Samuel Mausberg

GPU performance engineer: CUDA kernels, LLM inference engines, compilers

Full-time independent GPU work since Feb 2026; UBC mathematics coursework (on leave)

Vancouver, BC | US and Canadian citizen, open to relocation in the US and Canada | +1 (587) 216-8500

samuelmausberg@gmail.com | github.com/SamMausberg | linkedin.com/in/sam-mausberg | sammausberg.com

I make LLM inference faster on NVIDIA GPUs. My GH200 engine decodes a 27B model 1.49x faster at one request than mistral.rs and beats vLLM, SGLang and TensorRT-LLM. NVIDIA maintainers merged my fixes and API work into FlashInfer and CCCL, and my B200 kernels scored up to 4.46x over SOL-ExecBench's baseline.

Inference engines and serving

Independent LLM inference engine for one GH200, public benchmarks — Oct 2026

- Designed and built it for single-request decode speed: FP8 matrix kernels on Hopper, tree speculative decoding, recurrent and attention state management, and a vLLM worker plugin.
- Decodes Qwen3.8-27B-FP8 at 325 tok/s at one request, **1.49x faster than mistral.rs**, the fastest of four engines tested, and 2.07x faster than TensorRT-LLM, with output identical to its own non-speculative decoding.
- mistral.rs, vLLM and SGLang all ran DFlash2 speculative decoding; TensorRT-LLM ran its best of five configurations.
- On separate code, math and JSON prompt sets: 1.49–1.65x.

SGLang speculative decoding, Qwen3.5-4B on one GH200 — Oct 2026

- Stock SGLang snapshots about 48 MiB of recurrent state per draft position per request to verify speculative drafts. I replaced the snapshots with a replay of only the accepted prefix, bitwise identical to stock (SGLang #42209, open).
- With narrower Triton verify tiles and FlashAttention 4 also on: **+12.4–13.7% throughput** at 8–32 concurrent requests and +8.4–9.3% per-request output at 1–4 over tuned stock SGLang; greedy outputs match up to rounding.

Command A+ configuration study, vLLM on two H100 GPUs — Aug 2026

- Giving the KV cache the GPU memory vLLM left unused raised **KV capacity 1.79x** and end-to-end throughput 4–34% at each context's largest batch.
- Benchmarked the 218B mixture-of-experts model (25B active) with tensor parallelism across both GPUs over NVLink, in 91 runs from 512 to 128K tokens of context.

Merged upstream work

FlashInfer: fused-MoE crash fix for Blackwell RTX GPUs, merged (#5451) — Sep 2026

- A user-reported vLLM crash: padded batches with −1 expert ids sent the SM120 fused-MoE kernels into illegal memory accesses or wrong-row writes. I made every kernel path treat negative ids as unrouted.
- Added 35 regression tests; a maintainer tightened them, reran them on an RTX 5090 and showed 23 fail without the fix. **Shipped in FlashInfer v0.7.1**, and the maintainers extended the same rule to the FP8/MXFP8 backends (#5977).

NVIDIA CCCL (CUB): execution-environment APIs, merged (#11587, #11589, #11590) — Sep–Oct 2026

- Moved CUB's DeviceMergeSort, DeviceMerge and DeviceMemcpy APIs from a bare CUDA stream to CCCL execution environments, so callers can pass a tuning policy and memory resource.
- **Merged by the CCCL maintainer**, with 26 new tests and existing call sites unchanged; #11591 is approved.

FlashInfer: diagnostics API and bug finds — Sep 2026

- Added sparse-MLA diagnostics and a config API vLLM can call; merged in an NVIDIA rework with me as co-author (#4802).
- Showed the decode autotuner timed cache-resident inputs, skewing its results; the merge removed it (#4683).
- Found and reproduced a bug where ragged FP8 KV-cache prefill ignored its calibration scales; fixed upstream (#4984).

Kernels

NVIDIA SOL-ExecBench: official B200 evaluations (v1.1) Aug–Oct 2026

Ranked #1 on two kernels and #3 on a third (as of Oct 25, 2026); 110/110 listed workloads correct and faster. Speedups over the scoring baseline are geometric means; SOL scores 0.58–0.87 (1.0 = hardware limit).

- #1 attention backward: 4.46x; #1 MoE expert-parallel backward: 1.31x.
- #3 Mamba convolution with gating: 1.47x.
- Hyena depthwise convolution with split gate (FP32): 3.23x; fused ConvNeXtV2 MLP block (FP32): 2.52x.
- QKV projection with grouped-query reshape (BF16): 1.52x; block-scaled FP8 MLP gate/up with SwiGLU: 1.27x.

