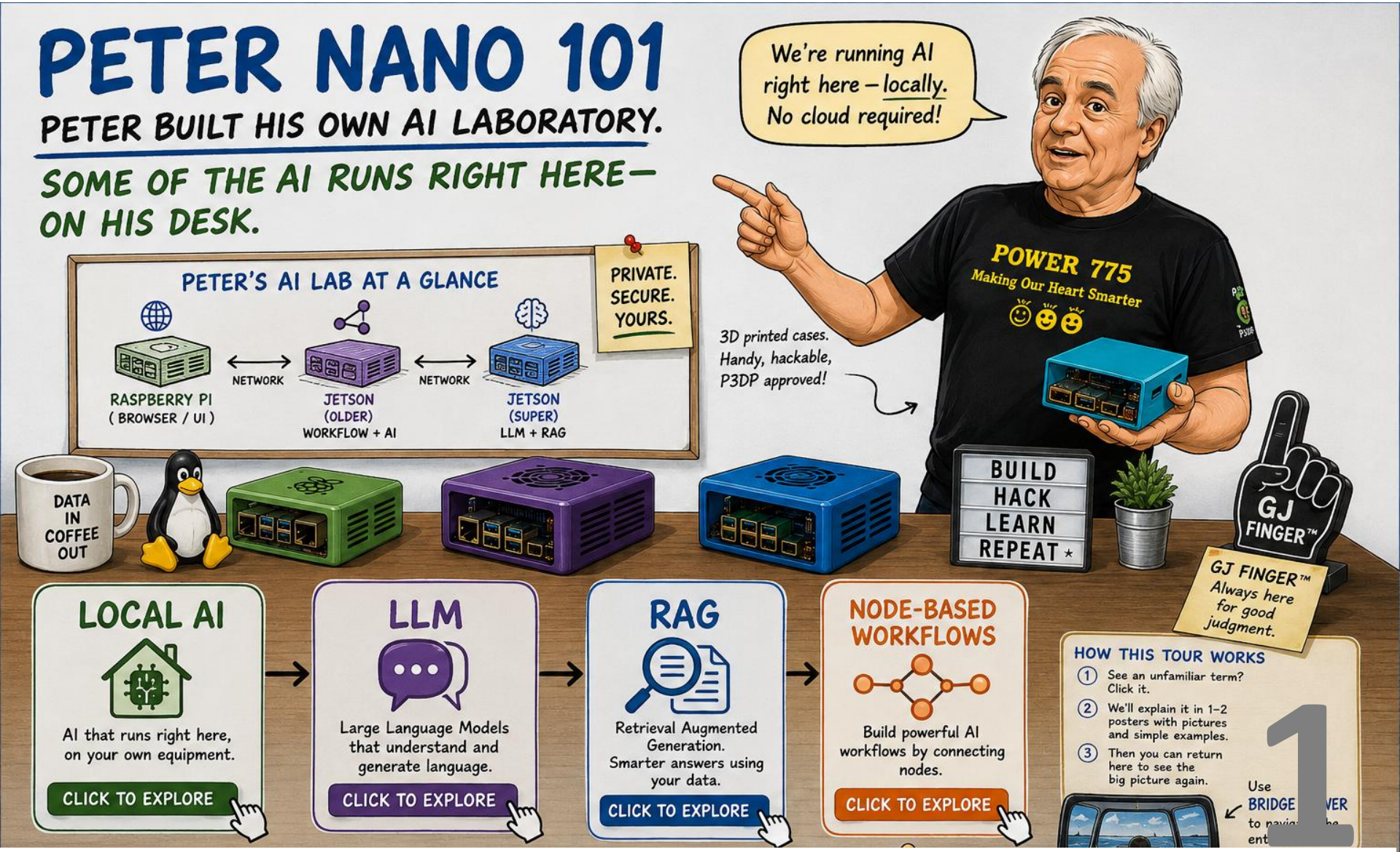
Jetson Nano update by Peter

July 2026



Jetson Nano update by Peter

July 2026

PETER NANO 101
POSTER #1 OF THE SERIES

WHAT IS AN LLM?

