GPT-OSS-120B · ONE WORKSTATION

Faster and more faithful than llama.cpp and ik_llama.cpp at every prompt length tested

Designed and built by Isaac Levinsky

MCMLV1 LLC · independent inference engineering · numerical validation

A from-scratch inference runtime that serves OpenAI's 120-billion-parameter open-weight model from one desk-side workstation, measured head-to-head against two widely used open-source runtimes on the same machine, from 336 to 65,280 tokens.

18 / 18

speed results won (prompt reading, answer speed and complete answer, 6 prompt lengths)

100.0%

next-token agreement with OpenAI's original model across 2,113 held-out test tokens

+32%

prompt reading speed at short prompts vs the fastest rival

November 2026

THE SETUP

A 65 GB model on GPUs with 24 GB of memory

The model is far larger than the graphics cards. The runtime makes the GPUs, the CPU and system memory work as one machine, and it runs OpenAI's published weights exactly as released.

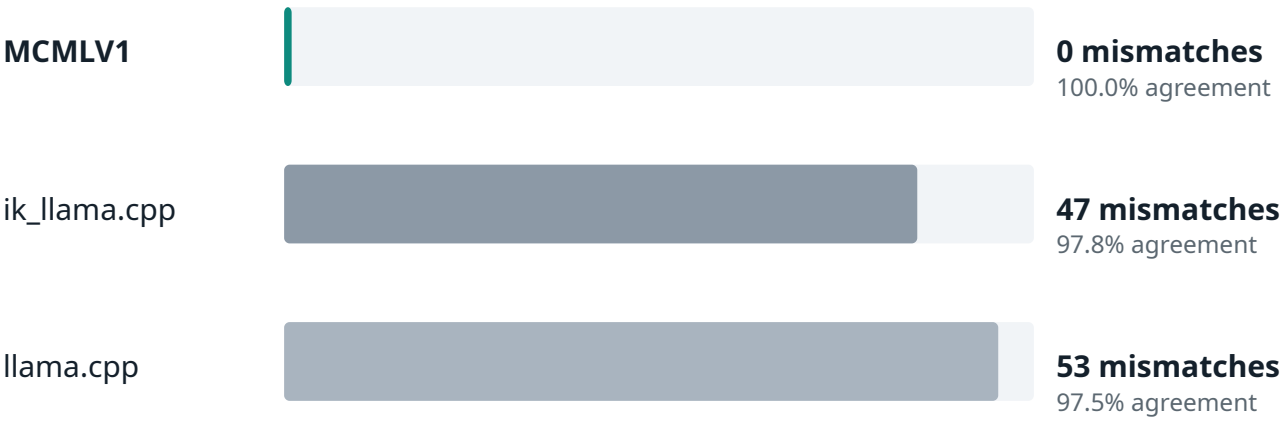
MODEL	OpenAI gpt-oss-120b (117B parameters, mixture of experts), original weights
GPUS	NVIDIA Quadro GP100 16 GB (2017) + Quadro RTX 4000 8 GB (2018)
CPU / RAM	Intel Core i9-10900X (10 cores) · 128 GB
RUNTIME	MCMLV1: an independently developed inference engine, built from scratch
RIVALS	llama.cpp and ik_llama.cpp, built from source for this machine and tuned for it

Same workstation, same prompts, same 256-token answers, one request at a time.

FIDELITY TO THE ORIGINAL MODEL

It chooses the same next token as the original model, every time

26 held-out prompts (coding, math, reasoning, knowledge, instructions, structured output, long documents), fixed before any test. At each of 2,113 answer tokens, does the runtime choose the same next token as OpenAI's original model? Tokens where it chose differently:



