

Shangdian (King) Han

✉ kingh0730@gmail.com · 🏠 kinghan.info · 🌐 kingh0730 · 📱 kingh0730



Personal Statement

- Full-stack research engineer.
- Proficient in software engineering, with 5+ years of experience in successfully collaborating with ML researchers to turn research into products.
- At Microsoft Research, I prototyped the new Office AI for Excel.



Education

University of California, Berkeley

Berkeley, CA

B.A. Honors Computer Science & Mathematics

May 2024

- Tech GPA: 4.0 | GPA: 3.92



Experience

UC Berkeley Electrical Engineering & Computer Sciences

Jan 2023 - Dec 2023

Researcher

- Led a team of 5 researchers to develop an ML service for circuit design on GCP and a reinforcement-learning library for circuit optimization using OpenAI Gym and Ray.
- Enabled ML researchers to design optimal circuits without any prior knowledge of circuit design, reducing the design time from 1 week to 1 day.

Huawei

Sep 2022 - Dec 2022

Software Engineer Intern

- Designed, implemented, and validated a new service for frequency management using C/C++, enabling cellular networks to support the new 5G standard.
- Optimized a deployed service using C/C++, reducing its runtime complexity from exponential to polynomial.

Tsinghua University

May 2022 - Aug 2022

Research Engineer

- Built a service for data processing and analysis using Django, enabling data scientists on the team to perform 10+ data pipelines, including association rule learning, dynamic time warping, etc.
- Used PyTorch CNNs to recognize objects in NASA satellite images, validating the feasibility of a new product.

Microsoft Research

Sep 2021 – May 2022

Research Fellow

- Prototyped the new Office AI for Excel using TypeScript, C#, and TensorFlow, enabling users to perform 10+ AI tasks. Shipped to the Excel product team.
- Improved a deployed ML classification model (LSTM + CRF) using TensorFlow, increasing its F1 score from 72 to 77.



Skills

- **Expert:** Python, PyTorch, JS/TS, React/Next.js
- **Proficient:** C/C++

- **Proficient:** C/C++