Joey **Zhu**

📱 (+1) 925-858-2048 | ✉ joey.j.zhu@berkeley.edu | 🏠 https://joey-j-zhu.github.io/ | joey.j.zhu | joey-j-zhu

Education

University of California, Berkeley *Berkeley, CA*
B.A. Computer Science *Aug. 2019 - Exp. Dec. 2022*

CS (3.5 GPA)	Data Structures, Algorithms, Machine Learning, Random Processes, Computer Architecture, Information Systems
Physics (3.7 GPA)	Analytic Mechanics, Perturbation Theory, Statistical Physics, Quantum Mechanics, Electrodynamics

Skills

Programming	Python, C, RISC-V, Java, JavaScript, React, Node.js, HTML, CSS
Tools and Frameworks	Git, Numpy, Scipy, Tensorflow, Tkinter, OpenMP, Photoshop, Illustrator
Soft Skills	Planning, Organization, Problem Solving, Graphic Design

Work Experience

UC Berkeley EECS Dept. *Berkeley, CA*
Academic Intern *Jun. 2020 - Oct. 2020*

- Worked with instructors in Data Structures (CS 61BL) and Discrete Mathematics (CS 70) courses to debug staff auto-graders and student labs
- Led sections of two groups of 25 students each, created practice material to help build intuition for data structures and mathematical proofs

UC Berkeley IEEE *Berkeley, CA*
Marketing Committee, Graphic Designer *Aug. 2020 - Jul. 2021*

- Created main designs for yearly apparel order
- Led apparel design project with two other designers and coordinated distribution logistics with executive members

Amazon *Seattle, WA*
SDE Intern (Incoming) *Jan. 2022 - May 2022*

Projects

NP Set Partition *Berkeley, CA*
Divide a set of items into several groups to maximize a total score, maintaining a cost threshold *Dec. 2020*

- Led implementation of the main solution structure, combining stochastic gradient descent and genetic algorithms
- Improved quadratic-time validity computations to linear-time, and linear-time sampling to amortized constant-time
- Generated pathological test inputs to test the resilience of solver

NumC *Berkeley, CA*
Accelerate native C/Python array-based routines, with focus on L1 cache and instruction set optimization *Jul. 2021*

- Used a combination of Intel x86 intrinsic vectors, OpenMP threading, cache optimization, and minimizing instruction set sizes
- Added transposition to matrix multiplication routine for more cache hits on successive dot products, speedup from factor of 5 to 80
- Minimized memory allocations, accesses, and compulsory cache misses, pushing matrix powering speedup from 780 to 820

Perlin Mapping *Home*
Tune Perlin fractal noise parameters to visually approximate and manipulate images *Jun. 2021*

- Lowered squared raster error from 300k to less than 1k for images with varying features, via gradient descent and error-passing for successive octaves
- Adjusting program and exploring more parameters for gradient descent process to behave more predictably and produce numerically similar parameters for visually similar images or successive frames