Coding Nexus · Follow publication

Member-only story

oLLM: How I Ran an 80B Model on My 8GB GPU

3 min read · Sep 30, 2025

Algo Insights Following

I didn't expect this. Running a 160GB model on a card with 8GB VRAM feels impossible.

Usually, you'd laugh and move on. But then I stumbled upon **oLLM** and thought, 'Fine, let's see.'

Turns out… it works.

oLLM is a small Python library. It's built on Hugging Face Transformers and PyTorch, but the trick is how it handles memory.

Instead of blowing up your GPU, it streams data from the SSD, offloads tasks to the CPU, and employs some clever attention tricks. No quantisation, just fp16/bf16.

So yeah — you can actually run **Qwen3-next-80B** or **GPT-OSS-20B** on a 3060 Ti.

. . .

What Makes oLLM Interesting

The cool part is how it does this.

Instead of attempting to fit everything into GPU memory, it **streams weights directly from SSD to GPU**.

KV caches? Those go to disk, too. It even allows you to offload layers to CPU RAM when needed.

A few recent updates (version 0.4.2) make it even better:

- **Faster and lighter:** `.safetensor` Files no longer eat up your RAM through `mmap`.
- **Bigger models are now supported:** Qwen3-next-80B runs with DiskCache support.
- **Speed bump:** that same Qwen3-next-80B manages ~1 token every 2 seconds — which is insane for its size.
- **FlashAttention-2 everywhere:** boosts stability and lowers memory usage.
- **Chunked MLPs:** split up big intermediate layers so they don't blow up your GPU.

It's like the devs found every little bottleneck and patched it.

. . .

Getting Started

I recommend a venv:

```
python3 -m venv ollm_env
source ollm_env/bin/activate
pip install ollm
```

If you want the source version:

```
git clone https://github.com/Mega4alik/ollm.git
cd ollm
pip install -e .
pip install kvikio-cu12   # adjust CUDA version
```

⚠️ Heads up: if you plan on using **Qwen3-next**, you'll need a special dev build of Transformers:

```
pip install git+https://github.com/huggingface/transformers.git
```

Slightly annoying, but necessary.

. . .

First Test: Llama-3 on Local GPU

Here's a quick example script I used. It's the "hello world" of oLLM:

```
from ollm import Inference, TextStreamer

o = Inference("llama3-1B-chat", device="cuda:0", logging=True)
o.ini_model(models_dir="./models/", force_download=False)
```

```
    # Offload layers if needed
    o.offload_layers_to_cpu(layers_num=2)
    # DiskCache helps with long contexts
    past_key_values = o.DiskCache(cache_dir="./kv_cache/")
    text_streamer = TextStreamer(o.tokenizer, skip_prompt=True, skip_special_tokens
    messages = [
        {"role": "system", "content": "You are a helpful AI assistant"},
        {"role": "user", "content": "List the planets in our solar system"}
    ]
    input_ids = o.tokenizer.apply_chat_template(
        messages, reasoning_effort="minimal", tokenize=True,
        add_generation_prompt=True, return_tensors="pt"
    ).to(o.device)
    outputs = o.model.generate(
        input_ids=input_ids, past_key_values=past_key_values,
        max_new_tokens=100, streamer=text_streamer
    ).cpu()
    answer = o.tokenizer.decode(outputs[0][input_ids.shape[-1]:], skip_special_toke
    print(answer)
```

The output is streamed token by token, just like you'd expect from ChatGPT. On an 8GB card. That still blows my mind.

How It Pulls This Off

Think of it like this: instead of keeping everything in one backpack (your GPU), oLLM spreads the load between your GPU, CPU, and SSD. A few tricks make this possible:

1. **Weights on demand:** Loads layer weights from SSD directly to the GPU one at a time.

2. **Disk-based KV cache:** Stores attention memory on SSD instead of GPU.

3. **CPU offloading:** Pushes heavy layers to RAM if your GPU gets cramped.

4. **FlashAttention-2: Maintains efficient attention without requiring large** matrices.

5. **Chunked MLPs:** Splits up giant layers so they don't overflow GPU memory.

The result? Huge models that used to require a $10,000 server can now run on a $200 card.

. . .

What You Can Actually Do With This

Running big models locally isn't just for bragging rights. Some practical uses:

- **Legal work:** Feed an entire contract or compliance doc into one pass.
- **Healthcare:** Summarise years of patient records without chopping them up.
- **Security:** Parse massive logs or threat reports offline.
- **Customer Support:** Scan historical chats to identify the most common issues.

Basically, anything where you don't want to lose context because the model keeps forgetting what came before.

. . .

