Kyle **Wade**

+1 (949) 887-7545 | kwade@ucsd.edu | kylewade.dev | kyle1373 | kylewade1373

Education

University of California, San Diego — *San Diego, CA*
M.S. in Computer Engineering (Machine Learning & Data Science) — *Sept 2023 - Mar 2025*
B.S. in Computer Engineering — *Sept 2020 - June 2023*

- **3.8 GPA** | Member of Honor Societies **Tau Beta Pi (TBP)** and **IEEE - Eta Kappa Nu (HKN)**
- Relevant Coursework: Advanced Data Structures and Algorithms, Object-Oriented Programming, Machine Learning, Operating Systems, Recommender Systems, GPU Programming, Deep Learning, Data Science, Robotics, Networking

Experience

Tesla — *Palo Alto, CA*
Software Engineer Intern — *Jan 2024 - April 2024*

- Fixed critical issues in Tesla's service engineering software by improving server APIs and **SQL** database queries which saved **2-5 seconds** per page request.
- Used **Python**, **C++**, **Typescript**, and **React.js** to create a **Linux** terminal allowing Service Engineers to safely execute vehicle commands over a websocket.
- Migrated 1 large, centralized service into 2 distributed microservices using **Docker** controlled by **Kubernetes**, which increased Tesla's backend efficiency.
- Built a **Golang** backend which syncs data from 10+ data sources into one centralized AWS S3 bucket to improve request time across multiple Tesla APIs.
- Wrote a new **Continuous Integration (CI)** and **Continuous Deployment (CD)** pipeline with **unit tests** and **integration tests** to ensure code robustness.

UC San Diego — *San Diego, CA*
Machine Learning Researcher @ Mobile Systems Design Lab — *Jan 2023 - Jan 2024*

- Created a mobile app using **Swift** and **React Native** in which patients record their physical therapy exercises at home and get ML-based real time feedback.
- Built a **Python** training server which takes in an exercise skeleton via an API endpoint and trains a new model which is used for inferencing on the phone.
- Conducted user interviews and deployed the platform to **2 physical therapy clinics** and **50+ patients**, with more trials still ongoing.

Instructional Assistant — *Sept 2021 - Mar 2023*

- Helped **1,400+ students** by hosting lectures, grading exams, giving constructive feedback with programming assignments, and holding discussion sessions in a data structures and algorithms class, teaching **binary search trees**, **linked lists**, **stacks**, **heaps**, **priority queues**, **hashmaps**, and more.
- Received a **99% student approval** rating according to UC San Diego's feedback system.

Qualcomm — *San Diego, CA*
Software Engineer Intern — *June 2023 - Sept 2023*

- Optimized the processing of IP log packets in the 5G data layer for Qualcomm's ARM-based chipsets by improving data throughput algorithms in **C++**.
- Improved CPU multithreading capabilities by establishing new locks and semaphores which handled race conditions and prevented new runtime crashes.
- Used **AWS SageMaker** and **AWS QuickSight** to transition the team in choosing chipset memory thresholds from trial-and-error to data-driven analysis.

Software Engineer Intern — *June 2022 - Sept 2022*

- Worked with **8 customers** to develop new features for the Qualcomm Snapdragon computer chip using **C++** with benchmarking tools in **C#** and **XAML**.
- Created new thermal stress testing software which tested the heat capabilities of Qualcomm's ARM-based chipsets and calculated optimal thermal zones.
- Achieved **3rd / 153 competitors** in Qualcomm's annual 5G hackathon through creating Forestshield, an early wildfire detection tool for first responders.

Navlabs — *Irvine, CA*
Software Engineer Intern — *June 2021 - Sept 2021*

- Developed device drivers for GPS receivers using **embedded C** which connects hardware sensors and microcontrollers to the operating system.
- Created a **Python** application which parses real time binary data outputted from GPS simulators and converts it into a readable format shown in a program.

Projects

Archiverse (600,000+ visits / month)
17TB Miiverse Archive — Website / GitHub

- Created an archive of Miiverse, Nintendo's discontinued social platform which had 8M+ users, 133M+ posts, and 216M+ replies, totaling 17TB of data.
- Built the frontend using **React.js** and constructed the backend using **Express.js** and **SQL** with optimized binary tree indexes for millisecond-level searching.
- Cached server responses into the web browser using **Redux**, which resulted in **90% fewer API calls** due to the data not needing to be fetched again.
- Acquired by **Pretendo Network**, a company which provides open source replacements for Nintendo's discontinued Wii U and 3DS online services.

osu! capital (11,000+ users)
Quantitative osu! Stock Market — Website / GitHub

- Created a stock market platform in **Next.js** that models osu! player performance statistics as stock prices which users can invest paper currency into.
- Utilized **Golang** for the REST API, **PostgreSQL** for the database, **Redis** for storing user sessions, **Stripe** for payment processing, **Docker** for running microservices, and **Kubernetes** for container management, lowering operation costs from **$300/mo** to **$15/mo** through migrating towards self-hosting.

Where2Be (2,000+ users)
University Events Mobile App — Website / GitHub

- Developed a **React Native** mobile app which scrapes data across Instagram, Facebook, and Discord and parses them into events for students to join.
- Built a **Fast API** backend written in **Python** integrated with a **Neo4j** database to create an event recommendation system based on student interests.
- Marketed the app to students across UC San Diego, UC Berkeley, USC, and the University of Illinois Urbana-Champaign, resulting in **2,000+ users**.

Stock Portfolio Optimization
Machine Learning Models for Predictive Investment Strategies — Paper

- Implemented various machine learning models (RF, LSTM, DMLP) to test stock price prediction on over 10 years of market data from Yahoo Finance.
- Outperformed the S&P 500 by **4.1x** using a Random Forest-based model for stock price prediction, combined with Mean-Variance Optimization (MVO) to dynamically adjust portfolio weights for maximizing returns and managing risk, implemented using **SciPy**, **PyTorch**, **pandas**, **NumPy**, and more.

Technical Skills

Languages: Python, C, C++, JavaScript, TypeScript, Golang, SQL, C#, XAML, Java, Swift, HTML, CSS
Technologies: Docker, Kubernetes, React.js, React Native, Next.js, Fast API, Node.js, Redux, Expo, Neo4j, OpenCV, TensorFlow