

# Arth Shukla

Redwood City, CA | <https://arth.website> | <https://www.linkedin.com/in/arth-shukla/> | (650) 850-9097 | arthshukla03@gmail.com

## EDUCATION

---

### University of California, San Diego

September 2021 – June 2025

Bachelor of Science in Mathematics-Computer Science

GPA – 4.0

#### Relevant Coursework

*Completed:* Deep Learning for 3D Data (Graduate Level), Rec Systems and Web Mining, Supervised Machine Learning, Optimization Methods for Machine Learning I and II, Design and Analysis of Algorithms, Data Science in Practice, Theory of Computation, Advanced Data Structures, Computer Organization and Systems Programming, Data Structures and Object-Oriented Design, Computer Science and Object-Oriented Programming - Java, Networks and Digital Communications, Data Warehousing to Big Data, Statistical Methods and Probability, Enumerative Combinatorics, Abstract Algebra I and II, Linear Algebra, Multivariable Calculus, Vector Calculus, Differential Equations

## EXPERIENCE

---

### AI Research @ Su Lab

#### AI Research Intern

June 2023 – Present

**Technologies Used:** Development: Jax, Pytorch, OpenAI Gymnasium, D4RL, Maniskill, Mujoco, Adroit; Tools: WandB, Docker, Nautilus, Kubernetes (kubect!, PVCs, etc), Mamba/Conda

**Personal AI Projects:** <https://github.com/arth-shukla>; <https://wandb.ai/arth-shukla/projects>

**Technologies Used:** Development: Pytorch (Torch, TorchVision, Datasets, Dataloaders, Cuda), OpenAI Gym, HuggingFace Transformers Library, TensorFlow, Keras, Gensim; Concepts: 3D CV (DenseFusion, PointNet), Reinforcement Learning (PPO, DDQN, DQN), NLP (Transfer Learning, Embeddings, Attention); Tools/Technologies: WandB, BERT/DistilBERT Pretrained, Conda

#### 3D Computer Vision

- Use Pytorch and TorchVision to implement DenseFusion with altered loss + ICP Refinement to solve 6D Pose Estimation on YCB-Video Dataset to achieve 1<sup>st</sup> place in competition in graduate-level course *CSE 275: Deep Learning for 3D Data*: <https://github.com/arth-shukla/densefusion>
- Use Pytorch and TorchVision to implement *PointNet: Deep learning on point sets for 3d classification and segmentation*, Qi et al. 2017 for part segmentation on chair point clouds: <https://github.com/arth-shukla/pointnet-part-segmentation>

#### Reinforcement Learning

- Use Pytorch to make PPO Agent (w/ entropy regularization, advantage normalization, early stop w/ approx. KL Div, GAE, etc) and Gymnasium to consistently beat Mario level 1-1 and 1-4 in under 1600-2000 episodes of training: <https://github.com/arth-shukla/ppo-mario>
- Use Pytorch to implement DDQN from *Human-level control through deep reinforcement learning*, Deep Mind 2015 in Gymnasium to beat Mario level 1-1 in under 13000 episodes of training: <https://github.com/arth-shukla/ddqn-mario>
- Use Pytorch to make simple PPO Agent and Gymnasium to consistently beat CartPole in under 140 episodes of training: <https://github.com/arth-shukla/ppo-gym-cartpole>

#### Natural Language Processing

- Use HuggingFace Transformers library to fine-tune DistilBERT model (transfer learning) trained on Stanford Question-Answer 2.0 (SQuAD 2.0) to answer a question given some context (article, paragraph, etc): <https://github.com/arth-shukla/squad2.0-bert-question-answer>
- Use Pytorch and HuggingFace to fine-tune DistilBERT model (transfer learning) to classify and approximate sentiment for Stanford Sentiment140 1.4-million Tweet Dataset: <https://github.com/arth-shukla/sentiment140-bert-transfer-learning>
- Use TensorFlow Keras to build LSTM and CNN and use Gensim to refit GloVe word embeddings for IMDB Review Sentiment Classification: <https://github.com/arth-shukla/gensim-embedding-training-imdb>

#### ACM AI's Element.AI Competition <https://github.com/acmucsd/Element.AI>

I led development and organization of Element.AI, an \$8000 RL competition at UCSD with over 200 participants.

