# Fangyi Zhou

me@fangyi.io    |    https://github.com/fangyi-zhou    |    London, UK

## EDUCATION

**Doctor of Philosophy (PhD) in Computing**  Imperial College London                    Sep 2019 — Aug 2023
Thesis Title: Refining Multiparty Session Types (Awarded Mar 2024)
Topics: Multiparty Session Types, Refinement Types, Distributed Systems, Programming Languages

**Master of Engineering (MEng) in Computing**  Imperial College London                    Oct 2015 — Jun 2019
1st Class Honours, Overall 86.89%, **Dean's List** for Years 1, 2, 3 and 4
Awards: Adrian Israel Memorial Prize, Corporate Partnership Programme Prize, G-Research Ltd Prize, Corporate Partnership Programme Award, Governors' Prize (for **best overall performance**)

## EXPERIENCE

**Software Development Engineer/Applied Scientist**  Amazon Prime Video, UK                    May 2023 — Nov 2024
Studios AI Lab (Nov 2023 — Nov 2024): I worked as a Software Development Engineer on building an application for extracting and processing information from documents using **Large Language Models (LLM)**.

- Developed medium-sized features (e.g. supporting ingestion and translation of non-English documents), **independently and in collaboration** with other team members in an **agile** development environment. Partnered with product and UX teams to gather and implement feedbacks, and ensure timely delivery.
- Performed **Full-stack** engineering in a **micro-service** environment on **AWS**, utilising **TypeScript** and **React.js** for frontend development, **TypeScript** for backend services, and **Python** for machine learning implementations.
- Bridged science and engineering teams, by **providing crucial engineering support for scientists**, e.g. productionising prototypes, and establishing cross-codebase interfaces.
- Delivered infrastructure improvement for the team, improving resilience and reliability, e.g. migrating a key **CI/CD pipeline** for the ML codebase to a modern system with more flexibility and less maintenance effort.
- Achieved **significant performance improvements** by optimising system bottlenecks, e.g. patching 3rd-party libraries (reduced semantic search time from ~5min to ~2s) and optimising LLM usages (reduced processing time from ~30min to ~5min), improving the overall efficiency of the system.
- Contributed to team growth by **mentoring** junior team members, and delivering technical presentations.

Automated Reasoning (May 2023 — Nov 2023): I worked as an applied scientist on building a **static analyser** for JavaScript/TypeScript, where I modelled language features of JavaScript/TypeScript in the analyser.

Skills: Python, TypeScript, React.js, Java, Scala, Docker, AWS, System design, DevOps, Rapid prototyping

**Research Assistant**  Imperial College London/Universify of Oxford, UK                    Sep 2019 — Mar 2023
I worked in the Mobility Research Group under the supervision of Prof. Nobuko Yoshida, published high-quality research papers in programming language conferences, including a **distinguished paper** at ECOOP 2023, and developed research software and prototypes. (Sep 2019 — Sep 2022 at Imperial, Oct 2022 — Mar 2023 at Oxford due to supervisor moved)

Skills: OCaml, F⋆, Docker, LaTeX, Source code generation, Program verification, Research paper writing

**Graduate Teaching Assistant**  Imperial College London, UK                    Oct 2019 — Jun 2022

**Software Engineering Intern (Industrial Placement)**  Facebook, UK                    Apr 2018 — Sep 2018
I worked in Hack Language team on implementing new experimental language features, and in Sapienz team on improving categorisation of crashes of Android applications.

Skills: OCaml, Python

**Undergraduate Teaching Assistant**  Imperial College London, UK                    Oct 2017 — Mar 2018

**Research Intern**  Arm, UK                    Jul 2017 — Sep 2017
I worked in Security Research Group on specification and verification of data structures used in hypervisor software for embedded systems.

Skills: C, Bounded Model Checking, Program Specification, Program Verification

## SKILL

**Programming**  TypeScript/JavaScript, Python, OCaml, Scala, Java
**Tools**  Git, AWS, Docker, Shell Scripting, SQL, LaTeX, SMT Solvers

## COURSE CERTIFICATE

- Machine Learning Specialization by Stanford University & DeepLearning.AI on Coursera.               Oct 2023
- Deep Learning Specialization by DeepLearning.AI on Coursera.               Nov 2023

## PUBLICATION

[1]   A. D. Barwell, P. Hou, N. Yoshida, and F. Zhou. "Crash-Stop Failures in Asynchronous Multiparty Session Types". In: *Logical Methods in Computer Science* (2025). To appear. URL: https://arxiv.org/abs/2311.11851.

[2]   F. Zhou. "Refining multiparty session types". PhD thesis. Imperial College London, 2024. DOI: 10.25560/110416.

[3]   A. D. Barwell, P. Hou, N. Yoshida, and F. Zhou. "Designing Asynchronous Multiparty Protocols with Crash-Stop Failures". In: *37th European Conference on Object-Oriented Programming (ECOOP 2023)*. Vol. 263. LIPIcs. Received **Distinguished Paper Award**. 2023, 1:1–1:30. DOI: 10.4230/LIPIcs.ECOOP.2023.1.

[4]   A. D. Barwell, A. Scalas, N. Yoshida, and F. Zhou. "Generalised Multiparty Session Types with Crash-Stop Failures". In: *33rd International Conference on Concurrency Theory (CONCUR 2022)*. Vol. 243. LIPIcs. 2022, 35:1–35:25. DOI: 10.4230/LIPIcs.CONCUR.2022.35.

[5]   A. Miu, F. Ferreira, N. Yoshida, and F. Zhou. "Communication-Safe Web Programming in TypeScript with Routed Multiparty Session Types". In: *Proceedings of the 30th ACM SIGPLAN International Conference on Compiler Construction*. CC 2021. ACM, 2021, pp. 94–106. DOI: 10.1145/3446804.3446854.

[6]   N. Yoshida, F. Zhou, and F. Ferreira. "Communicating Finite State Machines and an Extensible Toolchain for Multiparty Session Types". In: *Fundamentals of Computation Theory*. Cham: Springer International Publishing, 2021, pp. 18–35. DOI: 10.1007/978-3-030-86593-1_2.

[7]   A. Miu, F. Ferreira, N. Yoshida, and F. Zhou. "Generating Interactive WebSocket Applications in TypeScript". In: *Proceedings of the 12th International Workshop on Programming Language Approaches to Concurrency- and Communication-cEntric Software*. Vol. 314. EPTCS. 2020, pp. 12–22. DOI: 10.4204/EPTCS.314.2.

[8]   F. Zhou, F. Ferreira, R. Hu, R. Neykova, and N. Yoshida. "Statically Verified Refinements for Multiparty Protocols". In: *Proc. ACM Program. Lang.* 4.OOPSLA (Nov. 2020). DOI: 10.1145/3428216.

## PROFESSIONAL ACTIVITY

- Programme Committee: CC 2025, ICE 2023
- Artifact Evaluation Committee: COORDINATION 2024
- Reviewer: PLACES 2022, Science of Computer Programming