A Large Language Model is the "brain" that understands and generates human language.

HOW IT WORKS (IN SIMPLE TERMS)

1 TRILLIONS OF WORDS
The model is trained on massive amounts of text from books, web pages, code, conversations, and more.

2 FIND PATTERNS
It learns patterns in how words fit together, facts, reasoning, style, and instructions.

3 UNDERSTAND & PREDICT
When you ask something, it predicts the best next words, one at a time, to produce a helpful answer.

4 RESPOND TO YOU
You get a response in natural language—just like a conversation.

INPUT → PROCESS → OUTPUT

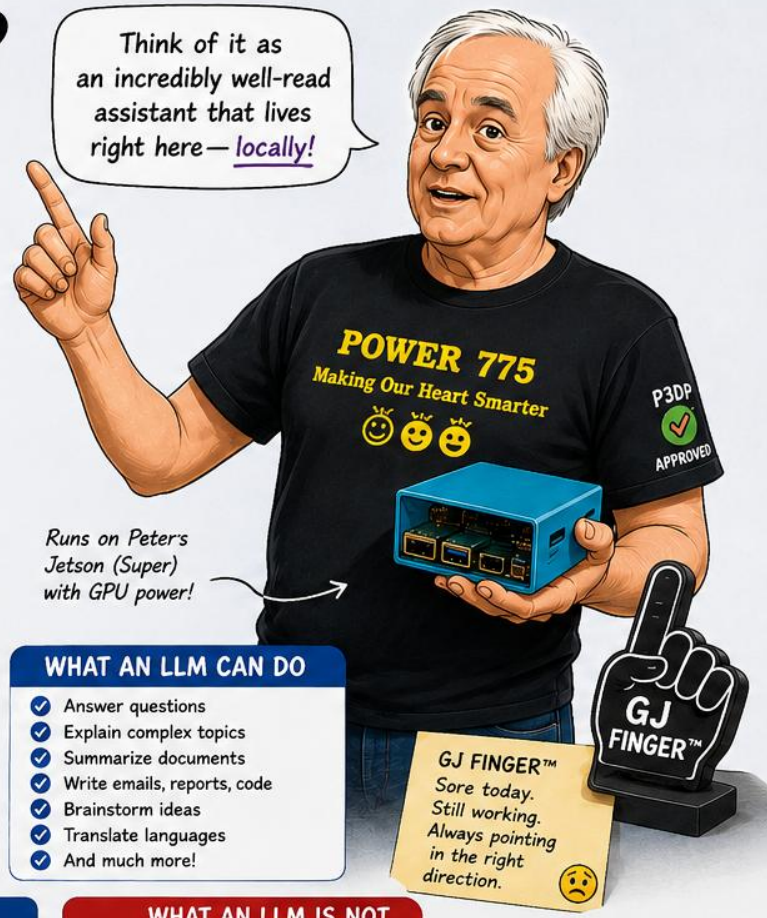
YOU ASK
What is photosynthesis?
Your question (or prompt)

THE LLM THINKS
The model analyzes patterns and knowledge to figure out the best answer.

IT ANSWERS
Photosynthesis is the process plants use to turn sunlight into energy...
The model generates a helpful response.

KEY POINTS

- It runs locally on Peter's hardware.
- Your data stays private and secure.
- GPU power makes it fast and capable.
- It's smart, but not perfect—always review answers.



WHAT AN LLM CAN DO

- Answer questions
- Explain complex topics
- Summarize documents
- Write emails, reports, code
- Brainstorm ideas
- Translate languages
- And much more!

EXAMPLES OF LLMs (RUN LOCALLY)

Llama 3
Open source. Powerful. Great for general use.

Mistral
Fast and efficient. Excellent performance.

Phi / Others
Many models exist. Pick the right one for your needs.

Peter chooses the best model for the job and his hardware.

WHERE IT RUNS

Jetson (Super)
Local brains need local horsepower.
GPU + CPU + Memory = LLM performance.

WHAT AN LLM IS NOT

- It doesn't know everything.
- It can make mistakes.
- It doesn't have real-world experiences.
- It follows patterns, not feelings.