Open kernel pull requests: FlashInfer and CCCL (not yet reviewed) Sep–Oct 2026

- MLA decode on GH200, 1.43–2.02x faster than main at batch 1 (8–64 heads): swap-AB WGMMA tiles for small query counts (proposed in #2995) and a parallel split-KV merge that profiling flagged (FlashInfer #4906, open).
- FlashAttention 3 prefill for 16-bit queries with an FP8 KV cache: 1.28–1.36x paged and 1.61–1.69x ragged over FlashInfer's earlier FlashAttention 2 path (#4977, open).
- A large-value DeviceTransform fill reserved up to 17.2 GiB of local memory on GH200. Staging one item per thread instead of 16 removes that stack and nearly doubles fill throughput on an RTX 5070 Ti (1 KiB values; CCCL #11919, open).

Compilers, runtimes and GPU architecture

CAIRN: a systems language that compiles to C++20 and CUDA Sep 2026

- Traced the three failures in its first GH200 run, one an **NVVM 7.0.1 miscompile** dropping a loop's exit when checked arithmetic could trap. Reduced it to 8 lines, showed newer NVVM (sm_100+) is correct, shipped workaround #164.
- Designed the language to catch races and barrier misuse in GPU code at compile time.

FindTensor: ML compiler and runtime, independent startup project Feb–Sep 2026

- Designed and built the pipeline: torch.export import, Rust lowering, PTX generation and a C++ CUDA Driver API runtime.
- The v3 architecture served Llama-3.2-1B with greedy output bit-exact to the reference; a v4 rewrite started in Sep 2026.
- An earlier version ran a single-process multi-rank NCCL executor with CUDA graph capture.

GPGPU-Sim/Accel-Sim, Tor Aamodt's group, UBC ECE (directed studies, then volunteer) Sep 2025 – May 2026

- Added functional support for Hopper/Blackwell-era PTX (ldmatrix, cp.async and bulk copies, mbarrier, TMA, cluster fences), validated against H100, and bit-exact BF16/TF32/FP8/FP4 types.
- Split the mbarrier, TMA and WGMMA work with Tim Rogers' Purdue students; presented to both groups four times.

Skills

GPU: CUDA C++, PTX, CUTLASS CuTe DSL, Triton, WGMMA, TMA, mbarrier, FlashAttention 3/4, CUDA graphs, NCCL.
Inference: vLLM, SGLang, FlashInfer, PyTorch, speculative decoding (EAGLE-3, DFlash), KV caches, MoE, MLA.
Numerics: block-scaled FP8, NVFP4, bit-exact BF16/TF32/FP8/FP4 emulation, int8 screening with FP64 fallback.
Profiling and verification: Nsight Systems, Nsight Compute, CUPTI, Compute Sanitizer, Lean 4, Z3.
Languages: C++, Python, Rust, TypeScript. **Hardware:** GH200 (SM90), B200 (SM100), H100, RTX 5070 Ti (SM120).

Education

University of British Columbia, Faculty of Science: mathematics coursework; on leave Sep 2023 – present

Research

- *The Work a Verifier Needs: Certified Int8 Output Heads That Match a Production Kernel's Rounding*: int8 screening with FP64 recomputation and fallback; 0 mismatches over 592,433 positions checked while serving manuscript
- *Near-Linear Attention in Three Dimensions*: randomized approximate softmax attention in near-linear time manuscript
- *An I/O Lower Bound for Exact Attention*: memory-transfer lower bounds in a bounded deterministic model manuscript
- *Exact Sampling from Neural Networks: Depth, Attention, and Decoding*: theory, no GPU implementation manuscript
- All sole-author, with code in public repositories. More on sammausberg.com.

Other projects

- GPU MODE practice leaderboards, H100: prefix sum 2nd of 27 (held 1st Sep 2–30, 2026); histogram 3rd of 27 (Oct 2026).
- CPU references: tensor parallelism over TCP with clean fault stops, and SmolLM2-135M inference matching the reference.
- KernelIndex: GPU benchmark catalog in TypeScript, Next.js and PostgreSQL, with REST and CLI access.
- Vortex Consulting, intern (Jun–Aug 2025): Power BI and Azure DevOps reporting and a file-anonymization script.