**Technologies Used:** Python, Conda, Java, Maven, OpenAI Gym, PettingZoo, PyGame, Jackson, Squid, Bash

#### Lead Developer

July 2022 – Feb 2023

- Use Python with Conda and PettingZoo ParallelEnv to create multi-agent gym environment based on popular PaperIO game
- Use Java with Maven and Jackson to create Java sdk for 45 participants (~22.5% of all participants)
- In coordination with UCSD ITS, use Squid proxy, IPTables and bash scripts to create instructor tools, allowing us to enable/revoke access to wifi, whitelist sites, enable/revoke access to files, and in general control the competition accounts with granularity, both targeted and en masse

- Write proposals and attend meetings to obtain \$8000 in sponsorships, attracting 200 participants (limited primarily by the number of UCSD Linux lab machines) with over 100 submissions

### **Nefeli Networks**

**Technologies Used:** Backend: Go, Docker, Kubernetes, etcd; Frontend: Angular, Less; DevOps: Git, Coder, Agile

### **Software Engineering Intern**

June 2023 – August 2023

- Use Go and Docker to integrate Infracost API in backend for Terraform cloud object cost and diff calculation, code used in production (23.09 release)

### **ACM AI UCSD**

ACM AI is UCSD's largest AI student org which fosters a community for those interested in AI and research.

**Technologies Used:** AI/ML Workshops and Projects: Python, PyTorch, TensorFlow, Google Colab; Web Development: TypeScript, React, LESS

### **Board – President**

May 2023 – Present

- Lead Operations, Competitions, Dev, and Marketing teams to develop competitions and workshops, and revamp our forward-facing resources (website, GitHub, ACM AI Wiki)

### **Board – Director of Operations**

May 2022 – May 2023

- Lead team of 7 event leads in creation of competitions, workshops, and socials related to ML/AI
- Coordinate with marketing and development teams to market events and create competitions

### **Board – Event Lead**

January 2022 – May 2022

- Coordinate with marketing, social, and other event leads to host workshops on NLP, deep learning, and ML topics
- Develop and organize competitions run by ACM AI (100-200 submissions on average)
- Mentor intermediate and beginner ACM Projects teams in developing AI/ML projects

### **ACM Projects – Machine Learning Engineer**

September 2021 – January 2022

- Create model to convert human faces to Cat-Human hybrid using DCGAN, PatchGAN, CycleGAN, and StyleGAN
- Coordinate with team of 3 front- and back-end devs to implement model into user-friendly tool

### **Bittner Development Group**

**Technologies Used:** Web Development: React, SCSS, Node.js; DevOps: GitLab, Git, WSL; Scripting and Automation: TypeScript, JavaScript, Java; Development Standards: WCAG 2.1 AAA, Aria APG, Norton Design System; Processes: Agile Methodology

### **Software Engineering Intern**

November 2019 – June 2023

- Use React and SCSS to develop component library and enterprise web application 'Interactive Builder'
- Web development, QA, and devops of over 10 education interactive projects in React to WCAG accessibility standards
- Manage and train two interns to complete projects using React and SCSS, GitLab, Git, and WSL
- Propose, lead, and develop internal and for-client automation projects using Node and native JavaScript

### **Personal Web Development Projects:** <https://github.com/arth-shukla>

**Technologies Used:** Web Development: TypeScript, React, Rollup, Jest, Webpack, Storybook, SCSS; DevOps: Netlify, Git, GitHub Pages, GitHub Packages; Development Standards: WCAG 2.1 AAA, Aria Authoring Practices Guide (APG)

### **Independent Developer**

- Personal website using React Typescript, SCSS, and Material UI, accessible by WCAG 2.1 AA standard: <https://arth.website>
- Icon Library with React TypeScript and SCSS, publish to GitHub packages: <https://github.com/arth-shukla/arth-components>; code demos and documentation: <https://arth-shukla.github.io/my-icons-documentation>
- Code mobile-compatible Dice Roller web app on React: <https://arth-shukla.github.io/dice-roller>

## **SKILLS**

- **Programming Languages** – Python, C++, Go, TypeScript, JavaScript, Node, React, Java, Ruby, Bash, SCSS, LESS, CSS, C, R
- **Packages and Libraries** – Pytorch, TorchVision, Jax, Keras, HuggingFace, OpenAI Gym, PettingZoo, Rollup, Jest, Webpack
- **Programs and Software** – Nautilus, Docker, WandB, Tensorboard, Git, GitHub, GitLab, Mamba/Conda, Maven, Visual Studio Code, WSL, Netlify, Storybook, Android Studio, Matlab, Microsoft Office, Microsoft Excel, Microsoft Powerpoint
- **Development Standards** – Web Content Accessibility Guidelines (WCAG) 2.1 AAA, Aria Authoring Practices Guide (APG)
- **Languages** – Fluent in English and French, Spoken Hindi