

TIMOTHY PAN

+1-647-532-9818 tjmpn@uwaterloo.ca
[linkedin.com/in/timothyyppan](https://www.linkedin.com/in/timothyyppan) github.com/timothyyppan

Skills

Programming Languages: Python, C, C++, Git, MATLAB, Bash, SQL, SystemVerilog
Python Libraries: PyTorch, TensorFlow, Pandas, Scikit-learn, YOLO, Hyperopt, OpenCV
RTL Tools: SystemVerilog, VHDL, AMD Vivado, Intel Quartus Prime, Verilator
Agentic Development Tools: Claude Code, Codex, Langchain, MCP Server
Language Proficiency: English (Fluent), French (Intermediate, DELF B1), Mandarin/Cantonese (Intermediate)

Education

University of Waterloo **Sep 2023 – 2028**
Bachelor of Applied Science in Computer Engineering - 3rd Year *Waterloo, Canada*

Experience

Apple **Jan 2025 – Aug 2025**
Software Engineering Intern | Packaging Team *Shanghai, China*

- Modernized Apple's legacy packaging test specifications by building a data pipeline that processes **1TB+** of real world shipment telemetry across various hazard events, with resulting data-based specifications projected to reduce redundant testing cycles and over-engineering at company-wide scale across future product lines.
- Trained an LSTM model to classify **30M+** shipment events by acceleration patterns, achieving **97%** accuracy in detecting drop events, then used resulting frequency data to show engineers which failure modes are statistically negligible vs. high priority, directly informing design trade offs during product builds that reduce over-engineering.
- Built **30+** vibration profiles using K-Means clustering and vibrational analysis on field telemetry, replacing unverified legacy specs with profiles grounded in real shipment conditions.
- Designed a metadata extraction and timeline alignment system that parses shipment logs across **3+** couriers and correlates events to environmental conditions, surfacing root causes of in-transit package damage.

University of Waterloo **May 2026 – Aug 2026**
Research Assistant Intern | Pearl Sullivan Engineering IDEAs Clinic *Waterloo, Canada*

- Designed and fabricated a bipedal robot using a Raspberry Pi 5, SolidWorks, and a 3D-printed chassis as well as trained a reinforcement learning policy in MuJoCo that transfers to hardware, enabling robot to walk using **3** distinct gaits and react to **3+** environmental stimuli types.
- Engineered a real-time audio effects processor on a Terasic DE10-Lite FPGA using KiCad, SystemVerilog, applying **5** distinct DSP effects to live mic input with **<5ms** latency, output via 3.5mm jack.
- Built and deployed a Linux remote-compile server supporting **50+** concurrent users to build Quartus Prime projects without local install, cutting per-user setup time by **2+** hours and eliminating 30 GB local disk/download requirement, currently being adopted by **3+** cohorts.

Projects

Product Defect Classifier | *Python, FastAPI, spaCy, SQLite*

- Built an NLP pipeline that embeds manufacturing defect descriptions using sentence transformers, classifies them into categories using cosine similarity, and flags low confidence predictions for human review, replacing manual defect triage.
- Implemented an active learning loop that recomputes category centroids from engineer corrections and clusters unrecognized descriptions using HDBSCAN to surface new taxonomy candidates.
- Built a FastAPI backend with CLI and REST API interfaces for batch defect classification, backed by a spaCy extraction pipeline that parses defect location, intensity, and scope from raw text.

ASIC Math Accelerator Unit | *SystemVerilog*

- Designed and taped-out a Math Accelerator Unit in SystemVerilog using TinyTapeout, supporting scalar, vector, matrix, and polynomial arithmetic on up to **four 8-bit** operands via parallel SIMD hardware multipliers, achieving **200%+** speedup over traditional general purpose CPU datapaths.
- Built a clock-synched, Spie-based wave I/O protocol to reliably transfer data under strict hardware pin constraints.

League of Legends Win Rate Predictor | *Python*

- Developed a real-time ML model in Python that predicts the outcome of a League of Legends match, leveraging Riot Games Developer API to retrieve personal game data.
- Processed over **100** personal game record queries, trained and tested a personalized logical regression model with **10+** features that achieved an accuracy of over **80%**.