Teaching Philosophy

Teaching, to me, is the most direct way to spread the knowledge I have learned and empower the next generation of computer scientists. My goal as an educator is not only to transfer technical skills but also to cultivate independent thinkers who can innovate responsibly and contribute to the evolving world of computing. I believe that innovation must be grounded in solid fundamentals—creative ideas arise from deep understanding and persistent practice.

My teaching philosophy is built upon three core beliefs: learning by doing, adaptive learning, and individualized education. I view the classroom as a dynamic laboratory where students learn through hands-on experimentation and problem solving. I strive to adapt my teaching to the diverse backgrounds, learning speeds, and interests of my students. Everyone has unique strengths, and effective education should help students discover and develop their own talents rather than conforming to a single model of success.

In practice, I emphasize interactivity, engagement, and continuous feedback. I prefer to guide my teaching through students' questions, discussions, and real-time reactions. By encouraging participation and curiosity, I help students actively construct their own understanding instead of passively receiving information. This approach not only improves comprehension but also nurtures confidence, creativity, and a long-term enthusiasm for learning.

Teaching Experience

For three consecutive years, I served as a teaching assistant for the undergraduate course "Operating Systems (Honor Track)" at Peking University. This small, selective class (around 30 students per section) focuses on hands-on systems design and requires each student to independently complete all the labs from MIT 6.828 JOS, building an operating system from scratch. The course aims to deepen students' understanding of OS design principles through real implementation. As the primary organizer, I was fully responsible for course logistics—including lab assignment design, in-class discussions, and grading—while continuously experimenting with different teaching strategies to enhance learning outcomes.

To help students grasp complex system concepts, I adopted an **interactive and problem-driven teaching style**. Before each lab session, I collected the main challenges students faced in the previous assignment and the key concepts they struggled with. I then used these insights to design targeted mini-lectures that addressed common pitfalls and encouraged peer discussion. Beyond the standard labs, I also organized student presentations and discussions on advanced topics closely related to the labs and recent research developments. Each student selected a topic of personal interest, gave a short presentation, and participated in collaborative debates. This approach not only strengthened their technical understanding but also trained their ability to communicate and reason about complex systems—skills essential for research.

During my tenure, the course became remarkably popular; enrollment was consistently oversubscribed. In Fall 2021, **five female students from the regular Operating Systems class requested to join my section** because they preferred the hands-on, exploratory learning environment I had created. After coordinating with the lead instructor, I supervised their entire lab sequence independently, organized dedicated discussion sessions, and set a customized assignment schedule and grading criteria. Their enthusiasm and progress were deeply rewarding—by the end of the semester, they all expressed that the course had inspired them to pursue further studies in computer systems. Notably, one of them later joined the Ph.D. program in operating systems at Northeastern University.

This three-year teaching experience provided me with a comprehensive understanding of **course organization, interactive instruction, and fair assessment design**. It also reinforced my conviction that effective teaching requires adaptability, empathy, and a balance between rigorous practice and creative exploration.

Mentoring and Advising

During my PhD study, I helped my advisors to mentor more than ten students across different stages of study—including undergraduates, masters, and Ph.D. students. Many of them have published their first papers under my guidance. I view mentoring as an extension of teaching: a personalized and long-term process of helping students build confidence, develop independent thinking, and discover their own research identity. My goal is to guide students not only toward academic success but also toward a sustainable mindset for scientific curiosity and collaboration. My first Ph.D. mentee, Yifeng Cai, conducted research in collaboration with Ant Financial on the security of mobile systems. His work focused on designing a more efficient on-device user authentication system to prevent unauthorized transactions on shared family phones. This is an important issue that affected millions of Alipay users. The resulting solution was later integrated into Alipay's production environment, enhancing the payment security of hundreds of millions of users. Under my mentorship, Yifeng published three top-tier papers and grew from a novice Ph.D. student into an independent researcher with his own vision.

After joining the University of Illinois Urbana-Champaign, I continued my mentoring work internationally. I helped my advisor to mentor a new PhD student, Hwiwon Lee, who initially struggled with the uncertainty of research progress and publication standards. I helped him plan the overall project timeline, set realistic milestones, and maintain focus on core contributions rather than overextending engineering efforts. Through regular meetings and iterative feedback, he gained confidence and successfully published his first paper at NeurIPS 2025. This is a milestone that marked his start of a security research career.

In addition, I have actively mentored female students and supported their entry into computer systems and security research. Meige Guan, a master's student at Peking University, began exploring energy-efficient in-memory architectures for large models under my guidance and is now preparing her first paper submission. