Netanel Haber

netanel.t.haber@gmail.com · netanel.dev · github.com/netanel-haber · linkedin.com/in/netanel-haber

I like working deep in inference: KV-cache behavior, sampling, batching and scheduling, memory layout, and making models run both faster and more reliable. I used to be full-stack/MLOps and still enjoy building small tools that just work.

NVIDIA · Systems Inference Engineer (via Deci.ai acquisition) · 2024–current

Worked across nearly the entire inference stack: serving, sampling logic, model forward pass, KV-cache and inflight-batching, accuracy and performance evaluation.

- **Variable Sliding Window Attention**: Allocate minimal blocks per window size, redesigned KV-cache to support multiple attention windows efficiently, partitioned memory per window, schedulers became window-aware. Also solved underallocation in VSWA+/VGQA with smarter per-window block budgeting.
- **Unified sampling surface**: standardized token-output across runtime, speculative decoding, and samplers without slowing things down
- **FP8 enablement**: activated FP8 flows for Llama3.1-Nemotron-51B in ModelOpt quantizer (VGQA + NoOp attention + NoOp FFN + linear layers made this nontrivial)
- **Upstream model support**: added Nemotron NAS support and evaluation in SGLang (PR)

Deci.ai · Infra + Inference (acquired by NVIDIA) · 2022–2023

- Worked on **InferyLLM**: OpenAI-compatible server; cancellable generation; shipped end-to-end (frontend → Kubernetes → CUDA) to launch DeciLM-7B
- Preserved **TensorRT timing cache** at scale (hundreds to thousands of models) across Kubernetes nodes
- **One-click training visualization**: on-demand TensorBoard via K8s Jobs (S3-backed, FastAPI proxy, React iframe)
- Before that: fullstack MLOps, backend, and frontend for Deci's computer vision trainers

CardLatch · Frontend Lead (React/TypeScript) · 2020–2022

Built async CRUD UI, facility map navigator, 16-stream websocket camera viewer, and migrated the stack to TypeScript.

Earlier

3 years in the IDF as a metallurgy lab technician focused on failure analyses and producing metallographic specimens, including reports on 3D-printed metals.

Freelance and Fun

- checkers – vanilla TypeScript PvP/PvC with web workers, undo/redo, mobile, no runtime libs
- 8086/8 disassembler
- localfiles.stream – local media player PWA with some vibe
- use-easy-infinite-scroll (npm)
- my ChatGPT custom instructions

Skills

Python, C/C++, CUDA; TensorRT-LLM, VLLM, HF, SGLang; continuous batching
Kubernetes, Docker; tracing and metrics; performance tooling (nsys, Chrome perf)
Frontend when it is fun