

# THOMAS TO

Oakland, CA, 510-387-5408, [thomas.to.bcheme@gmail.com](mailto:thomas.to.bcheme@gmail.com), [LinkedIn](#), [AI/ML Portfolio](#)

## PROFESSIONAL SUMMARY

Senior AI/ML Engineer deploying GenAI solutions on Google Cloud Platform (GCP) with Google Vertex AI for pipeline monitoring and orchestration. Expert in the full model lifecycle from fine-tuning to architecting multi-agentic RAG systems. Collaborates with IT teams to integrate AI into enterprise business systems and with stakeholders to identify requirements for AI-driven workflows. See [AI/ML Portfolio](#) for demonstration.

## RELEVANT EXPERIENCE

### Founder | AI/ML Open Source | Oakland, CA | Oct 2023 – Present

- Created and maintained Claude Code Marketplace plugins for the open-source community.
- Deployed end-to-end MLOps pipeline: CI/CD RAG frontend on AWS (Vercel: Next.js, TypeScript) and CT/CD Python ML models (scikit-learn, PyTorch, TensorFlow) on Hugging Face (FastAPI, Docker) with Google Vertex AI monitoring.
- Built multi-agentic orchestration (LangGraph) with tool-use (MCP, LangChain) for RAG-initiated plan-and-execute workflows, optimized through fine-tuning for domain-specific use cases.

### Founding Fullstack Engineer | Canventa Life Sciences | Emeryville, CA | Oct 2023 – Present

- Deployed fullstack DevOps SaaS on GCP via CI/CD, reducing daily calculation time by 87% (-40 min) by converting serial to concurrent workflows. Forecasted production within 3 of actual.
- Trained ML models on Snowflake to select cell isolates from donor characteristics, reducing stakeholder decision-making from hours to minutes with a RAG AI Agent.
- Enriched RAG fine-tuning with Atlassian (Confluence) to enhance GenAI (data-to-text, text-to-image) context, improving learning rate by 80% (Wright's Law: Stanford-B model).
- Fine-tuned Snowflake Arctic-TILT model to extract 5+ years of handwritten documents to cloud infrastructure. Architected ETL/ELT pipelines with Python, Google Apps Script, and dbt to load to Snowflake.
- Reverse engineered data pipelines using Tableau, Snowflake, dbt, Fivetran, GCP, SAP, and SQL for gap analysis. Deployed stop-gap pipelines and scoped long-term cloud architecture improvements.

### Software Engineer | Genentech | South San Francisco, CA | Jun 2022 – Dec 2022

- Engineered full-stack applications using Python REST API, Vue.js, and Node.js with git and GitHub, streamlining workflows for medical writing teams.
- Designed knowledge management platforms using HTML, CSS, JavaScript to incorporate UI/UX feedback.

### Process Engineer | Genentech | Vacaville, CA | Jun 2021 – Jun 2022

- Developed automated data consolidation algorithms using Python, R, and SQL, reducing processing time by over 99%.
- Deployed data-driven process monitoring tools on Google Cloud Platform (GCP) to track KPIs.

## SUPPLEMENTAL EXPERIENCE

### Research Engineer | Dr. Alex Dunn | Stanford, CA | Dec 2025 – Present

- Computational de novo protein design targeting malaria.

### Founder | Proprietary FinTech | Oakland, CA | Jun 2022 – Present

- Optimized Python algorithmic and agentic trading frequency to Coinbase's allowable public REST API limit.

### Laboratory Technician | Canventa Life Sciences | Emeryville, CA | Jan 2023 – Oct 2023

- Translated bioprocessing expertise into data systems architecture, mapping data models to physical reality.
- Modeled empirical data using Python (numpy, pandas) to scale validated datasets over 300k datapoints.

### Research Engineer | UC Davis (Nandi, McDonald, Wan, Siegel Labs) | Davis, CA | Sep 2019 – Jun 2021

- Optimized model using Python (numpy, pandas) and numerical methods, reducing costs by \$63.2 Million.
- Quantified organoid growth using Python (OpenCV, ImageJ, Fiji) for image processing analysis.
- Published novel variants using pyRosetta, pyMol, and Benchling with wet-lab biomanufacturing techniques.

## EDUCATION

### Bachelor of Science, Biochemical Engineering University of California, Davis (Graduated) Davis, CA

Advanced Machine Learning with Agentic AI (on going)

AWS Cloud Technical Essentials (completed)