

# Shizhuo Li

Email: [shizhuo@andrew.cmu.edu](mailto:shizhuo@andrew.cmu.edu)

LinkedIn: <https://www.linkedin.com/in/shizhuo-li-206109223/>

Profile Website: <https://www.shizhuo-profile.com/>

*Undergraduate senior majoring in computer science and minor in machine learning. Currently seeking SDE / MLE summer 2025 internship or new grad full-time positions.*

## EDUCATION

---

### **Carnegie Mellon University**

*Bachelor of Science in Computer Science*

*August 2021 ~ May 2025 (Expected)*

*Pittsburgh / Pennsylvania*

- **Academic:** Cumulative GPA 3.83 / 4.00
- **Field of Study:** Major in Computer Science, Minor / Concentration in Machine Learning
- **Related Courses:** Introduction to Machine Learning (phd), Advanced Natural Language Processing, Responsible AI, Distributed Systems, Computer Vision, Introduction to Computer Systems, Parallel and Sequential Data Structures and Algorithms, Constructive Logic, Great Ideas in Theoretical Computer Science, Principles of Imperative Computation.

## SKILLS

---

- **Programming:** Python, C, Go, Java, JavaScript, HTML, CSS, Standard ML, Sui Move
- **System and Software:** Data Structures and Algorithms, Distributed Systems, Software / System Development, Git, Node.js
- **Machine Learning Related:** Data Modeling, Pytorch, LLM, Machine Learning, Deep Learning, Natural Language Processing, Computer Vision, AWS Sagemaker, LangChain

## EXPERIENCE

---

### **Research Assistant: SUIGPT-LLM Code Generation for Novel Blockchain Languages**

*Carnegie Mellon University – LearnLab*

*May 2024 ~ now*

*Pittsburgh / Pennsylvania*

- Developed and explored methods in enhancing LLM code generation performance of Sui Move Programming Language
- Performed Data cleaning of a Sui Move code dataset using Python Pandas, and built optimized vector dataset for Retrieval.
- Designed architecture of applying Retrieval Augmented Generation, natural language parser, and Move code compiler in assisting LLM's code generation process.
- Conducted extensive testing of performance
- Assisted with prompt engineering, fine-tuning and in-context learning for SuiGPT, and constructed APIs for seamless LLM pipeline integration.

### **Team Project: Retrieval-Augmented Multi-hop Code Generation with CodeLlama and Unlimiformer**

*Carnegie Mellon University*

*March 2024 ~ April 2024*

*Pittsburgh / Pennsylvania*

- In team of 3, developed an advanced code generation system integrating Retrieval-Augmented Generation (RAG) and multi-hop execution to generate more reliable and compilable python code from natural language descriptions.
- Implemented iterative code generation using error messages feedback to LLM and the Unlimiformer, unlimited-length-input transformer, for allowance of longer input of LLM.
- Conducted extensive experiments using benchmark datasets, total of more than 20K python functions, outperformed the baseline model.

### **End-to-End NLP System Building – An InfoBox for LTI**

*Carnegie Mellon University*

*February 2024 ~ March 2024*

*Pittsburgh / Pennsylvania*

- Design and build an end-to-end NLP system LTI-"hat capable to answer question related to Carnegie Mellon University and its Language Technology Institute questions in team of 3.
- Collected, processed, and embedded information relevant to "CMU and LTI for vector database construction by scripting.
- Boosted LLM's performance of generation by in context learning and refining the pipeline of database search.
- Achieved good performance (0.84 overall recall and 0.73 precision) in answering on our set of self-annotated test questions.

### **Distributed Systems Team Project - LSP and Distributed Bitcoin Miner**

*Carnegie Mellon University*

*September 2023 ~ October 2023*

*Pittsburgh / Pennsylvania*

- Design and implemented novel network protocol: Live-Sequence Protocol (LSP) which combines advantages of UDP and TCP for purpose of distributed system inner communication using Golang
- Implemented a "Client-Server concurrent interaction model with LSP with network / machine failure tolerance and handling.
- Design and implemented a scalable distributed bitcoin miner with LSP's client-server model, and achieved expected performance.
- Boosted the speed performance of mining by refining scheduling algorithm, adapting quicker task splitting of miners, and more reliable task reallocating mechanism in case of minor failure.

### **Computer Systems Course Project – Malloclab**

*Carnegie Mellon University*

*March 2023*

*Pittsburgh / Pennsylvania*

- Implemented functions malloc, calloc, and realloc in C with newly designed data structures and algorithms such as explicit list and seglist
- Tested the performance including utility and speed performance of malloc of different data structures
- Refined the implementation with techniques such as mini memory blocks.