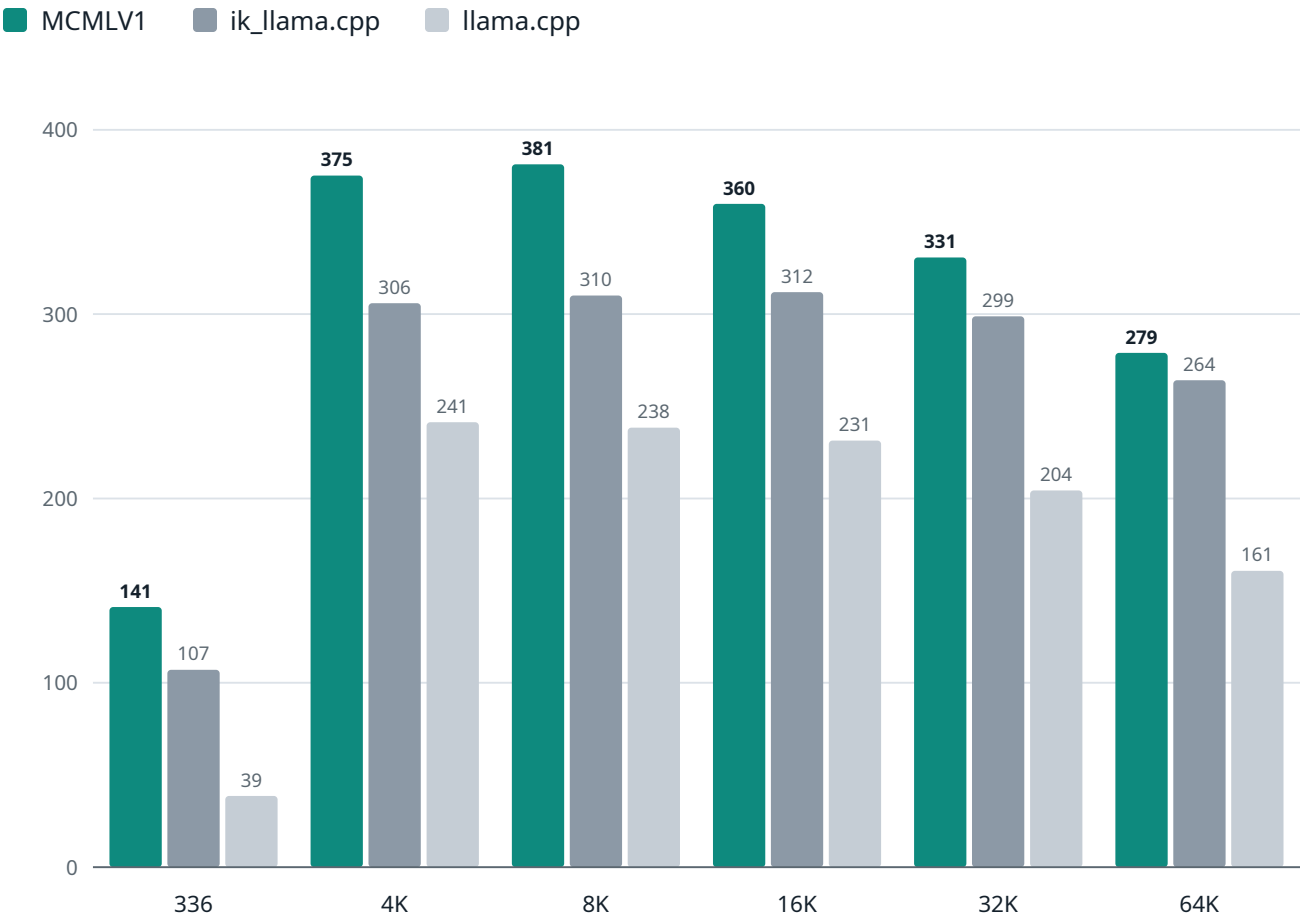
MCMLV1 matched **2,113 of 2,113**, in every category, with the same confidence as the original (mean probability 0.9374 vs 0.9374 for the original itself).

This measures faithfulness to OpenAI's model, not whether the model's answers are factually correct: a runtime can only be as right as the model it runs. Every runtime received identical input tokens.

PROMPT READING SPEED

Reads prompts faster at every length

Tokens per second while reading the prompt, from a short question to a 64K-token document. Higher is better.

