Shawn (Sihyun) Lee

✉ sihyunlee@cmu.edu ☎ 412-726-3704 🔗 shawnlee.io in sihyunlee shawnsihyunlee

Education

Carnegie Mellon University *Aug 2022 – May 2026*

BS in Computer Science, Concentrations in ML and Systems

- GPA: 4.0/4.0

Experience

Jump Trading *Chicago, IL*

Software Engineer Intern *June 2025 – Aug 2026*

- Connected Jump to a new exchange. Working in low level C++.
- Built the MVP of a new low-footprint internal protocol between strategies and exchanges.

Bodo.ai *Pittsburgh, PA*

Software Engineer Intern *May 2024 – Aug 2024*

- Managed BodoSQL's SQL decimal coverage project, adding native fixed-point decimal support to 30+ existing SQL commands, optimizing C++ array kernel performance by 300% using MPI, and maintaining/tracking current decimal coverage.
- Newly implemented SQL DDL commands for BodoSQL by writing JavaCC parsers to implement 20+ new commands, and expanding Bodo's compatibility with Iceberg.

CMU Language Technologies Institute *Pittsburgh, PA*

Research Assistant *May 2023 – Nov 2023*

- Independently created novel, universally applicable prompting technique which improves theory of mind capabilities of LLMs over current SoTA by **+29.5% absolute accuracy**, while part of the MultiComp Lab led by Louis-Philippe Morency. Published to ACL 2024 Main Conference.

CMU School of Computer Science *Pittsburgh, PA*

Teaching Assistant *Jan 2023 – Current*

- Teaching assistant for two of CMU's core computer science courses: 15-251 and 15-112.
- Led recitations of >20 students, held review sessions for >100 students, and provided close mentorship of >10 students every semester.

Publications

Think Twice: Perspective-Taking Improves Large Language Models' Theory-of-Mind Capabilities Jan 2024

Alex Wilf, ***Sihyun Shawn Lee***, Paul Pu Liang, Louis-Philippe Morency

10.18653/v1/2024.acl-long.451

Projects

C0 Compiler *Github Link (private)*

- Wrote a full compiler for C0, a safe subset of C, using OCaml. Incorporates optimizations such as SSA, register allocation, sparse conditional constant propagation, dead code elimination, loop invariant code motion, tail call optimization, function inlining, and strength reductions.
- Tools Used: OCaml

Skills

Languages: Python, C++, C, OCaml, SML, HTML/CSS/JS