

# Thomas Lu

☎ (650) 398-9421 | ✉ [thomaslu@berkeley.edu](mailto:thomaslu@berkeley.edu) | [tsl628.github.io](https://github.com/tsl628)

## EDUCATION

### University of California, Berkeley

*Bachelor of Arts in Computer Science and Physics (GPA 4.0/4.0)*

Aug. 2022 – Present

*Expected May 2026*

## RELEVANT COURSEWORK ([FULL LIST](#))

- Data Structures
- Advanced Linear Algebra
- Optimization Models in Engineering
- Probability and Random Processes
- Computer Architecture
- Artificial Intelligence
- Advanced Algorithms
- Machine Learning
- Natural Language Processing

## EXPERIENCE

### Speech and Computation Lab at UC Berkeley

*Undergraduate Research Fellow*

August 2022 – Present

*Berkeley, CA*

- Developed deep learning models of human speech using Generative Adversarial Networks (GANs), incorporating articulatory models to create realistic models that parallel human cognitive function (refer to publications).
- Implemented spoken language models in PyTorch and developed architectural and performance improvements.

### Rivian Automotive, Inc.

*Camera Systems Engineering Intern*

May 2024 – August 2024

*Palo Alto, CA*

- Developed an in-house automated tool for computing image quality metrics using Python and OpenCV, matching leading commercial tools (i.e. Imatest, DXOMARK) while greatly increasing image throughput.
- Contributed to ISP tuning for in-house camera drivers on the NVIDIA Jetson Orin platform, delivering improvements on metrics such as sharpness, white balance, noise, and LED flicker.

### Lawrence Berkeley National Laboratory

*Undergraduate Researcher*

August 2023 – July 2024

*Berkeley, CA*

- Designed a pipeline to apply redshift corrections to DESI quasar observations, reducing dispersion by 30%.
- Incorporated multiprocessing using JobLib and the NERSC High-Performance Computing cluster to visualize and analyze datasets as large as 200GB, delivering up to a 70x speedup on certain analysis tasks.

### Multi Physics and Circuit Lab at San Jose State University

*Student Researcher*

May 2021 – December 2022

*San Jose, CA*

- Designed deep learning tools for developing integrated circuits, extracting hidden device properties from empirically measured data and enabling inverse design based on desired device characteristics.
- Designed pipelines in Tensorflow with VAEs and shallow regression to approximate highly nonlinear device properties with  $R^2 = 0.98$  accuracy. (First-author publications in IEEE T-ED and IEEE SISPAD)

## SELECTED PROJECTS AND PUBLICATIONS ([FULL LIST](#))

### Unsupervised Articulatory Learning in GANs | *IEEE TALSP (pending revision)*

- Using a pretrained ema2wav model, demonstrated that GANs could learn to generate speech by moving vocal articulators with only indirect feedback, paving the way for more realistic models of human language acquisition.
- Greatly reduced noise in the model by adjusting kernel sizes and adding a low-pass convolutional filter, producing clearer and more interpretable results.
- Adapted the model to use the AWS Trainium accelerator for a 4x speedup compared to original GPU solution.

### Syntax from Speech in GANs | *Journal of Memory and Language (under review)*

- Discovered that GANs trained on single-word inputs would spontaneously generate multi-word outputs at extreme latent value inputs, and quantified this behavior using a logistic regression model with  $p < 0.0001$ .
- Implemented a fine-tuned Whisper model to automatically transcribe data, allowing for 100x increase in total data analyzed. Produced a 30% accuracy increase and 4x cost reduction compared to the off-the-shelf OpenAI solution.

### Sentiment Analysis with QRNN | *UC Berkeley Deep Tech Conference 2023 (oral presentation)*

- Developed a Quantum Recurrent Neural Network (QRNN) for use in sentiment analysis tasks and implemented a classical small-scale simulation with 5 input qubits using PyTorch as a proof-of-concept.
- Achieved a 15% accuracy increase compared to a naive Variational Quantum Eigensolver implemented in Qiskit, and a 20% accuracy increase compared to a classical RNN implemented in PyTorch with equal parameter count.

## TECHNICAL SKILLS

**Languages:** Java, Python, C, C++, R, RISC-V  
**Tools:** Git, AWS EC2, GDB, VSCode, Slurm, Vim

**Frameworks:** Tensorflow, PyTorch, QisKit, OpenMP, OpenMPI  
**Hardware:** KiCad EDA, Fusion360, 3D printing