Nguyen Quoc Huy (Leo)

AI Engineer | AI Agents & Evaluation | Production ML Systems | Medical AI | Software Engineering

nguyenhuy190303@gmail.com | linkedin.com/in/nguyen-huy-1903 | github.com/NguyenHuy190303 | github.com/MooSun-Leo | Vietnam

Summary

AI engineer whose strongest area is designing AI agents, evaluating them and keeping them reliable when users and inputs are unpredictable; started the company's LLM evaluation platform and brought in Langfuse tracing. A software engineer first (backend, then full-stack, then AI) who understands software from algorithms and memory up to distributed systems, and model behaviour from architecture down to the training-data distribution, so reaches for the fix the problem needs rather than an LLM by default. Hands-on across computer vision, time series, tabular ML, deep learning, LLM post-training and agents, and has worked in this direction since before joining: in mid-2025 rebuilt a healthcare assistant on a single-threaded agent for reliability, and wrote up a hybrid multi-agent design and a medical-LLM plan (SFT then DPO) later carried out on 2×8×H100. Production results include a real-time proctoring service at 5.6× throughput (30 concurrent exam sessions for 3 hours, 99.99% of frames served, zero restarts) and a 6,400-hour Vietnamese speech corpus. Reads people and the overall situation quickly, which makes coordination and guidance work across product, QC, software and AI teams.

Core Strengths

- **Agent design, evaluation and reliability:** designs agent flows to fail safely, then tests them before users do (6 prompt variants × 30 simulated patients with red-flag detection, a 65-case agent test suite, red-teaming for jailbreak and prompt injection, judge metrics) and hardens the runtime against loops, key-leaking request overrides and forged context.
- **Software engineering first, the right tool for the problem:** backend, then full-stack, then AI. Fixes problems at the layer where they live: fixed input shapes instead of a new model for 68% frame loss, an API-layer bug behind a 23.8% TTS error rate, one client per process for connection-pool exhaustion, and a decision between two teams instead of more code.
- **Big-picture view and coordination:** maps who is doing what and where work overlaps, then turns it into the few decisions that unblock everyone (team catch-ups, a 13-module AI platform map, handover audits, cross-team gap reports, 77 scoped Jira work items, production-readiness standards).
- **Understanding people and growing them:** mentors and pairs with teammates, adapts guidance to the person, and turns what one person learned into something the whole team can reuse (6 of the 15 items in the team's shared agent playbook).
- **Context sync and communication:** written decision records, handovers, specs and explainer pages; seminars for engineers; demos for hospital and government stakeholders; explains AI results to non-technical people in plain language.
- **Domain plus technical depth:** clinical workflows (history-taking, ICD-10, HIS/EMR), legal-exam proctoring and personal-data consent, together with hands-on computer vision, time series, tabular ML, deep learning, LLMs, serving, data platforms, streaming and Kubernetes.

Experience

AI Engineer Sep 2025 – Present

Trivita AI — healthcare and legal-tech AI Vietnam

AI agents: design, evaluation and reliability — LangGraph, LiteLLM, Langfuse, FastAPI, Celery/Redis, Postgres, DeepEval, DeepTeam, Next.js

- Started the company's self-hosted LLM and agent evaluation platform and built the first working version single-handedly (API, async pipeline, red-team engine, CLI, Docker, 57 tests); now developed by 3 engineers. Proposed Langfuse for tracing and prompt management and integrated it into the eval platform; the agent service adopted it as well. Set the team rule of *eval before training.*
- Built the simulation harness for the MedVita medical agents: 6 system-prompt variants × 30 simulated patients (180 conversations) with red-flag and stop detection and a transcript viewer; fixed role bleed between simulated patient and agent. Wrote the dialogues for MedVita's 65-case agent test suite (safety, truthfulness, prompt injection, out-of-scope requests, value conflicts, edge cases).
- Built a runtime hardening patch for the agent service after an incident in which a model retrying a failing tool could loop for thousands of calls: a 25-model-call cap per run with a graceful reply, rejection of request fields that could redirect model calls or leak the API key, stripping of user-forged "verified context" markers, and no raw errors sent to users.
- Designed medical G-Eval and LLM-as-judge metrics (two-layer summary hallucination, empathy, emotional intelligence), conversational eval and red-teaming (jailbreak, prompt injection, PII leakage); added MedHALT and lm-eval runners with a 35B MoE judge on H100; built the SOCRATES history-taking eval (8 models × 9 scenarios).
- Fixed Langfuse tracing at the source (one trace per conversation turn instead of 5–20 fragmented root traces; one connection pool per process) and migrated it to Langfuse v4.

Real-time AI exam proctoring — service owner since Jul 2026; FastAPI, Triton, LiveKit, Kafka, Kubernetes

- Took ownership of the proctoring service (face verification, gaze, pose, object, voice and keystroke signals) and improved throughput from **21 to 118.7 frames/s (5.6×)** with ~30% lower per-frame latency.