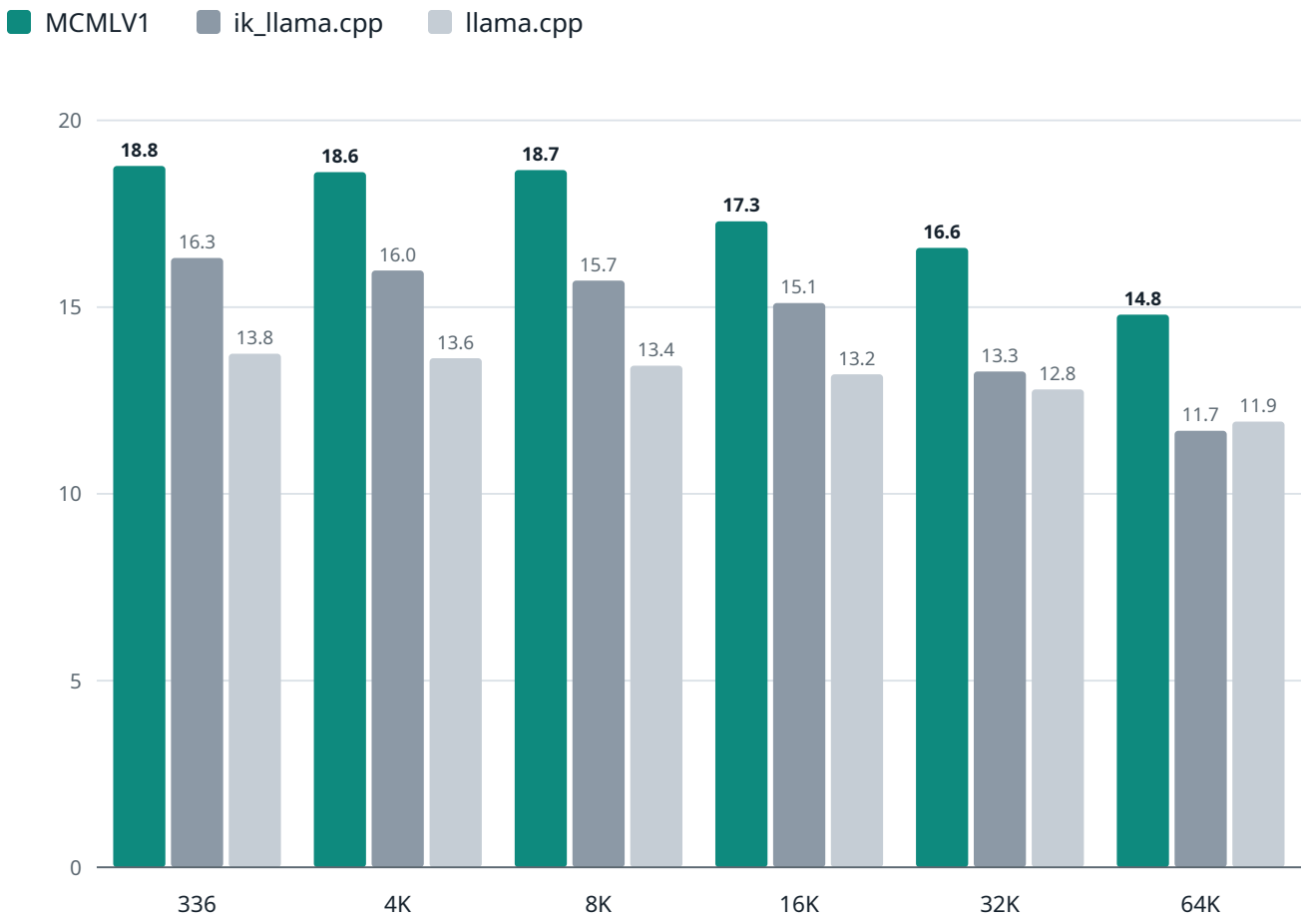


Ahead of the fastest rival by **32%** on short prompts and **23%** at 8K tokens.

ANSWER SPEED

Writes answers faster, and keeps the lead out to 64K context

Tokens per second while writing the 256-token answer. Higher is better.



At 32K and 64K tokens of context, answers come **25% and 24%** faster than the best rival.

COMPLETE ANSWER

Finishes first, from question to last word

Seconds from sending the request to receiving the complete 256-token answer, measured by the client. Lower is better.