Performance on My 3060 Ti (8GB VRAM)

Here's what I saw when testing:

ModelContextBaseline VRAMoLLM VRAMDisk (SSD)**Qwen3–80B**50k~190GB~7.5GB180GB**GPT-OSS-20B**10k~40GB~7.3GB15GB**Llama-3–1B**100k~16GB~5GB15GB**Llama-3–8B**100k~71GB~6.6GB69GB

That's not a typo. A **160GB model with 8GB of VRAM**.

. . .

What's Next for oLLM

The roadmap looks ambitious. Coming soon:

- **Gemma-3–27B** (Sept 30)
- **Voxtral-small-24B ASR** (Oct 3)

- **Qwen3-VL (vision-language)** (Oct 10)
- **Multi-token prediction** for Qwen3-next (R&D)
- Faster weight loading (R&D)

So yeah, this thing is evolving quickly.

AI Llm Llm Applications Python Ai Agent

Follow

Published in Coding Nexus

7.8K followers · Last published 1 hour ago

Coding Nexus is a community of developers, tech enthusiasts, and aspiring coders. Whether you're exploring the depths of Python, diving into data science, mastering web development, or staying updated on the latest trends in AI, Coding Nexus has something for you.

Following

Written by Algo Insights

3.4K followers · 6 following

Algo Insights: Unlocking the Future of Trading with Code, Data, and Intelligence. https://linktr.ee/Algo_Insights

Responses (7)

Bgerby

What are your thoughts?

Hassan Mohamed
Oct 2

Great article describes a great idea, but something important is missing: the speed. As we all know, there are couple of performance bottlenecks: speed difference between GPU-RAM and SSD, and transferring DATA across SSD/CPU/GPU. I think providing... more

68 Reply

Lyledg
Oct 3

This is fantastic, I've got a rtx4090 24gb GPU, so going to experiment with and even bigger models

11 Reply

Ling Li
Oct 4

I imagine you are probably going to struggle to get more than 5tokens per second..... Not particularly useful at that speed...

3 Reply

See all responses

More from Algo Insights and Coding Nexus

In Coding Nexus by Algo Insights

4 Open-Source Tools That Made Me Rethink My Dev Setup

I’ve been coding for a while now. Most of the tools we use every day… they’ve been the same for years. Editors, browsers, frameworks. Just…

✦ Sep 3 👏 450 💬 9

In Coding Nexus by Civil Learning

The Guy Who Let ChatGPT Trade for Him — and Somehow It Worked

You know how everyone says, “Don’t let AI touch your Money”? Well, someone on Reddit decided to ignore that.

✦ Oct 8 150 6

In Coding Nexus by Civil Learning

6 Open-Source AI Projects You Must Try (Agents, RAG & Fine-Tuning)

The AI world is complex right now. Every week there’s a new repo, a new framework, and a new promise that this one will change everything…

✦ Sep 18 660 5

In Coding Nexus by Algo Insights

How to Build Your Own AI Rig for Running Local LLMs (Gemma, Mistral, Qwen, GPT-OSS and Llama)

About three months ago, I realised I was utterly dependent on companies that didn't care about anything except power, Money, and control.

✦ Oct 8 354 12

See all from Algo Insights

See all from Coding Nexus

Recommended from Medium

Tosny

7 Websites I Visit Every Day in 2025

If there is one thing I am addicted to, besides coffee, it is the internet.

✦ Sep 23 5.2K 184

In Generative AI by Thomas Reid

Google puts another nail in the RAG coffin with URL Context Grounding

Eliminate model hallucinations when processing online data

✦ Oct 2 373 17

In CodeToDeploy by TechToFit - Master Your Life with Tech

I Tried Google's New AI Agents. It's a Gold Rush.

I spend my days deep in the world of AI, but every so often, something drops that makes me stop everything. This is one of those times...

✦ Oct 10 👏 164 💬 3

In Towards Deep Learning by Sumit Pandey

Why Everyone Will Want DGX Spark on Their Desk—Yes, Everyone

I just saw this picture today and was amazed, I've been waiting for this moment for a long time. (No, it's not Elon.) It's that tiny...

✦ Oct 14 👏 23 💬 7

Bogdan Ilyin

Denmark Just Triggered Putin's Worst Nightmare

Europe’s quietest country just made one of the loudest moves against Moscow’s war machine.

Oct 6 13.8K 172

Bytefer

A 0.9B Open-Source Model for SOTA Document Parsing: Outperforming GPT-4o and Gemini 2.5 Pro

Beyond OCR: Parsing text, tables, formulas & charts in 109 languages. Get SOTA document parsing that runs locally.

✦ Oct 17 64 3

See more recommendations