- Passed a **3-hour, 30-concurrent-candidate load test**: 0 pod restarts, 1,619,905 of 1,620,000 frames answered (99.994%). Sized production autoscaling at 6–10 pods (≈297–495 frames/s against a 150 frames/s requirement).
- Root-caused frame loss in a live exam (68% of frames dropped at 10% CPU): dynamic input shapes made two detectors take 1.6–3.0 s per call vs 13–33 ms for fixed shapes, and Triton dynamic batching ran ~44× slower than batch-1. Fixed-shape inputs cut load-test frame drops from **97% to 11–15%**; cold start 83 s → 10 s.
- Wrote the service's architecture and scale review: traced a >200× per-frame payload amplification (29 KB → 7 MB), cut uplink traffic 73% by sending 8-bit integers (verified to leave results unchanged; enabled in production), measured the single-process ceiling under Python's GIL (~1.25 cores), sized 1,000 concurrent candidates (~300 pods, ~1,200 vCPU), and flagged the missing GPU metrics and alert rules.
- Built a labelled video benchmark and made it a ~1 s CI merge gate; recalibrated alert thresholds from **7% to 86% agreement** (20/20 clips at the correct alert level) and fixed 5 classes of false alerts that blocked candidates.
- Built voice anti-cheat end to end across 4 repositories (merged Sep 2026): LiveKit track egress → Kafka → VAD-gated speaker verification against enrolled voiceprints, at +0.34 Mbps and ~9 ms CPU per audio-second per candidate.
- Added Sentry, Prometheus/Grafana and structured logging; fixed DB connection-pool exhaustion and automatic video recovery after network loss; introduced decision records and trunk-based GitOps deploys with ArgoCD; on call for the live pilot exam.

Medical AI: MedVita, clinical data and post-training — vLLM, SFT/DPO, LoRA/QLoRA, DVC, MinIO, Label Studio

- Built **MedVita**, a Vietnamese medical history-taking assistant: extraction and normalisation of symptoms, labs, diseases and drugs to **ICD-10 VN**, negation and family/past-history context detection, SOCRATES-structured history-taking per department, P1–P5 safety guardrails; integrated with a central hospital's HIS/EMR with idempotent writes and audit logs.
- Built the real-world medical data pipeline: EMR/HIS and multi-dialect audio ingestion, PII/PHI redaction before any prompt, Label Studio golden batches, versioning on DVC + MinIO + Hugging Face Hub, MinHash dedup and contamination checks.
- Generated ~**1M synthetic patient personas** matched to national census (GSO) distributions × scenario × risk profile, batched through vLLM with prefix caching and CPU offload.
- Co-built the post-training loop (eval → targeted data generation → SFT/DPO with LoRA/QLoRA) on a **2×8×H100** on-prem cluster with 24–48 h iteration cycles; served models with vLLM tensor parallelism.
- Ran a 20+-arm prompt/harness ablation for a clinical-note agent (21 real pediatric visits, blind LLM judging): spec redesign lifted open-weight Gemma-4-26B from **0.60 to 0.80** and showed the golden reference itself scored only 0.78.

Vietnamese medical speech (ASR / TTS) — data, evaluation and serving

- Built a **6,448-hour, 1.98M-segment** Vietnamese speech corpus from 47 sources with a streaming build–upload–prune pipeline and licence gate; storage 521 GB → 187 GB.
- Designed a stratified benchmark of the production speech-to-text API (15k clips from 83k across 4 domains, bootstrap CIs): server-side denoising raised WER on every tier (clinical **15.4% → 11.0%** with it off); WebM/Opus 32 kbps is 8× smaller than WAV for ≤0.9 WER points.
- Wrote the speech evaluation spec with evidence-gated release verdicts (pass, pass with known risks, block, cannot certify): a gate without minimum evidence cannot pass, and error rates alone cannot raise severity above S2.
- Root-caused a text-to-speech regression to an API layer silently dropping the reference transcript; fix plus guard took production round-trip WER from **23.8% to 0.6%**.
- Supported coordination of the speech-annotation QC workflow (golden batches, inter-annotator agreement; authored 66% of the 117 gold diarization annotations); built the healthcare ASR release gate (40k-variant robustness suite) and serving metrics.

Data platform: data-factory — company dataset registry, v0 accepted Oct 2026; Python, PostgreSQL, MinIO/S3, MLflow

- Designed and built the company's dataset registry: stable dataset IDs, immutable snapshots on MinIO, pinned releases and aliases in PostgreSQL, and an SDK that resolves `name@release` to sha256-verified files and logs the pin to MLflow, so any training run can be re-run on exactly the same data.
- Built sample-level lineage for personal data from first ingest, with a `trace` command for consent withdrawal. Surveyed 4 existing pipelines and the infra conventions before writing code; accepted v0 on the on-prem H100 cluster with a real consented speech snapshot.

