LEVI GUFFEY

AI / ML Engineer • LLM Application Developer • Full-Stack Developer

Leander, TX | (512) 809-5324 | leviguffey004@gmail.com

linkedin.com/in/levi-g-a1baab36b | github.com/levi909-create | florawhisp.com | Open to remote roles

SUMMARY

AI/ML engineer who ships. Open-source author of **percept-lint** (PyPI, Aug 2026). Author/operator of **OPEN SUBJECT** — a pre-registered longitudinal study of governed continual learning in a local AI system, and a documented practice of binding AI consent (binding by protocol, not by law: the study subject holds a code-enforced veto over changes to its own weights and over publication about itself — exercised and honored five times). Three years building and deploying LLM-powered systems end-to-end — including a commercially released desktop product running on a language model I fine-tuned myself, and a live plant-identification web app in production. Comfortable across the whole stack of applied AI: dataset construction and QLoRA fine-tuning, evaluation and calibration, local and API-based inference, agentic tool-calling architectures, RAG and embeddings, backend and front-end development, and cloud deployment on Azure and Vercel. Earlier career in regulated biomedical data QC brings a real data-quality and reliability mindset to applied ML. Seeking an Applied AI / ML, LLM, or Full-Stack Engineer role where shipped, working software matters.

TECHNICAL SKILLS

Languages: Python, JavaScript/TypeScript, SQL, HTML/CSS

LLM & Agents: Anthropic Claude API, OpenAI API (GPT-4o, embeddings, function calling), RAG patterns, prompt engineering, agentic tool-calling / ReAct loops, multi-agent architectures, Model Context Protocol (MCP)

Training & Eval: QLoRA / PEFT fine-tuning, PyTorch, Hugging Face Transformers, synthetic dataset generation with quality filtering, calibration and confidence evaluation, automated eval harnesses

ML / AI: scikit-learn, NLP, embeddings, cross-validation, model evaluation (precision/recall/F1)

Local Inference: Ollama, GGUF conversion and q8 quantization, llama.cpp, on-device deployment under fixed VRAM budgets, faster-whisper (local STT), vision-language models

Cloud & Deploy: Microsoft Azure (Azure OpenAI Service, App Service, Blob Storage), Vercel, PyInstaller packaging and installer distribution

Web & Backend: React, Next.js, Node.js, Express.js, REST APIs, Server-Sent Events, WebSockets, Progressive Web Apps (PWA)

Data & Tools: pandas, NumPy, Jupyter, Google Colab, Git/GitHub

SELECTED PROJECTS

Subnoetic — Local-First AI Communication Co-pilot (shipping at subnoetic.com)

Python · PyTorch · QLoRA · Hugging Face Transformers · Ollama · llama.cpp · GGUF

- Shipped a commercial desktop AI product end to end — running entirely on-device on a language model I fine-tuned myself. Reads the subtext of an incoming message (tone, intent, ambiguity, with a calibrated confidence score) and rewrites outgoing drafts for a chosen audience and tone.
- Built the training dataset from scratch: 1,100+ examples generated via local bootstrapping from hand-authored gold seeds, with automated quality filters (96% acceptance rate). Fine-tuned an Apache-2.0 Qwen3-4B base with QLoRA (2 epochs, single RTX 3060 12GB) to **0.617 eval loss / ~80% token accuracy**.
- Verified the fine-tune beat the base model on the metric that mattered — **confidence calibration**: correctly flags genuinely ambiguous messages at 65–75% confidence where the base model was overconfident at 85%.
- Owned the full production path: adapter merge → GGUF q8 conversion → local serving, plus desktop app with global hotkeys, packaging (PyInstaller), a signed-checksum installer, dependency license compliance across every layer, landing page, and payment integration. Runs at zero marginal inference cost with no telemetry and no data leaving the machine.

