

# Jingzhi Jiang

Berkeley, CA | (503)431-0379 | jiangjingzhi@berkeley.edu | linkedin.com/in/jingzhi-jiang2003

## EDUCATION

### University of California, Berkeley

Aug. 2022 – Jun. 2026

B.A. in Computer Science

GPA: 4.0

- Coursework: Deep Learning, Efficient Algorithms and Intractable Problems, Discrete Math and Probability, Principle and Technique of Data Science, Data Structures, Machine Structure, Computer Security, Computer Vision

## PAPER

- Aidan W., Norah A., Jingzhi J., Andrew J., Karen S., Andy Z., Basel A., David W. SeedAlchemy: LLM-Driven Seed Corpus Generation for Fuzzing. *2025 Workshop on the use of Large Language Models for Cybersecurity*. accepted

## RESEARCH

### Dawn Song Research Group | Computer Security Researcher | Berkeley, CA

Oct. 2025 – Present

- Developed e2e-cyber-bench: a large-scale, high-quality cybersecurity evaluation framework to rigorously assess the capabilities of AI agents on real-world vulnerability analysis tasks.
- Integrated 1,507 benchmark instances with historical vulnerabilities from 188 large software projects.
- Designed a testing pipeline to evaluate AI agent's ability to patch code vulnerabilities using fuzzing and unit testing.

### David Wagner Research Group | Computer Security Researcher | Berkeley, CA

Jan. 2025 – Present

- Automated fuzzing processes for GitHub projects with LLMs to identify vulnerabilities more thoroughly.
- Improved fuzzing code coverage for projects by integrating OSS-Fuzz and other fuzzing tools like LibFuzzer.
- Built a web scraper using Scrapy and Search API to search for specified files with diverse file features
- Constructed a Docker container to automate seed corpus minimization and fuzz benchmark testing, and extended the Magma benchmark to support multi-corpus testing in parallel.
- Co-authored a paper accepted to the LLM4Sec Workshop at ICDM 2025.
- Used LLM to generate log rules to discover malicious activity from host system logs to detect attack in real time.

### Eric and Wendy Schmidt Center for Data Science & Environment | Researcher | Berkeley, CA

Apr. 2024 – Present

- Determined the optimal strategy for green crab catching with deep reinforcement learning techniques like RecurrentPPO.
- Finetuned deep reinforcement learning model and constructed hyperparameter optimization scripts using Optuna.
- Built a simulation to showcase changes in green crab population over time using Gymnasium.
- Constructed different RL training environments with varying observations (time-aware or catch effort) and randomness to enhance model robustness.

## INDUSTRY EXPERIENCE

### KPMG | Software Development Intern | Shanghai, China

Jul. 2024 – Aug. 2024

- Implemented prompt engineering algorithms auto COT to improve ERNIE LLM's response accuracy by around 10%.
- Developed PDFtoText prototype with added feature for image extraction. Improved text extraction accuracy.
- Completed graphRAG implementation using Qwen LLM as the base model, Microsoft graphRAG and LM Studio.
- Evaluated TextToSQL models performance with KPMG data using HuggingFace and LlamaIndex.

### Floras and Berkeley Skydeck | Full Stack Software Development Intern | Berkeley, CA

Jan. 2024 – May 2024

- Designed a website for Floras showcasing 200 sustainable projects with searching and filtering features.
- Implemented the storage backend with Python and MongoDB for fast metadata queries from the Patch database.
- Developed front-end website integration using Flask, HTML, JavaScript, and CSS.

## TEACHING EXPERIENCE

### UC Berkeley CSM | CS61C mentor | Berkeley, CA

Aug 2024 – Dec 2024

- Taught a small group of students about CS61C contents including RISC-V, C, CPU structure, Cache, etc.

### UC Berkeley EECS Department | CS61B Academic Intern | Berkeley, CA

Apr 2023– Aug 2023

- Supported weekly sections of 100+ students to help reinforce core data structure concepts

## PROJECT

### WebBot | Python/Javascript

Dec. 2024 – Feb. 2025

- Developed a Chrome extension with Azure LLM to summarize website content and answer questions using embeddings and RAG.
- Enabled the extension to automatically detect and solve CAPTCHAs using Azure computer vision and LLM.
- Incorporated Flask, Azure API, React, Javascript, and HTML into the extension, eventually deploying in Azure.

## SKILLS

- **Technical:** Python(numpy, pandas), Java, C, SQL/MySQL, Go, RISC-V, HTML, CSS, Javascript, React.js, Docker, Firebase, MongoDB, HuggingFace, LlamaIndex, Langchain, Pytorch, Scikit-learn, Azure, Linux