Engineering standards and production readiness — GPU infrastructure, speech serving, proctoring, eval platform

- Proposed the GPU infrastructure stack to the director: DCGM GPU metrics, Prometheus and Grafana (all three adopted as the company's GPU monitoring stack), Slurm for job scheduling (the infra team evaluated the options and adopted SkyPilot for that role), and Airflow for pipeline orchestration.
- Defined the production-readiness standard for the speech-serving API as 7 work items: rate limiting, retry/backoff and circuit breakers, an API contract with model metadata, dynamic batching, alert rules routed to Slack, metrics, and secret rotation with access audit. Followed up with a readiness review that found production metrics scraped on the wrong port (so alerting saw the target as down), CI running no tests, and personal health data in object storage with no lifecycle policy.
- Built the speech API's client error-code contract, Prometheus metrics and Grafana dashboard (18/18 error cases verified on staging) and fixed the Helm chart so Prometheus scrapes the Ray metrics port (production had collected no serving metrics for 14 days).

- Made lint a CI merge gate for the proctoring service (cleared 18 findings first), joined CODEOWNERS for speech serving, and review and merge teammates' PRs on the eval platform (e.g. the fix for API pods OOM-killed during 78k-sample runs).
- Onboarded 3 speech repositories to the company's docs-sync pipeline (shared GitHub Action to Confluence) and wrote their docs indexes, data-source catalogue, benchmark and incident pages, so other teams can find and reuse the work.

Team enablement and shared learning — Koruvai (the team's shared AI-agent layer), Claude Code, Codex

- Authored **6 of the 15 published items** in the team's shared agent playbook (from 23 proposals): the Sentry MCP connection, guardrail hooks that stop agents pushing to main or hand-editing lockfiles, a debugging skill built on Sentry and Grafana, an HTML report and presentation skill, and a Vietnamese technical-writing skill.
- Audited 30 days and 515 sessions of agent logs for token, context and cache cost, and published a team note on cutting agent spend (cache writes were ~46% of cost, so the main lever is not re-sending large context); published the team's plugin setup guide and an explainer page on how shared notes and skills flow between teammates' agents.
- Mapped the company's AI platform into **13 modules with a contract per module** (model gateway, prompts, tools/MCP, memory, eval and more) plus the data-flywheel gaps, first wave checked against the code, so software engineers on other modules can integrate without waiting on the AI team.
- Built a Jira-based team catch-up (per person: problem, why it matters, status, risk) and used it to show that two teams were building the same three components with no decision on which is canonical, plus the three decisions to settle before a new 6-person R&D team started coding.
- Co-proposed, with the principal AI engineer, the proposal to adopt Claude Team for engineers (phased rollout starting at 10 seats).

Cross-team work

- Supported coordination and quality assurance with product owners, QC and software engineers: turned product requirements into acceptance criteria and triaged QC-reported bugs. Created **77 Jira work items** across 3 projects, picked up by 10 teammates.
- Broke the company's data-privacy, security and AI-regulatory compliance programme into 13 work items under one epic, all 13 closed: DPIA, DPA template, data-flow maps for 3 products, a design-boundary document showing the assistant is not Software as a Medical Device, consent and withdrawal flow, audit logging per Circular 46/2018, encryption in transit and at rest, backup and restore drills, 72-hour breach reporting, and Level-3 information-system registration under the Cybersecurity Law.
- Reviewed the exam platform end to end and raised 14 product gaps in one pass (consent for face data, licence verification, appeals flow, multi-monitor detection, real-time support, official result certificates), all closed. Scoped MedVita's speech modules (symptom dictation, doctor/patient consultation transcripts, spoken replies) and infra resource governance (local-first workflow, ownership, policy, monitoring).
- Planned work for two interns and mentored one to independent delivery (4/4 handed-off tasks completed: diarization pipeline, Langfuse-integrated eval UI, eval landscape survey, ASR/TTS labelling) and wrote the intern's evaluation; review code and help teammates debug across the AI team.
- Supported the speech-team handover audit (112 tickets, 158 model/dataset repos); ran an internal seminar for 15+ engineers on agent-assisted engineering; technical contributor to 3 company patent filings and to hospital and government demos that supported a national-level research project.

