

Zane Alumbaugh  
San Francisco Bay Area  
(510)599-6558  
zanedma@gmail.com

---

## EDUCATION

---

### University of California, Santa Cruz

September 2017-Present

*Santa Cruz, California*

- Bachelor of Science in Computer Science (expected Spring 2021)
- Achievements: UCSC Dean's Honors List – Spring 2020, Winter 2020, Winter 2018

---

## COMPUTER SKILLS

---

### Languages

- Proficient in: Python, C/C++, Golang, HTML/CSS, Java
- Familiar with: JavaScript, Scala, Prolog, Haskell, SQL, Postgres

### Software

- APIs/Libraries/Frameworks: PyTorch, Django, Bootstrap
- Software: Windows, Unix, Git, Docker

---

## EXPERIENCE

---

### Student Research Assistant/Python Programmer

June 2020-Present

*University of Washington*

- Implement action policies for Reinforcement Learning Linear Quadratic Systems and extend their application to a multiplayer environment.
- Update policy gradient algorithm to multiplayer environment using OpenAI's existing Vanilla Policy Gradient and PyTorch
- Author credit in: Liyuan Zheng, Tanner Fiez, Zane Alumbaugh, Benjamin Chasnov and Lillian Ratliff (2020), Stackelberg Actor-Critic: A Game-Theoretic Perspective

### Student Research Assistant/Python Programmer

July 2019-September 2019

*University of Washington*

- Implemented gradient-based learning algorithms for training Generative Adversarial Networks (GANs) using the PyTorch Framework.
- Implemented an optimizer class for PyTorch that used second-order derivatives to predict future optimizations and allowed for the learning agent to act on these predictions.
- Built foundations for a novel approach to accelerate the GAN learning process by leveraging the eigen structure of the Jacobian of the learning dynamics.

### Website Developer

February 2019

*Bay Area Geophysical Society (BAGs)*

- Created current BAGs website from scratch. Employer requested a barebones static implementation. Used HTML, CSS, and minimal JavaScript.

### Freelance Software Engineer

July 2020

*BuddhiBox*

- Created a bot that scrubs daily emails from "Help a Reporter" (HARO) and picks out links containing relevant keywords and phrases. The bot then notifies the user via email with only the relevant links. Implemented using Python, BeautifulSoup, and Google's official Gmail API.

References available upon request.