Max Naidenov

maxnaid41@gmail.com | maxnaid.com | github.com/naidx0 | linkedin.com/in/max-naidenov | x.com/maxnaid | Toronto, ON

AI / ML engineer focused on the layer around the model: evaluation that cannot fool itself, retrieval and tool-calling agents people adopt, fine-tuning (LoRA, SFT, DPO) on one consumer GPU, and the harness that says whether a change helped before anyone believes it did. Python and TypeScript daily; PyTorch through peft and trl.

EDUCATION

York University **Toronto, ON**
B.Sc. Software Development *expected Fall 2028*

- Coursework: Data Structures & Algorithms, Object-Oriented Programming, Databases, Software Design, Operating Systems.

TECHNICAL SKILLS

Machine Learning: PyTorch via peft/trl, fine-tuning (LoRA, SFT, DPO), evaluation design (held-out sets pinned by hash, per-family scoring, noise floors, paired tests), LLM-as-judge instruments with preregistered checks, run records carrying prompt, model, metric and eval fingerprint, GPU inference profiling (VRAM, KV cache, TTFT, tokens/s), quantization (GGUF Q4/Q8)
AI Systems: Retrieval-Augmented Generation (RAG), embeddings (pgvector, ONNX), hybrid retrieval with rank fusion and cross-encoder reranking, agents and tool calling, MCP servers, prompt optimization measured against a baseline, guardrails and groundedness gates
Harness Engineering: Fact ledgers that separate measured from stated from asserted; gates that refuse a number without a witness; CI that asserts every discovered test ran; baselines scored before training data exists (196 rows pinned by hash, scored per family); LLM judges preregistered as instruments (one caught fabricating quotes 4 times in 10 halted a 1,700-call experiment by rule)
Languages: Python, TypeScript/JavaScript, SQL
Infra: Ollama and llama.cpp, Docker, PostgreSQL, Redis, Supabase, FastAPI, React, Node.js, REST APIs, CI gates, Git, cloud deployment (Vercel, Render), observability (request, token and latency logging)

EXPERIENCE

ML / AI Engineering Intern **Jun 2026 - Present**
SCHWAI *Toronto, ON (Hybrid)*

- Designed a prompt-optimization sandbox that tunes 4 internal AI copilots against a measured reward: eval banks scored by an LLM judge across 5 quality dimensions behind deterministic hard gates, anchored-diff edits, and train/validation/held-out test splits with a noise floor. It auto-rejected a tuning run that beat validation and regressed on held-out test, which is the failure a single split hides.
- Built **Project Brain**, SCHWAI's enterprise knowledge base (brain.schwai.com): Retrieval-Augmented Generation (RAG) over the company's own corpus with open-source LLMs on a FastAPI + React stack, PostgreSQL/Supabase, Redis, and background ingestion workers; adopted by engineering and non-technical teams for sourced, plain-English answers.
- Diagnosed a flat 'groundedness' score across two optimization rounds as a retrieval gap rather than a prompt problem, redirecting the fix to corpus ingestion and saving days of prompt iteration.

Software Developer Intern **Jan 2025 - Jul 2025**
Neo Financial *Calgary, AB (Hybrid)*

- Delivered 5+ production features on a fintech platform serving 1M+ customers using React and Node.js, through code review, automated tests, and staged releases.

PROJECTS

Sequence | A map of your software that keeps getting truer | trysequence.app **2026**

- **Architect and maintainer.** TypeScript pnpm monorepo (10 packages, 2,050+ commits): a scanner reads 5 languages into an architecture graph with evidence on every edge, a 15-tool MCP server, and a CI check that fails on design drift. Its coverage gate refuses to call a component absent when the scan never covered it: **64 accusations to 0 false positives** on a 1,124-file repository.
- Shipped as a desktop app a stranger can download (v0.1.0, Windows and Mac): open a folder, see the map, click an arrow, land on the line that proves it.

ML Harness | Training-decision layer: LoRA, DPO, evals (private, walkthrough on request) **2026**

- **Built to answer one question: should you train at all?** Python/FastAPI engine (65 routes) that runs evaluations through a tool API against a local model provider and ledgers every fact with its origin: MEASURED when an instrument produced it under the engine's watch, STATED when a person supplied it, ASSERTED when it arrives without a witness. An assertion opens no gate, so a model cannot mint a number a person is meant to supply.
- Scored the held-out baseline **before any training data existed**, so the bar an adapter must clear was fixed before the corpus that would produce it: **196 sha256-pinned rows** across 4 families, reported per family because pooling lets a model that memorised pattern names hide behind an average. Two full runs of the base model landed at **32.7% and 35.7%** (64 and 70 of 196).
- Measured a single instruction change lifting node-label F1 from a **53% baseline to 79%** on a 20-row held-out set, matching a 27B model's 81% on the same rows; a preregistered LLM-judge instrument caught the judge fabricating quotes 4 times in 10 and halted a 1,700-call experiment by rule.

Selvage | Hybrid-search knowledge layer | github.com/naidx0/selvage (MIT, public) **2026**

- **The one project here a reader can clone and run.** Team knowledge (Slack, wikis, code, databases) through a single PostgreSQL + pgvector store, an MCP server, a CLI and a web UI. Three retrieval signals because each misses what the others catch: full-text finds an exact error code, vector finds a paraphrase, rare-term finds the unusual token, fused with Reciprocal Rank Fusion and reranked on CPU via ONNX. Not deployed, so no latency figures yet.