

(929)-676-7527  
New York, NY  
shuyangsun10@gmail.com

# Shuyang Sun

## Software Engineer

github.com/shuyangsun  
linkedin.com/in/shuyangsun

I am a self-motivated developer who likes to learn new things, work with smart people and learn from them. When I am not coding, I like to read work of Daniel Kahneman and Nassim Taleb.

\* H-1B visa sponsorship required to work in the U.S.

### SKILLS

**Programming Languages** C/C++, Go, SQL, Python, Rust, TypeScript, R  
**Other** Machine Learning, Data Engineering, Scientific Computing, API Design

### EXPERIENCE

**Software Engineer** Sep 2022 — Present  
*Google* New York, NY

#### • Contribution

- Designed and implemented multiple features that powered Geo (Maps) organization's user interaction analysis.
- Refactored Geo's most resource-intensive analysis tool, resulted in a 20 times memory usage reduction and 75% of compute resource savings, which saved the company millions of US dollars on infrastructure cost.
- Proposed, designed and implemented novel ways to remodel features of datacube, these features enabled our clients to perform analytical tasks in ways weren't possible due to high computational cost.
- Kick-started quarterly newsletter program for our team to foster better relationship with internal clients.

**Tech Lead** Oct 2021 — July 2022  
**Back End Developer** Mar 2020 — Oct 2021  
**Python Developer** Mar 2019 — Mar 2020  
*Haven Technologies* New York, NY

#### • Contribution

- Lead a team of 4 developers, responsible for the design and implementation of actuarial calculation engine in production.
- Built POC of a low-effort solution to use off-the-shelf YACC technology to replace manually implemented Lexer and Parser components of compiler for our in-house programming language.
- Designed and implemented a generic mathematical solve library from end to end, which is a wrapper of our GraphQL calculation APIs that can solve monotonic functions.
- Designed and implemented experimental actuarial pricing solver algorithm using linear programming.
- Lead developer of a new cloud-based actuarial pricing platform and other SaaS-related software, voiced concerns which resulted in a positive directional shift for the project.

#### • Lessons Learned

- Understanding business context and requirements in depth is the very first step in writing good code.
- Communicating the risk of inaction is more challenging than communicating the risk of action, a well-written proposal and analysis sometimes can help.

**Risk Manager** Jan 2019 — Mar 2019  
**Risk Engineer, Data Science Engineer** Nov 2017 — Dec 2018  
*CommonBond* New York, NY

#### • Contribution

- Lead developer of In-School business line decision engine. Implemented proxy class to solve data retrieval and validation issue using the idea of computational graph in ML (generates a DAG and retrieves values automatically).
- Initiated development of the C++ version of decision engine from scratch, increased credit report parsing speed by 500 times (200ms to 0.4ms). Reduced parsing duration for 224k credit reports from 10 hours to 90 seconds, dramatically sped up credit report feature engineering for multiple ML projects.
- Implemented multiple validation methods and automation in decision engine to prevent human error during pricing change.
- In-house IRR function implementation, Python version runtime speed beats Numpy version by 3 times, reduced runtime complexity from  $O(n^3)$  to linear time. C++ version beats Numpy version by 200,000 times (9ms vs 45ns). Reduced APR calculation per loan application on production environment from 15 seconds to 2 seconds.
- Developed machine learning model to filter high-quality direct mail marketing campaign target customers from credit pool, increased conversion by 20% comparing to standard channel.

# Shuyang Sun

(929)-676-7527

New York, NY

shuyangsun10@gmail.com

Software Engineer

[github.com/shuyangsun](https://github.com/shuyangsun)

[linkedin.com/in/shuyangsun](https://www.linkedin.com/in/shuyangsun)

- Developed loan portfolio carving algorithm for Capital Market team, the algorithm solves for target WAC while maxing volume and retain product composition. Worked with finance team, modified project requirements to reduce problem time complexity from NP-Complete to  $O(n \log(n))$ . Reduced carving process from days to seconds.
- Built deep learning model to predict probability of default on credit report data during recession, achieved F-1 score 0.67 on a 75/25 label split dataset.
- Built deep learning model to predict FICO score from credit report features, achieved MAE of 17 (out of 550).
- Built conversion prediction, product selection prediction models that surpass expert human-level performance.
- **Lessons Learned**
  - Don't use production environment as a playground to test out new technologies, prefer off-the-shelf tools.
  - Reevaluate different options after learning from mistakes, be aware of the sunk cost fallacy.
  - Be very careful with drawing conclusion from data, we don't know what we don't know.

## iOS Developer

Bytemark

Jun 2016 — Nov 2017

New York, NY

### • Contribution

- Worked on main mobile product Customer Mobile 4.0. A customizable iOS project template that generates apps based on configuration files dynamically.
- Worked with platform team to design REST API for efficient communication between iOS application and AWS server.
- Built deep learning models to predict e-ticket usage volume per hour, reached 64% accuracy.

### • Lessons Learned

- Good user research and UI/UX design can pay off in a big way, it's always good to validate the idea first.

## Software Engineer Intern

Workiva

May 2014 — Dec 2014

Ames, IA

- Worked on Find and Replace feature in web presentation app with Action Script, implemented unit tests and functional tests.

## PROJECTS

### Goo-GoL-Conway

Rust

- A modular cellular automaton framework designed from scratch, written purely in Rust.
- Demonstrated the modularity architecture by turning a cellular automaton with square cells into triangular cells with minimal code change.
- Implemented shaders and custom CPU/GPU-based graphics rendering pipeline using low-level library gfx-rs. Supports CUDA, OpenGL, Metal and CPU rendering.
- No use of unsafe code besides graphics rendering pipeline, memory-safe parallel computing and IO concurrency to support real-time playback as well as saving and loading replay.
- Check out [source code](#) or read the [research paper](#).

### Morse Transmitter

Swift, iOS Framework

- Designed and implemented iOS application that translates between text and Morse-code.
- Designed statistical algorithm on top of Fast Fourier Transform to decode Morse code audio into text in real-time, with excellent accuracy comparing to competitors.
- [Download](#) on App Store or check out [source code](#).

### Orthogonal Projection

CUDA, C++, Python, SciPy

- Researched mathematical function approximation methods based on linear algebra.
- Implemented Orthogonal Projection Calculator over inner product space of functions in Python and SciPy.
- Compared traditional gradient based approximation methods with projection method.
- Significantly improved performance of complicated CUDA kernels by 2.5x (on NVIDIA Titan X Pascal).
- Optimized implementation of polynomials, increased performance on GPU by 10% comparing to standard implementations.
- Check out [source code](#).

## EDUCATION

MS in Computer Information Science, Harrisburg University of Science and Technology

May 2019 — Mar 2021

BS in Computer Science, Iowa State University of Science and Technology

Aug 2012 — May 2016