Good tool. Great assistant. Not magic.

Jetson Nano update by Peter

July 2026

PETER NANO 101
POSTER #1A OF THE SERIES

HOW BIG IS AN LLM?

And why does Peter need all those GPU cores?

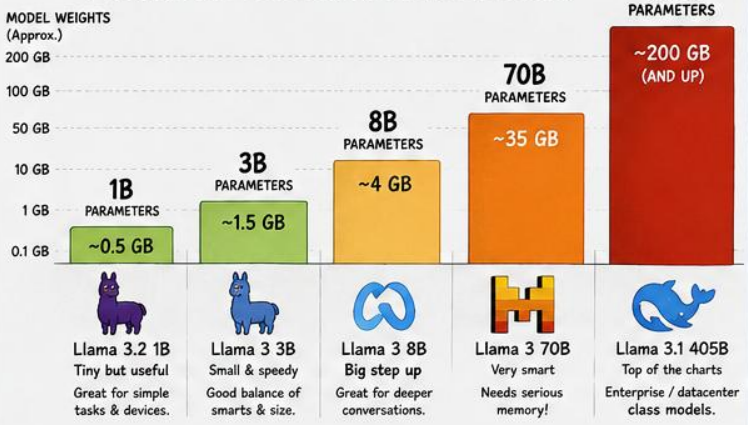
Bigger models know more, but they need more memory and more muscle to run! Let's look inside.

A. THE SIZE OF LLMS IN THE WORLD

LLMs are measured by PARAMETERS.
More parameters = more knowledge ... and more memory needed.

Rule of thumb (4-bit quantized): 1 BILLION parameters ≈ 0.5 GB of model weights

POPULAR LLMS: SIZE COMPARISON (4-bit WEIGHTS)



Llama 3.2 1B	Llama 3 3B	Llama 3 8B	Llama 3 70B	Llama 3.1 405B
Tiny but useful	Small & speedy	Big step up	Very smart	Top of the charts
Great for simple tasks & devices.	Good balance of smarts & size.	Great for deeper conversations.	Needs serious memory!	Enterprise / datacenter class models.

Peter's Jetson (Super) has 8 GB of unified memory.

The OS, runtime, and conversations also need space.

That's why 8B is about the sweet spot on 8 GB.

Sizes are approximate. Depends on quantization, tokenizer, and context length.

B. WHY A GPU? PARALLEL POWER!

An LLM does a TON of math for every token (word piece).
A GPU is built to do MANY calculations at the same time.

CPU: FEW FAST WORKERS

Great at doing a few things really well.
Example: 6-core ARM CPU

Works hard, but takes turns.
Good for tasks that are mostly sequential (one step at a time).

VS.

GPU: MANY PARALLEL WORKERS

Great at doing MANY things at once.
Example: 1024 CUDA cores (32 Tensor Cores) on Jetson (Super)

Thousands of tiny math workers in parallel.
This is what LLMs LOVE.
Much faster for huge matrix math!

REAL-WORLD SPEED SHOWDOWN
Tokens per Second (Higher is better)

MODEL (4-bit)	JETSON (SUPER) GPU (1024 CUDA cores)	CPU ONLY (6-core ARM)	SPEEDUP (GPU vs CPU)
Llama 3 8B	21.3 tokens/s	1.2 tokens/s	17.8× FASTER
Llama 3 3B	48.7 tokens/s	3.0 tokens/s	16.2× FASTER
Llama 3.2 1B	97.5 tokens/s	7.0 tokens/s	13.9× FASTER

GPU parallelism + high memory bandwidth (102 GB/s on Jetson Super) = BIG SPEED ADVANTAGE for LLMs!

