One of the most fascinating aspects of academia is the opportunity to inspire and learn from students through teaching. I consider two elements essential in this process: an **interactive environment** and **continuous refinement for both teachers and students in a supportive setting**.

Interactive environment. I believe that teaching is a shared responsibility between teachers and learners to ensure deep understanding. During my time at Johns Hopkins University's Center for Language and Speech Processing (CLSP), I was inspired by the discussion-based and highly interactive teaching styles in seminars and research meetings. I saw how instructors guided collective reasoning, questioned assumptions, and co-developed understanding. Effective strategies they used include inviting students to explain their current understanding, tailoring materials to their backgrounds, posing interactive questions, and introducing deliberate pauses for reflection. In particular, my favorite tool is to explain with the whiteboard, especially during one-on-one meetings where learners and I can co-construct understanding.

Continuous refinement in a supportive environment. My favorite approach to teaching is through continuous growth and refinement. I see every class as an opportunity for my students and me to learn, reflect, and improve together. To achieve this, I strive to create an open and supportive environment where constructive feedback, no matter how tentative or incomplete, is welcomed, valued, and explored. I also encourage students to both give and receive feedback with confidence and to view mistakes as essential parts of the discovery process, as long as they use thoughtful and supportive language.

My Teaching Experience

Teaching Assistant: In 2020, I was a teaching assistant for two undergraduate courses at the Singapore University of Technology and Design (SUTD). The first was *Introduction to Algorithms*, a freshman-level course with approximately 150 students. Due to the COVID-19 pandemic, the course was conducted entirely online. I closely observed how the instructor managed the virtual classroom and maintained engagement through both synchronous channels such as Zoom and asynchronous ones like Slack, using them to support students, clarify concepts, and address logistical questions. In addition, I assisted in preparing assignments, grading, and proctoring exams. Through this experience, I realized that teaching large classes requires additional effort to sustain interaction and motivation. I also learned that students appreciate well-calibrated assignments that balance challenge and attainability. In the same year, I also assisted in teaching the sophomore-level course *Graphics and Visualization*, which had around 30 students. I communicated with them on Slack to help solve problems. For example, I found that one of their main challenges at the early stage was setting up the development environment on their computers. I then created a detailed step-by-step tutorial that guided them through the setup process and enabled them to start their coursework more efficiently. Across both roles, my teaching approach focused not only on helping students solve problems, but also on communicating interactively through platforms such as Zoom and Slack to gauge their understanding and anticipate the questions they might have.

Guest Speaker: I have also been fortunate to serve as a guest speaker several times to share the latest advancements in AI for Qualitative Analysis. These included talks at the School of Public Health at JHU, the HCI Research Club at Hong Kong for students interested in HCI, and a class at Western Michigan University where I introduced junior students to how I began my research on human–AI collaboration in qualitative coding. These guest lectures provided me with valuable exposure to varying levels of teaching experiences. Through them, I learned how to capture and sustain students' attention, tailor my presentations to different levels of understanding, and further develop my skills as an instructor.

My Mentoring Experience

I have been fortunate to mentor 10 students in total: 3 undergraduate students, 3 master's students, and 4 Ph.D. students, across SUTD, Singapore-MIT Alliance, and JHU. This experience has strengthened my ability to adapt my mentorship to each student's unique background and learning style. My mentoring philosophy as a mentor is to help **1) sustain their motivation and confidence through encouragement** and **2) supporting students' long-term career development**, rather than focusing solely on individual

projects. I learned this from my advisors, who offered me warm encouragement during those lowest points of my PhD and postdoc journey. I believe that encouragement is not only essential for providing emotional support, but also for inspiring students to challenge themselves, take risks, and realize their full potential. I also believe that the success of students reflects the success of mentorship and ultimately brings long-term benefits to both the mentor and the broader research community. For example, I informally mentored a Ph.D. student, Ruyuan Wan, then at the University of Notre Dame, whose research explores the intersection of HCI and NLP. She proposed that the methodological pipeline of qualitative analysis could inform and enhance approaches to subjective data annotation. Together, we conducted a literature review on a collection of academic papers. Our collaboration resulted in two workshop papers. The first was presented at the LLMs as Research Tools workshop at CHI 2024. She later worked with my another student, Haonan Wang, a CS master student at JHU, which led to another position paper at the HCI plus NLP workshop at EMNLP 2025. In that paper, Ruyuan was the first author and I served as the last author. In addition to Ph.D. students, during my postdoc at SMART, I also mentored a first-year undergraduate student, Erika Lee from the University of San Diego, who joined my project on human–LLM collaboration as a summer research intern. This mentoring experience was especially rewarding. Erika shared that participating in my research helped her succeed in a scholarship interview with the U.S. Department of Defense, which included a job offer to work there as a research scientist after graduation. As she once told me, *"I really look up to you as a researcher, and I'm grateful to have the chance to learn from you!"* Her message reminded me how meaningful mentorship can be in shaping a student's confidence and career aspirations. Overall, these experiences have strengthened my belief in the value of early research exposure and the importance of providing inspiration and guidance to young scholars.

Proposed Courses

My interdisciplinary background and prior teaching experience have prepared me to effectively teach both undergraduate and graduate-level courses in Human–Computer Interaction, User Research, Product Design, and introductory-level core computer science courses, such as *Introduction to Algorithms*. Here I list several example courses I will develop:

Human–AI Collaboration and Interaction: an advanced undergraduate and graduate course explores the emerging frontier of human–AI collaboration. Students will examine state-of-the-art technologies in HCI and AI, and analyze how AI functions within human workflows, as an assistant, advisor, and partner. The course critically investigates the human factors and values that shape these interactions, including autonomy, agency, oversight, trust, and accountability. Students will also reflect on next-generation human-AI collaborative technologies and discuss large-scale societal applications of human–AI collaboration, addressing the ethical, social, and technical challenges that may arise. Course activities will include system prototyping, in-class discussions, and literature review.

Computational Qualitative Analysis: an advanced course that would support students from computer science or information science to build basic skills from theories to practices of qualitative analysis, and equip them with toolboxs they need to use. Topics could include types of unstructured data in computer science research (e.g., user interviews, user logs, AI agent logs, etc.), theories such as Grounded Theory, Thematic Analysis; Practices include open coding, axial coding or themes development; Software and AI tools could include MaxQDA, NVivo, Atlas.ti, and their AI functions. Course activities will include in-class discussions, group projects, presentations, and hands-on qualitative data collection and analysis conducted in real-world settings.

Human Aspects of Software Engineering: a graduate-level course equips students with the skills to understand programmers' behaviors from a HCI perspective and to design human-centered tools informed by this understanding. Students will explore how programmers think, reason, and collaborate during software development, and how these insights can inspire the design of next-generation development tools and environments. Topics include programming behavior and cognition, program comprehension, code navigation, qualitative and quantitative methods in software engineering research. Course activities include literature reviews, in-class group discussions, data analysis, prototype design, and presentations.