[redacted]

+1 [redacted] • [redacted]@[redacted] • linkedin.com/in/[redacted] • gitlab.com/[redacted]

Education

University of Wisconsin–Madison

Sep. 2023 – May 2025

B.S. in Computer Sciences, Mathematics, Physics

- CS 252, 300; Math 521, 541, 551; Physics 311
- 4.0 GPA

Awards

USACO	USAMO	USAPhO
Platinum Division	**Qualification**	**Qualification**

Research

[redacted]

Python, Qiskit, Tensorflow Quantum, Typst

- Coauthor of paper under peer review for the International Journal of Quantum Information.
- Developed a comprehensive theoretical framework to apply quantum annealing principles to hypergraph partitioning problems, identifying key variables and quantum states relevant to NP-completeness.
- Formulated a set of algorithms designed to efficiently find near-optimal solutions to hypergraph partitioning.
- Performed a complexity analysis of the proposed algorithms under various computational scenarios.

Projects

[redacted]

Rust, LALRPOP, inkwell, qip, LLVM

- Designed a FP language based on homotopy type theory supporting classical and quantum computing paradigms.
- Authored a comprehensive standard library, including a symbolic math library.
- Developed a compiler that translates high-level language into IR while ensuring type safety.
- Built a simulator for the hybrid IR to test and debug algorithms across computing paradigms.

[redacted]

Python, PyTorch, Numpy, Scipy, OpenAI Gym, Docker

- Developed a Pokémon Showdown AI with an architecture combining MCTS with a deep learning evaluation function.
- Integrated CNN layers for spatial feature extraction and LSTM units for understanding temporal game dynamics.
- Employed a mix of supervised learning with historical data and reinforcement learning via self-play.
- Implemented dropout, batch normalization, and adaptive learning rates for robust and efficient network training.

[redacted]

Astro, SolidJS, Tailwind CSS, TypeScript, Elixir, Apache Kafka, Redis, PostgreSQL, GraphQL, WebRTC, Docker

- Developed a hybrid chat/forum application using Astro, Solid, Tailwind, and TypeScript for a responsive front end.
- Engineered an efficient backend in Elixir, with Apache Kafka for real-time messaging, Redis for in-memory caching, and PostgreSQL for data storage, with support for GraphQL queries.
- Ensured secure P2P communication and data consistency in a decentralized network environment using WebRTC.

Tools

- **Languages:** Python, TypeScript, Java, Kotlin, C++, C, Rust, Go, Haskell, Raku, Elixir, Ocaml, Zig, Clojure, Nim
- **Data:** PostgreSQL, Redis, MongoDB, Datomic, DuckDB, Supabase, Cassandra, Elasticsearch, Kafka, RabbitMQ
- **ML:** PyTorch, Tensorflow, Keras, scikit-learn, Numpy, Theano, Scipy, Pandas, Hugging Face, OpenCV, CUDA
- **Web:** React, Angular, Vue, Svelte, Phoenix, Elm, Astro, Solid
- **DevOps:** Docker, Kubernetes, Ansible, Terraform, Linux (Alpine, Arch, Debian, Nix), Bash, Git, CircleCI