Research Before Industry

- **AI Researcher**, *AIMA* (Jun 2024 – 2025): medical time-series forecasting and anomaly detection with doctors and clinical experts.
- **Multi-Agent VQA Research Team Lead**, *VLAI, remote* (2023 – 2024): multi-agent visual question answering; experiment design, task coordination, progress reporting.
- **Time-Series Research Team Lead**, *AI VIET NAM AI Residency, remote* (2022 – 2023): 4-person team building deep learning models for medical time-series data; mentored junior members and set milestones.
- **Project Lead, Mental Health Support System**, *VAMK*: RAG-based (LlamaIndex) system built by an international team of 4; perfect score (5/5) from the university evaluation committee. Later rebuilt as **SAGE** (github.com/NguyenHuy190303/Salus-Analytica, Jul 2025) on a single-threaded linear agent, chosen for reliability and context integrity over multi-agent complexity (FastAPI backend, crisis-detection and safety-compliance layers).
- **AI Intern**, *DzStar Digitals* (2023): agent and automation prototypes integrated with Google Cloud and AWS.

Publication & Community

- Nguyen Huy, et al. *Convolutional neural network-based artificial intelligence for immune subtype classification in pan-cancer.* Presented at the American Association for Cancer Research (AACR) Annual Meeting.
- Co-founder, **AIO Learners Hub** — AI learning community of 900+ members.

Projects & Competitions

- **VLSP 2026 shared tasks** (Vietnamese speaker verification, expressive TTS, speech-quality assessment): ensemble quality predictor (HuBERT, XLS-R, WavLM; 0.669 vs 0.657 best single model); best-of-N reranking for non-verbal TTS (0.90 recall / 0.87 precision).
- **EEP** (github.com/NguyenHuy190303/EEP): open-source agent-skills plugin (12 skills) for Claude Code and Codex with its own eval harness. **vllm-autotuner** (github.com/MooSun-Leo/vllm-autotuner): finds the fastest vLLM serv-

ing config on multi-GPU hosts.

- **AIO 2024 coursework and personal projects** (github.com/NguyenHuy190303): tabular ML (credit-card fraud, heart disease, housing prices), computer vision (YOLOv10 helmet-detection fine-tune, stereo depth estimation, image retrieval, diffusion colourisation, hand-gesture light control), audio generation (conditional VAE), big-data processing (Spark, MongoDB).
- **Capybara** (founder, side project, 2025–): multi-agent learning platform with a content scraper and AI tutoring.

Skills

- **LLM, post-training & eval:** LLM fine-tuning (SFT, DPO, GRPO, LoRA/QLoRA; Unsloth), synthetic data, RAG (LlamaIndex, LangChain, FAISS), agents and tool calling, LLM-as-judge, G-Eval, DeepEval, DeepTeam, lm-eval-harness, Langfuse, W&B, MLflow, red-teaming; clinical NLP (medical NER, ICD-10 VN / RxNorm normalisation, negation and context detection, relation extraction).
- **Training & serving:** PyTorch, TensorFlow, distributed training (DP/TP/PP, ZeRO, FSDP; Megatron-LM), vLLM, TGI, TensorRT(-LLM), Triton Inference Server, ONNX Runtime, Ray Serve; 2×8×H100 on-prem, GB200 NVL72 (partner access).
- **Vision & speech:** computer-vision model training and fine-tuning, YOLO (detection, pose), InsightFace (SCRFD, ArcFace), MediaPipe, OpenCV, NeMo, Whisper, pyannote, Sortformer, speaker verification (CAM++), Silero VAD, ffmpeg/Opus, WER, DER and cpWER evaluation.
- **Backend & streaming:** Python, FastAPI, Pydantic, Celery, Redis, PostgreSQL, SQLAlchemy/Alembic, Kafka, LiveKit/WebRTC, WebSocket, gRPC, MinIO/S3, DVC, Dagster, Airflow, Hugging Face Datasets, pandas, SQL, C.
- **Cloud-native & CI/CD:** Docker, Kubernetes (HPA, KEDA, Ingress, RBAC), Helm, ArgoCD (ApplicationSets, GitOps), GitHub Actions, trunk-based development, Infisical, Cloudflare Access, SkyPilot, Slurm, GPU sharing (HAMi), pytest, Playwright, ruff, pre-commit, uv.
- **Observability:** Prometheus (ServiceMonitor), Grafana, Loki/VictoriaLogs, Sentry, OpenTelemetry, structlog, DCGM GPU metrics.
- **Frontend:** TypeScript, React, Next.js, Tailwind, Zustand.

Education & Certifications

Bachelor's degree, Information Technology, *Vaasa University of Applied Sciences (VAMK), Finland* 2021 – 2025
Software engineering first: started in backend, then full-stack web (first full-stack project: a Twitter clone), then moved into AI.

AIO 2024 certificate (GenAI & LLMs, CV & NLP, DL & ML), *AI VIET NAM* 2024 – 2025

CS50x, CS50 AI, Using Python for Research (HarvardX, 2024) • NASA Space Apps Challenge 2024 — Galactic Problem Solver
Languages: Vietnamese (native), English (C2), Swedish (A2).