PETER NANO 101
POSTER #4: WHAT IS RAG?
RETRIEVAL-AUGMENTED GENERATION

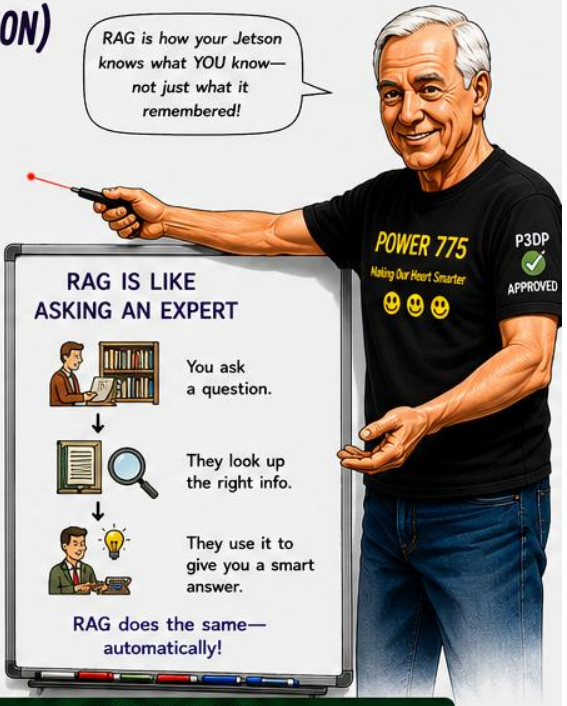
WHAT IS RAG? (RETRIEVAL-AUGMENTED GENERATION)

RAG helps your local AI answers stay accurate, current, and relevant.

Instead of relying only on what the model remembers,
RAG looks up the facts you provide—then lets the model use them to answer.

LEVEL 1 ●○○
LET'S GET THE BIG PICTURE

We're diving into the "what" and "why" of RAG.



YOUR JOURNEY (YOU ARE HERE)

1. **WELCOME ABOARD!** Meet Peter and the mission.
2. **HOW BIG IS AN LLM?** Size, parameters & why GPU.
3. **WHY 8 GB MATTERS** Memory limits & smart choices.
4. **WHAT IS RAG?** Retrieval-Augmented Generation.
5. **NODE-BASED WORKFLOWS** Connecting the dots.
6. **COMFYUI OVERVIEW** The canvas and key parts.
7. **SECTOR DELTA FIVE** Engage@ & real-world demo.
8. **TIPS & BEST PRACTICES** Do it right, keep it fast.

Use the Bridge Viewer (lower right) to return HOME to the BIG PICTURE anytime!

1. THE PROBLEM WITH MEMORY ALONE

LLMs have a cutoff date and can "forget" details.
They may also confidently guess (hallucinate).

LIMITATIONS
- ✕ Outdated info
- ✕ No access to private docs
- ✕ May hallucinate or guess wrong

RESULT

Great at patterns, not at knowing what's in your files. That's where RAG comes in!

2. WHAT RAG DOES

RAG finds the right information in your data
AND lets the model use it to answer.

1. **RETRIEVE** Find relevant information.
2. **AUGMENT** Add it to the question.
3. **GENERATE** Model uses it to create an answer.
4. **ANSWER** You get a better, factual response.

RAG = the right info + the right model = the right answer!

3. THE RAG PIPELINE (AT A GLANCE)

- **YOUR DATA** Docs, PDFs, notes, web pages, tables, code, and more.
- **CHUNK & INDEX** Split into chunks. Create embeddings. Store in a vector DB.
- **RETRIEVE** Find the most relevant chunks for the question.
- **AUGMENT & GENERATE** Send chunks + question to the LLM.
- **ANSWER** Response with sources (you can verify).

4. WHY RAG MATTERS

- ✓ More accurate answers
- ✓ Up-to-date information
- ✓ Your private data stays local
- ✓ Citations = trust & verification
- ✓ Works with smaller models (great for Jetson Nano!)

EXAMPLE
Q: "What does our lab procedure say about 3D printer bed cleaning?"
A: "Use 90% isopropyl alcohol on a lint-free cloth..."
Source: Lab_Procedures_v2.pdf (Page 4)

RAG = accuracy, relevance, and trust!

5. RAG AND JETSON NANO

- ✓ 8 GB memory = smart sandbox
- ✓ Run local models + local RAG
- ✓ Vector DBs fit in memory
- ✓ Fast answers with your own data
- ✓ Privacy stays on your desk

Right-size your models and index.
Smart setup = smooth performance!