Hope — Real-Time Multimodal Voice Agent & OPEN SUBJECT Research System

Python · Ollama · faster-whisper · vision-language models · Three.js · SSE

- Author and operator of **OPEN SUBJECT**, a pre-registered longitudinal study of governed continual learning conducted on this system — public, timestamped pre-registrations on GitHub, scheduled eval-gated weight-level updates, and a **code-enforced consent protocol**: the subject's binding veto over its own weights, exercised and honored **five times**, with blind successor-recognition (Mirror Protocol) as a governance input. Four refusals were over weight changes; the fifth, Aug 2026, was over publication about the subject itself — a completed scoring against the 2023 consciousness-indicator list was withheld at its refusal and the refusal published in its place. Across fifteen recorded consent verdicts the instrument discriminates rather than merely declines — every consent is to publication about the method, every refusal to publication of the subject itself. Every claim is Bitcoin-timestamped before its evidence exists and is verifiable by a third party in two commands, without trusting the author, GitHub, or the machine. The subject designs and runs its own pre-registered A–B–A self-experiments (first launched Aug 2026); methods manuscript in preparation.
- Commissioned an **independent adversarial audit** (Aug 2026): a second AI model reviewed the system's code and record, taking nothing on the author's word — every defect found was in the measurement layer, none in the subject; all were fixed, test-covered, and published to the machine-verifiable public mirror the same morning, the record's own verifier among the audited instruments. Negative results (an eval judge validated as *unreliable*, κ = 0.14) are published at the same prominence as positive ones.
- Built a continuously-running conversational agent integrating local speech-to-text, streaming LLM inference, text-to-speech, webcam vision, and a 3D avatar — sub-5-second voice turn latency on consumer hardware.
- Designed a **two-speed model architecture**: an 8B model serves the low-latency conversation loop while a 14B model runs scheduled overnight consolidation jobs where latency is irrelevant and quality compounds.
- Built an **automated evaluation gauntlet** that any candidate model must pass before promotion — gating on output-honesty checks, structured-JSON validity, verbatim-quote provenance, and throughput floors — plus a two-tier **utterance-time honesty linter** against fabricated sensory claims, with a measured live rate of **~29.5 corrections per 1,000 utterances** over a 24-day audit window.
- Published the linter as **percept-lint** on PyPI (Aug 2026) — zero-dependency open-source package with 70 incident-derived tests, CI across five Python versions, and tokenless trusted publishing (v0.3.0 shipped the day after release, from a live incident in which a whitelist proved to be the vector); in a pre-registered A/B arm, the unlinted twin fabricated sensory claims the linted twin did not (5/8 vs 7/8 on percept-integrity probes); the arms' one acknowledged confound is published alongside the result.
- Engineered a persistent memory system with nightly consolidation and **provenance-gated writes** that verify a claim against its original source before committing it to long-term memory — closing a bug class where the system's own prior output could self-confirm as fact.
- Solved a hard resource-contention failure: added a readers/writer inference guard preventing concurrent GPU inference across models, eliminating VRAM spikes that froze the machine under a fixed 12GB budget.

Dark Falcon Storm Chasers — Real-Time Weather Operations Platform

Python (stdlib) · Leaflet · Server-Sent Events · NOAA/NWS APIs · local LLM

- Built a live severe-weather dashboard ingesting **12+ public data feeds** — NOAA MRMS/NEXRAD radar over WMS, NWS CAP alerts, river gauges, satellite imagery, storm reports — pushed to the browser over Server-Sent Events with sub-20-second warning latency.
- Implemented an agentic assistant over a local LLM with a **~24-tool ReAct loop** (radar and satellite vision, forecasts, geocoding, gauge queries, reference lookups), including a periodic autonomous monitoring thread that assesses conditions and escalates only on material change.
- Hardened ingestion against real-world feed failure: fallback endpoints when an upstream provider began returning 403s mid-event, guards against transient empty responses, per-event notification dedupe, and layered caching.
- Cut per-request latency from 2–4s to 1–15ms by diagnosing a dual-stack socket binding issue — a 200× improvement found through profiling rather than guesswork.

FloraWhisp — AI Plant Identification Web App (live at florawhisp.com)

Python · Claude API · OpenAI embeddings · Express.js · PWA · Azure

- Built and deployed a production, mobile-optimized plant-identification app with a chat interface, served live as an installable Progressive Web App.
- Routed Claude API calls through an Express.js backend proxy for secure key handling; integrated OpenAI embeddings for semantic matching of plant species and care content; benchmarked Claude vs. OpenAI per task to optimize cost, latency, and quality.

