Tailai Ying

385-256-3856 | tty6@cornell.edu | tailaiying32.github.io | linkedin.com/in/tailai-ying-099041260 | github.com/tailaiying32

Education

Cornell University — Ithaca, NY
B.S. in Computer Science — *Expected May 2027*

Technical Skills

Languages: Java, Python, C/C++, Go, SQL, JavaScript, TypeScript, OCaml, HTML/CSS, XML
Systems & Infrastructure: SLURM Workload Manager (HPC), Lambda Cloud, gRPC, Protocol Buffers (Protobuf), Linux
Tools & DevOps: Docker, CI/CD (GitHub Actions), Git, Shell, Postman, Weights & Biases
Web & Backend: Spring Boot, Node.js, Flask, PostgreSQL, Prisma, Next.js, React, React Native, TailwindCSS
AI & ML: PyTorch, Hugging Face, RAG, llama.cpp, scikit-learn, NumPy, Hydra, Optuna

Relevant Experience

Technical Lead — Jan 2025 – May 2025
CommuniCare — *Ithaca, NY*

- Architected and deployed a Flask-based microservices backend, implementing a Geospatial Abstraction Layer to optimize provider discovery and **reduce query latency by 20%.**
- Designed and implemented a robust Role-Based Access Control (RBAC) system to ensure HIPAA-compliant handling of sensitive user metadata across the API layer.
- Built and maintained the GitHub Actions CI/CD pipeline, reducing deployment friction and **ensuring 99.9% uptime** during the initial launch.

Research Assistant — May 2025 – Present
EmPRISE Lab — *Ithaca, NY*

- Owned theoretical and practical implementation for active learning core of personalized caregiving robotics framework, learning user reachability with **14% more accuracy** than current methods; **2nd author** for paper submitted to RSS 2026.
- Engineered distributed ML infrastructure for robotics research, implementing CUDA-accelerated training pipelines and batched processing that **improved model efficiency by 10×**.
- Designed and managed distributed data pipelines on HPC clusters and Lambda Cloud, utilizing SLURM and custom shell scripts to parallelize hyperparameter searches, **reducing training latency by 50×**.

Machine Learning Intern — May 2025 – Aug 2025
Aria Lab — *Salt Lake City, UT*

- Developed a high-concurrency 3D swarm simulation using a modular controllers to support diverse multi-agent configurations.
- Scaled simulation throughput by parallelizing agent updates across HPC nodes, enabling simultaneous evaluation of **100+ swarm configurations** and optimizing CPU/GPU resource utilization.
- Engineered a high-throughput telemetry pipeline using clustering and novelty search algorithms to programmatically classify **10+ emergent agent behaviors** from massive simulation datasets.

Selected Projects

Borglite - Distributed Task Scheduler | *Go, gRPC, Protobuf* — Jan 2026 – Feb 2026

- Designed and implemented a distributed task scheduler capable of managing tasks across multiple worker nodes by engineering a Master-Agent architecture with gRPC and Protobuf for efficient RPC communication.
- Achieved fault tolerance and system resilience by implementing a heartbeat monitoring system that automatically detects node failures **within 5 seconds** and re-schedules affected tasks using a custom retry mechanism.
- Optimized resource utilization across the cluster by developing a custom First-Fit scheduling algorithm that dynamically assigns tasks to agents based on real-time CPU and memory availability.
- Streamlined cluster management by creating a user-friendly CLI tool that allows users to submit jobs, monitor real-time status updates, and inspect agent health with sub-second latency.

HTTP Web Server | *OCaml* — Apr 2025 – May 2025

- Implemented a fully functional HTTP web server from scratch using low-level socket programming, request parsing, and concurrent connection handling to ensure protocol correctness and stability.
- Designed a multithreaded connection model to support simultaneous client requests with low latency and efficient resource utilization.
- Built custom request/response parsers and error-handling logic compliant with HTTP specifications, improving robustness against malformed client input.

Jarvis - Multi-modal Voice Assistant | *C++, llama.cpp, sherpa-onnx* — Dec 2025 – Jan 2026

- Prototyped a high-fidelity audio-first voice assistant using Qwen3, Whisper, and Sherpa-ONNX, optimizing for natural speech flow and low-latency user interaction.
- Engineered an intelligent phrase-boundary detection system to optimize real-time inference, **reducing end-to-end latency to sub-1s** to ensure a seamless consumer experience.
- Orchestrated concurrent model execution and audio pre-buffering to maintain pipeline performance and scalability.

CritterEvo - Artificial Life Simulator | *Java* — Dec 2024 – Feb 2025

- Engineered evolutionary ecosystem simulator in Java, featuring procedural world generation, neural network controllers, and genetic algorithms to simulate emergent behavior and natural selection.
- Optimized performance with multi-threading and lazy loading, increasing simulation **throughput by over 80%**.
- Optimized entity navigation by implementing an A* pathfinding engine with state caching, significantly reducing computational overhead.
- Achieved **95% line coverage** with comprehensive JUnit test suite, ensuring robust functionality across edge cases.

Lockd - Smart Lock System | *Raspberry Pi, React Native, Flask* — Oct 2024

- Developed an IoT smart lock system featuring remote control and vibration/sound anomaly detection with **under 200ms alert latency** from edge device to client, and real-time push and email notification services to ensure immediate user response to security threats.
- Awarded Finalist and Beginner's Prize at BigRedHacks (**135+ hackers**) for technical implementation and system design.

Ear Trainer - Ear Training Web App | *Spring Boot, React, PostgreSQL* — Jul 2024 – Aug 2024

- Architected a RESTful API using Spring Boot and PostgreSQL to support a music education platform to train aural skills.
- Engineered a scalable data model to manage complex music theory metadata, including interval relationships, chord progressions, and rhythmic patterns for interactive ear-training exercises.
- Integrated robust error-handling middleware to ensure stable communication between the React frontend and the data layer.