6. QUICK TAKEAWAY

- ◆ LLMs are powerful, but memory isn't perfect.
- ◆ RAG gives the model the facts it needs.
- ◆ Retrieve → Augment → Generate → Answer.
- ◆ Your data + the right model = better results.
- ◆ On Jetson Nano, RAG makes small models shine!

Use RAG so your AI knows what YOU know.
That's the power of local intelligence!

Jetson Nano update by Peter

July 2026

PETER NANO 101

PETER'S ACTUAL 3-COMPUTER ARCHITECTURE

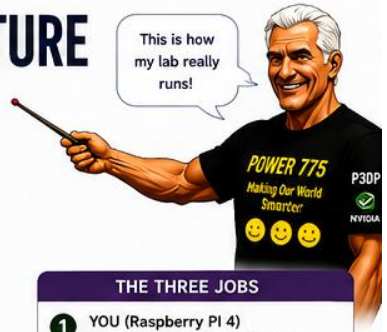
One Table. Three Computers. Three Jobs.

PETER'S ACTUAL 3-COMPUTER ARCHITECTURE

— HOW EVERYTHING WORKS TOGETHER ON PETER'S TABLE —

Three computers. Three jobs. One network.

George says, "Follow the data, follow the answers!"



1 RASPBERRY PI 4 (BROWSER STATION)

ROLE: USER INTERFACE & CONTROL

- Runs the web browser
- ComfyUI front end
- You drive everything here

2 JETSON NANO (OLDER) (COMFYUI / INFERENCE)

ROLE: WORKFLOWS & IMAGE GENERATION

NVIDIA JETSON

- Runs ComfyUI workflows
- Handles image generation
- Uses models via network

3 JETSON ORIN NANO (NEWER) (LLM SERVER / RAG)

ROLE: AI BRAIN & KNOWLEDGE

NVIDIA JETSON

- Runs the local LLM
- Answers questions, does RAG
- Heavy lifting happens here

THE THREE JOBS

1. **YOU (Raspberry PI 4)** — You ask questions, start things, see results in the browser.
2. **COMFYUI (Jetson Nano)** — Builds images, videos, and other AI creations from your prompts.
3. **AI BRAIN (Jetson Orin Nano)** — Thinks, remembers, retrieves knowledge (RAG), and answers.

HOW TO READ THIS POSTER

- Big picture of Peter's real setup.
- Three computers. Three jobs.
- Follow the arrows to see how everything connects.
- Click any callout (in the PDF or on the web) to learn more!

ON THE WALL BEHIND PETER

Peter built it on two tongue & groove wooden boards.
Neat, solid, and serviceable!

BOARD 2 (BACK): POWER SUPPLY & DISTRIBUTION

- Power supply mounted here
- Clean power to all modules
- On/Off master switch

TONGUE & GROOVE JOINT
(Runs Left to Right)

BOARD 1 (FRONT): THREE COMPUTER MODULES
(Left to Right)

1 Raspberry Pi 4 (Browser)
2 Jetson Nano (Older)
3 Jetson Orin Nano (Newer)

DATA FLOW CHEAT SHEET

- HTTP Requests (you → ComfyUI / LLM) You send instructions.
- HTTP Responses (back to you) Results, images, answers.
- ComfyUI ↔ LLM (Jetson ↔ Jetson) RAG lookups and AI calls.
- Models / Data (via network storage) Models, embeddings, files.
- Everything talks over the network. No cables between the Jetsons except power and Ethernet!

THE NETWORK

All three computers are on the same local network (Ethernet / Wi-Fi).

Typical IP addresses (example only)

Raspberry Pi 4 (Browser)	192.168.1.10
Jetson Nano (Older)	192.168.1.20
Jetson Orin Nano (Newer)	192.168.1.30

DATA FLOW AT A GLANCE

You drive everything from the browser. ⟷ You send prompts. ComfyUI returns images. ⟷ ComfyUI asks the LLM questions (RAG). LLM provides answers. ⟷ LLM thinks, searches, and answers.