Arjun Patel

(609) 480-4071 • arjunpatel151002@gmail.com • arjun-u-patel • arjunUpatel

Education

Rutgers University • **New Brunswick, NJ** May 2025
Computer Science B.S. GPA: 3.8/4.0

Highlighted Coursework

- **Computer Science** Computer Architecture, Operating Systems, Deep Learning, Programming Languages and Compilers
- **Mathematics:** Linear Algebra, Multivariable Calculus, Randomized Algorithms

Work Experience

LLVM Compiler Infrastructure – Compiler Software Contributor September 2024 – Present

- Developed target address resolution for **RISCV** instruction sequences in **C++**, expanding support from **4** to **19** instructions
- Pioneered support for interleaved instruction sequences increasing the total number of resolutions by **7%**
- Improved disassembly readability, reducing developer reverse engineering and debugging by **10%**

TD Bank – Software Engineering Intern June 2023 – August 2023

- Secured user access by integrating support for elevated access authorization based on past and current user interactions
- Collaborated with **3 developers** within an Agile environment to successfully complete backlog items in Java
- Leveraged Splunk to pinpoint precise timestamps of occurring issues, facilitating efficient troubleshooting and resolution

Rutgers Residence Life – Resident Assistant January 2023 – May 2025

- Organized monthly events, engaging an average of **30** students to foster meaningful connections among residents on a floor
- Acted as an avenue for **100** residents to voice daily concerns and seek mental health, professional and academic resources
- Enforced university policy and mediated **5** roommate conflicts to create a safe and inclusive living environment for all
- Mentored and supported **11** new staff members for an easy transition to the role and seamless collaboration

Amneal Pharmaceuticals – Engineering Intern June 2022 – August 2022

- Created and conducted tests to validate readiness of **20** machines for use in production of Topical, Transdermal and Oral products
- Led the initiative to streamline the review process through online collaboration tools, resulting in a **70%** reduction in review time
- Followed Current Good Manufacturing Practice (cGMP) documentation guidelines to record all procedures in a timely fashion

Projects

Mine Learning – Python, PyTorch

- Explored the feasibility of utilizing deep learning methods to solve the game of Minesweeper
- Conceived a **Convolutional Neural Network** model capable of out performing logic based bots by opening **2 times** more tiles
- Developed and tested a **Transformer Neural Network** with a custom attention system to handle inputs of arbitrary sizes
- Reduced model training time by **1 hour** by minimizing disk IO operations executed to obtain data tensors

Danger Detector – Python, Anaconda, Numpy

- Created a logistic regression machine learning model to differentiate dangerous and safe wiring diagrams
- Extended functionality by implementing a softmax regression model to identify the ordering of the wires
- Optimized model training features, resulting in an average accuracy rate of **95%**
- Integrated techniques to combat over-fitting like hold out sets, data augmentation and early stopping reducing training loss by **55%**

RISC-V Optimizer – C, Bash

- Created compiler extensions to analyze RISC-V instructions and reduce the total number of operations before execution
- Implemented constant folding, strength reduction and dead code elimination optimizations to reduce number of instructions
- Automated testing with shell scripts to execute multiple test cases with ease

U-threads – C, Linux

- Implemented a threading library using linux ucontexts to enable concurrent processing
- Integrated mutual exclusion primitives for proper data security during concurrent processing
- Optimized processing time by implementing MLFQ scheduling algorithm, improving processing time by an average of **15%**

Hard threads – C, Linux

- Implemented cooperative multitasking in the Linux kernel by modifying core scheduling logic
- Designed and exposed custom syscalls to allow parent processes to suspend and resume child threads
- Conducted in-depth analysis of Linux scheduler internals, including load balancing and wait queue mechanisms

Type wrecker – Ocaml

- A polymorphic type-checking system in to verify type safety in a statically-typed language environment, enhancing code correctness
- Extended functionality to support polymorphic types, enabling the inference of generalized types, reducing the need for annotations
- Achieved a **15%** reduction in type-checking runtime by optimizing unification processes and minimizing redundant checks

Skills

- **Programming languages:** C/C++ | Python | Java | Ocaml | Web (HTML/CSS/JavaScript)
- **Strong understanding:** Git | LaTeX | Linux | VS Code | Neovim | PyTorch
- **Familiar with:** Unity | Firebase | React | SQL | Shell | Android Studio | Jira | ChatGPT