

# Peterson Guo

647-606-9486 | [petersonguo@gmail.com](mailto:petersonguo@gmail.com) | [in](#) PetersonGuo | [G](#) PetersonGuo

## EDUCATION

### University of Waterloo

*Bachelors of Applied Science in Honours Electrical Engineering Co-op*

Waterloo, ON

## SKILLS

**Languages:** C/C++, Python, JS, Java, Bash, SQL

**Software:** Git, PyTorch, Tensorflow, CUDA, ROCm, NumPy, Pandas, Spark, Docker, AWS, GCP

**Others:** ASICs, Drivers, Operating Systems, Linux, Windows, CNNs, LSTMs, Neural Networks, Kernel Debugging, FPGA

## EXPERIENCE

### Software Engineer Intern

09/2024 – 12/2024

AMD

Markham, Ontario

- Led the development of a lightweight **ML-based** upscaling PoC using **PyTorch**, **ROCm**, and **Python**, integrating temporal and spatial optimization techniques to enhance display pipeline efficiency
- Developed kernel drivers in **C** and **C++** for next-gen AMD graphics units, improving **hardware** compatibility and performance between the GPU/APU frame buffer and display
- Resolved **25+** kernel-level GPU issues, leveraging WinDbg, crash dumps, hardware register analysis, ETL traces, and firmware traces to debug OS crashes, BSODs, system hangs, and GPU stability issues, enhancing system responsiveness and stability
- Collaborated with **Microsoft** to optimize OS-GPU interactions, analyzing **firmware** and **memory** dumps, resolving initialization failures and performance bottlenecks to ensure seamless driver integration
- Contributed to the Navi4x GPU and Ryzen AI APU launch, resolving high-priority driver issues for AMD display specific features, ensuring **stability** and **performance** for production readiness, ultimately delivering the most stable software release for a newly released AMD GPU
- Contributed to **Linux's** open-source AMD display driver, fixing visual corruption issues affecting color calibration and frame synchronization

### Security Developer Co-op

01/2024 – 5/2024

eSentire

Remote

- Significantly boosted engagement for 3+ clients by pioneering an AI threat analytics dashboard in **Vue3** and **Python** using data visualization and automated reporting, becoming a key project and driving adoption of advanced analytics
- Optimized JSON parsing speed by over **50%** to automate JSON-to-database conversion, by implementing efficient **data structures** and utilizing Python specific performance optimization techniques
- Automated CI/CD pipelines with Terraform, AWS, and GitHub Actions, cutting deployment time by 30%
- Dramatically enhanced data processing speed of **Python** logging functions by over **400%** using task-specific algorithms
- Refined the logging, infrastructure, and security protocols for multiple AI projects by implementing **3** standardized methods in **Rest API's** to ensure robust access control and system monitoring, preventing vulnerabilities such as prompt injection, model theft, XSS attacks and SQL injection
- Improved an open-source PCAP scrubber using **Python** and **PyQt**, adding over **10** functionalities such as a GUI, multi-threading, text editing, automatic checksum validation, autosave, and find and replace to enhance user experience

### Software Developer

06/2023 – 09/2023

COBWEB, University of Toronto

Toronto, ON

- Developed four physics-based simulation models in **C++** & **Python**, improving simulation speed by 20%
- Optimized genetic algorithms by improving **multi-threading** efficiency and **memory management**, reducing execution time by 10% and memory footprint by 15%

## PROJECTS

### ML Upscaling | ROCm, CUDA, Machine Learning, CNNs, PyTorch, Python

- Designed an ML upscaling model for **AMD graphics cards**, leveraging **transformer** based **super sampling** to enhance visual fidelity and performance

### InvestIQ | Python, LSTMs, IBKR API, CUDA

- Built an automated trading alert system, monitoring market indicators (RSI, MACD, Stochastic Oscillator)
- Sanitized training data and utilized **LSTMs** to use upcoming events and indicators to predict future volatility

### Bionic Evo | C/C++, Assembly, Neural Networks, TensorFlow, CUDA, STM32

- Engineered a humanoid arm prototype for amputees by utilizing **STM32** and **EMG** sensors, integrating **pattern recognition** to achieve **precise gesture classification** and seamless arm control

### Assisted Reader | ESP32, Python, OCR, Machine Learning, React, Tesseract API, Autocorrect, JS, Python

- Launched a user-friendly OCR platform that assisted users with vision disabilities in seamlessly accessing printed material; integrated text-to-speech functionality, resulting in a significant decrease in learning barriers
- Leveraged Tesseract, ESP32 camera, and **autocorrect technologies** to improve accuracy and provide real-time feedback for users with impaired vision