June Kim

Vancouver, Canada • june@june.kim • www.june.kim • 604 356 1191 • github.com/kimjune01

Summary

Evaluation engineer. I audit frontier coding benchmarks for construct validity, finding where the headline metric measures something other than what it claims, and I ship the receipts as preregistered, re-runnable artifacts with public repos and DOIs. 10+ years of software engineering at Google, Loom, and startups. 111 merged PRs across 89 external open-source repositories in Rust, Go, C++, and Python is how I read a benchmark's test suite in whatever stack it ships.

Selected Research & Systems Work

Independent Evaluation Engineer — Frontier Coding Benchmarks *2026 - present*

- SWE-bench Pro, determinacy audit (flagship): audited all 728 tasks; proven 15% underdetermined floor, 3 gold patches fail the benchmark's own verifier. Pro is what OpenAI now recommends over Verified. Preregistered and DOI-archived; findings filed with the maintainers.
- SWE-bench Pro, harness run: 95.3% (694/728) under the official grader, preregistered and frozen, every verdict re-gradable from a committed diff. Solo, on a $200/month plan.
- ProgramBench: showed the "% Resolved" metric scores recall of published algorithms, not source-blind reconstruction; 21+ programs gated on recalling a hash, cipher, or codec. Filed with the authors as a right of reply.
- DeepSWE: applied each reference solution to its own verifier; 4 of 113 fail. Under $1, under an hour, preregistered two-pass protocol.
- SWE-bench Verified: 426/500 (85.2%), denominator reconciled instance by instance; flags the set as contamination-compromised.
- SWE-rebench: determinacy audit, 14.5% pointer-checkable claimable spine. Built determinacy, the reusable auditor behind it and the Pro audit.

Work Experience

Independent Contractor *Applied AI Engineer* 2025 — 2026

- Anyteam.com: Designed and shipped an accessibility data pipeline for the Sales OS and a browser-extension web scraper to ingest and retrieve domain knowledge.
- Buildbetter.ai: Built third-party integrations (Circle.so, Notion, Front, Attio) with incremental sync, OAuth2, and deduplication, and an LLM field-classification system that maps import columns with confidence scoring.
- Shipped CI/CD quality gates, E2E test infrastructure, and reusable developer tooling (Claude Code skills for log querying, migration gen, CI failure analysis).

Little Bird Software *Applied AI Engineer* 2024 — 2025

- Designed agentic data ingestion pipelines using LLM-based condensation (Claude, GPT-4, Gemini Flash) and deduplication to cut noise 90%, improving retrieval accuracy and chat grounding.
- Architected macOS and Chrome integrations (Swift, Tauri, Rust) over Accessibility APIs for real-time context injection, and core Python backend services for prompt orchestration and dedup with zero-downtime migrations.

Loom *Senior Software Engineer* 2022 — 2023

- Raised core video reliability from 97% to 99.7% via multiresolution UI and Shaka Player interfacing; led a full TypeScript refactor of the Electron desktop app.

YouTube / Google *Software Engineer* 2019 — 2022

- Re-architected the YouTube iOS app's rendering layer (C++ / TypeScript), cutting UI deployment cycles from months to days.
- Directed the launch of a Premium sign-up framework, a 2% conversion lift across 50M+ users.

Earlier Experience *Software Engineer* 2013 — 2019

- iOS / mobile (2013–2019): Firework (patented video-view tech), Lipsi (#1 Lifestyle, US App Store, 2.3M users), and others.

Independent Research & Open Source

Published Research

- The Hypothesis Graph, Verifiable Knowledge, and What Cannot Be False Cannot Be True (2026): DOI-archived preprints with reproducible artifacts; full record and code at june.kim.
- Adversarial review loops push test-passing LLM code from 43% to 91% merge-readiness; deployed as 101 real-maintainer PRs.
- Methodology in public: a 22-question preregistration checklist, a published null result, and a post-mortem of a $1,000 mistake caused by held-out-test leakage.

Open Source Software 2026

- Enzyme autodiff compiler: a proof-by-cases soundness gate reproduced two compiler bugs from structure alone (filed upstream); separately landed two autodiff fixes (merged).
- Representative merged fixes: godotengine/godot, hyperium/hyper, envoyproxy/envoy, servo/servo, pingcap/tidb, EnzymeAD/Enzyme, flux-rs/flux, wild-linker/wild.

Education

Bachelor of Science (2nd degree) Simon Fraser University, Canada 2015—2017

- Second degree in computing, after a prior business degree; focused on machine learning, systems, algorithms, databases, security, and computer vision.

Bachelor of Business Administration Simon Fraser University, Canada 2008—2012

- Business administration degree focused on product, operations, strategy, finance, and entrepreneurship.

Skills

- Evaluation: benchmark auditing, construct validity, LLM evaluation, red-teaming, evaluation harness design, data contamination / decontamination, ground-truth and label-quality audits, preregistration, reproducible artifacts
- AI/ML: agentic workflows, tool use / function calling, RAG, embeddings, vector databases, Model Context Protocol (MCP), Claude Code
- Languages: Python, Rust, TypeScript, C/C++, Go, Swift
- Infrastructure: FastAPI, CI/CD, Git/GitHub