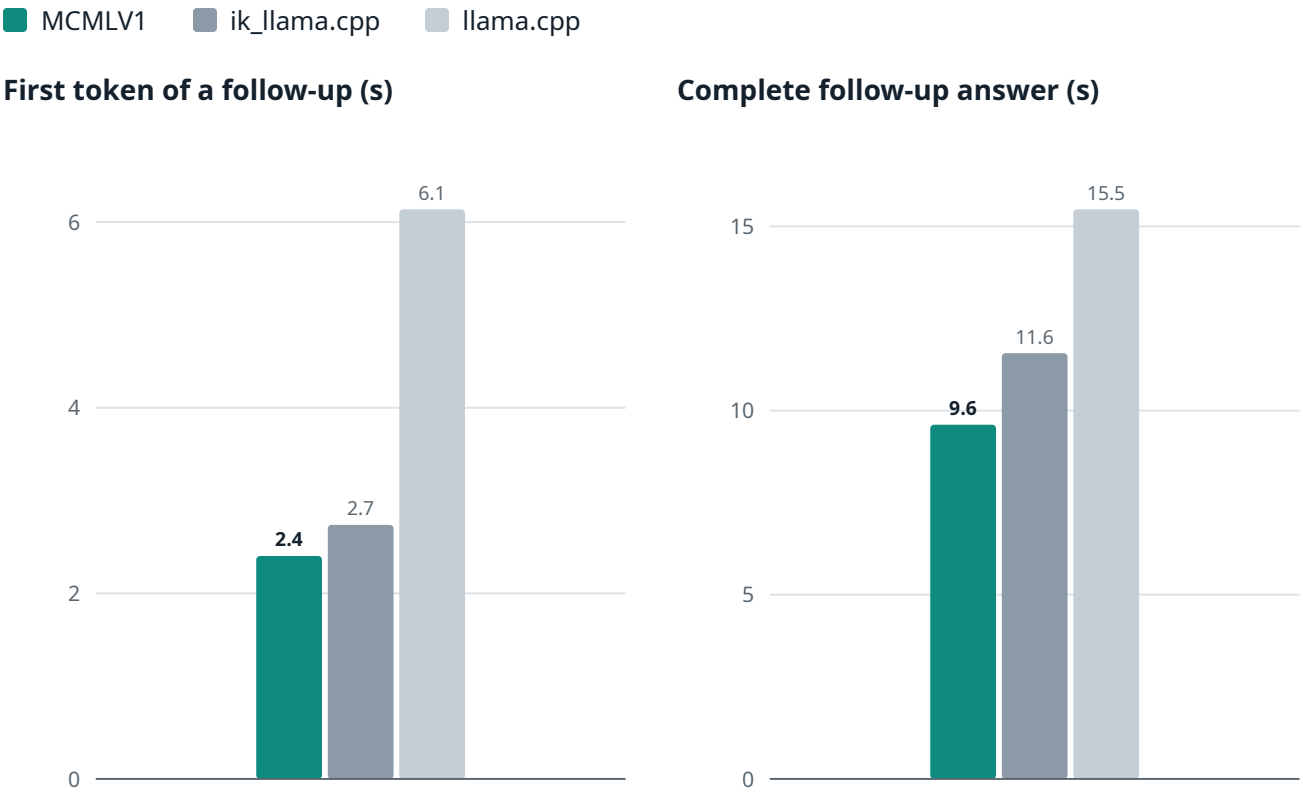
Prompt tokens	MCMLV1	ik_llama.cpp	llama.cpp	Time vs best rival
336	**16.0**	18.8	27.3	15% less
4,096	**24.6**	29.4	35.7	16% less
8,192	**35.2**	42.7	53.4	18% less
16,384	**60.4**	69.5	90.2	13% less
32,768	**114.7**	129.0	180.4	11% less
65,280	**251.8**	269.2	427.7	6% less

A 64K-token document (about 49,000 words) is read and answered in **252 s**, vs 269 s and 428 s.

ONGOING CONVERSATIONS

Continues long chats faster

Two chats, each opened with a 16K-token document, interleaved over three turns. All three runtimes reuse earlier context; this measures how fast each follow-up turn comes back (each rival at its best setting).



Lower is better. Averages of the four follow-up turns.

HOW IT WAS MEASURED

Same machine, same inputs, every number recorded

- Identical prompts, verified token by token: 336 to 65,280 tokens, 256-token answers, greedy decoding.
- Rivals built from source at pinned versions, given this machine's best offload settings and screened for their fastest batch size; all runs on an otherwise idle machine.
- Accuracy is measured against OpenAI's original model weights, not against any runtime, on prompts that were fixed and fingerprinted before testing.
- Automated real-task checks (code that must compile and pass tests) pass on every build.
- One measured run per size per runtime (two at 336 tokens); a repeat MCMLV1 run agreed within 1.2% at 4K to 64K.

Results are for this workstation; a technical brief with full tables and method is available under NDA.

BEYOND THIS MODEL

The skills carry over to other models and other hardware

GPU performance engineering
A custom execution engine built for this model and these GPUs

Memory-constrained serving
A 65 GB model served on GPUs with 24 GB of memory

Heterogeneous execution
Two GPU generations and a multi-core CPU working as one machine

Numerical validation
Fidelity verified against the original model, token by token

Benchmarking
Pinned rival builds, identical inputs, held-out tests fixed before testing

Production serving
A live API service with streaming, conversation memory and safe rollback

Inference engines built by Isaac Levinsky

GPT-OSS-120B	Quadro GP100 + RTX 4000 + CPU (this deck)
GPT-OSS-20B	NVIDIA RTX 3060, 12 GB
Gemma 4 26B A4B	MacBook, Apple M5, 32 GB
Qwen 0.6B · Gemma 4 E2B	Android phones

Speed and accuracy results in this deck are for the GPT-OSS-120B runtime only.

PRIVATE AI ON HARDWARE YOU OWN

A 120-billion-parameter open model, on premises, on hardware you own

Data stays in the building. Capacity is bought once, not rented by the token. Fidelity is verified against the original model before any speed is counted.

Interested in a live demonstration, a technical briefing under NDA, or inference engineering work? Get in touch.

Isaac Levinsky
MCMLV1 LLC

mcmlv1.com