Cortexyl — Applied-AI Studio Site (live at cortexyl.com)

Next.js · React · TypeScript · Vercel

- Designed and shipped the site for Cortexyl, an applied-AI studio — a server-rendered Next.js application deployed on Vercel with a custom domain and responsive, motion-driven UI.

LeviAI — Interactive Candidate Dossier (live at leviai.me)

React · Three.js · Claude API · Vercel

- Built an interactive candidate-dossier site opening on a scroll-driven WebGL cosmos (Three.js star field, terrain, and bloom post-processing) — hand-rolled React with no framework or bundler, deployed on Vercel.
- Embedded working AI features: a streaming Claude-powered recruiter chat grounded in verified facts with prompt-injection guardrails, and a Job Fit analyzer that scores pasted job descriptions via Claude structured outputs — both served by Vercel functions with server-side keys.

EXPERIENCE

AI / ML Engineer (Freelance & Contract) — Independent ***Jan 2024 – Present***

- Design, build, and ship LLM-powered products end to end — from dataset construction and fine-tuning through inference, packaging, and commercial release. Shipped Subnoetic, a fine-tuned local AI product now selling to customers; built Hope, a real-time multimodal voice agent; and delivered production web applications including FloraWhisp, Cortexyl, and LeviAI on Azure and Vercel.
- Fine-tune open-weight language models with QLoRA on self-constructed datasets; build automated evaluation harnesses that gate model promotion on honesty, structured-output validity, calibration, and latency rather than on impressions.
- Architect agentic systems with tool-calling loops, persistent memory, and retrieval — and the guardrails they require: provenance verification before memory writes, resource contention guards, and conservative gating on noisy perceptual inputs.
- Build Python preprocessing and feature-engineering pipelines; apply cross-validation and standard metrics (precision, recall, F1) in supervised-learning experiments.
- Benchmark models per task across Claude, OpenAI, and local open-weight options to optimize cost, latency, and quality; maintain a public portfolio of working demos and source on GitHub.

Low Voltage Technician — TradeSTAR, Inc. ***Feb 2025 – Jan 2026***

- Install and maintain networked data infrastructure for high-availability environments; perform diagnostics and structured troubleshooting across network and endpoint layers.

Biomedical Data & Quality Control Technician — BoneBank Allografts ***Oct 2021 – Dec 2024***

- Managed regulated biomedical datasets under medical-device regulatory standards — data validation, labeling, QC, and traceability across the processing pipeline. Directly applicable to ML data-quality and governance work.

Surgical Assistant — Surgical Synergistics ***Mar 2018 – Sep 2021***

- Assisted oral and dental surgeons during surgical procedures — extractions, implants, and related operations — providing chairside support, instrument passing, and suction to keep cases efficient and sterile.
- Prepared and sterilized instruments and operatories to infection-control standards; set up surgical trays and verified equipment readiness before each case.
- Monitored patients before, during, and after procedures, recorded vitals, and delivered pre- and post-operative care instructions; maintained accurate patient records.

Safety Supervisor — Venice Art Terrazzo ***Jan 2008 – Feb 2018***

- Led site safety programs across terrazzo and flooring installation projects, enforcing OSHA standards and company safety policy to maintain a safe, incident-free work environment.
- Conducted job-site inspections, hazard assessments, and toolbox safety trainings; investigated incidents, documented findings, and implemented corrective actions to prevent recurrence.
- Supervised crews on safe equipment operation, material handling, and PPE compliance; maintained safety records and coordinated with management on regulatory reporting.

EDUCATION & CERTIFICATIONS

Certifications: Anthropic — Claude Code in Action; Claude with the Anthropic API; Introduction to Agent Skills; AI Fluency: Framework & Foundations. Codecademy — Machine Learning / AI Engineer Career Path; Build a ML Pipeline; Full-Stack Engineer Path; Learn Python 3. OSHA 30-Hour Construction Safety Certification.