

Jinxuan Liang

(510) 599-0981 | degaliang@gmail.com | Berkeley, CA

EDUCATION

University of California, Berkeley

Aug 2021 - May 2024

- **Major:** Computer Science, B.A. (GPA: 3.92); Electrical Engineering & Computer Science, M.S. (GPA: 4.00)
- **Honors:** Dean's Honor List, Distinction in General Scholarship
- **Programming Coursework:** Algorithms & Data Structures, Software Engineering, Architecture, Security
- **AI/ML Coursework:** Statistical Learning, Deep Learning, Computer Vision & Graphics, Parallel Computing
- **Master's Thesis:** [Skeleton-based Fall Detection using STGCN with Learnable Edges](#)

SKILLS AND TOOLS

- **Coding:** (*proficient*): Python, C, Java (*familiar*): C++, SQL, Ruby, JavaScript, Matlab, CUDA, Git
- **Frameworks:** (*proficient*): PyTorch, OpenMP (*familiar*): MPI, UPC++, Rails, Next.js

WORK EXPERIENCE

Student Researcher

UC Berkeley Barsky Lab

Aug 2022 - May 2024

Real-time Fall Detection System, Project Initiator and Team Leader

- Obtained 96% accuracy in vision-based fall detection by utilizing AlphaPose for feature extraction and Spatial Temporal Graph Convolutional Networks (STGCN) as the backbone model for prediction.
- Improved the model prediction accuracy by 15% through integrating learnable edge weights into STGCN and implementing training-time batch-balanced sampling.
- Established a more accurate evaluation metric for fall detection through the analysis of 20GB video data
- Leveraged knowledge in Deep Learning and Computer Vision, programmed in Python, debugged the neural networks using Weights & Biases

Head-Tracking PC Controller for Quadriplegic Users

- Implemented voice control using Google Speech for speech recognition and PyAutoGUI for automation.
- Developed the dictation module by utilizing the built-in copy-paste function of the OS for text insertion.
- Leveraged knowledge in Git and signal processing, programmed in Python

MLE Intern(Unpaid)

Oracle(China) Software Systems Co Ltd

Jun - Aug 2023

- Implemented a predictive maintenance model for industrial blending machines using Multivariate State Estimation Technique (MSET).
- Developed a user interface with Plotly to visualize machine data and model outputs for customers.
- Leveraged knowledge in linear algebra and statistical learning, programmed in Python.

PROJECT EXPERIENCE

Personal Website: jliang.me (for more details), implemented using Next.js and deployed on Vercel

Vision-Language Model (BLIP) for Pose Estimation

- Achieved 90% accuracy in pose estimation using VLM by fine-tuning it on a customized captioning dataset.
- Increased accuracy by 5% and improved the model's robustness to occlusions through the analysis of decoder's self-attention layers and the rearrangement of token generation order.
- Utilized: Python, PyTorch, Scikit-learn, Numpy, Kaggle

Sliding Puzzle Solving with Transformers: Vision-Language Model and Masked Autoencoder

- Created a 2000-image puzzle dataset by sampling and shuffling image patches derived from ImageNet.
- Obtained 78% of test-time success rate using 1600 images for fine-tuning pre-trained VLM and MAE.
- Spearheaded research comparing multimodal and vision models in the puzzle-solving task, showing VLM's strength in object puzzles (cars, bikes) and MAE's superiority in texture puzzles (grassland, desert).
- Utilized: Python, PyTorch, Scikit-learn, Numpy, Kaggle

3D Reconstruction: Rebuild Google Street View scenes as 3D Point Clouds

- Reconstructed individual street view scenes as local point clouds by collecting depth and RGB data from Google Street View API, and transforming image pixels to points in 3D Cartesian coordinate system.
- Aligned local point clouds in 3D space for reconstruction of a large scene through SIFT and FPFH feature-matching in pixel space and geographical coordinate matching in the geodetic coordinate system.
- Rendered the reconstruction of Time Square using Potree and displayed the online demo through AWS.
- Utilized: Python, OpenCV, Numpy, Scikit-image, Matplotlib, Open3D, Potree

District Map: US Congressional Districts Lookup Site

- Developed a SaaS App that displays representative information by district using Rails.
- Implemented RESTful APIs for users to search representatives by name or address in the database.
- Designed and implemented Cucumber test cases for Behavior-Driven Agile Development.
- Utilized: Rails, Behavior-Driven Agile Development, Cucumber, Heroku, Codecov

Parallelized N-body Simulation

- Developed an $O(n)$ algorithm for simulating particle interaction by utilizing a distributed data structure to compute particle forces in parallel. Minimized runtime communications among computing units.
- Utilized: C programming, OpenMP, MPI, CUDA, CUDA Profiling Tools Interface (CUPTI), Matplotlib.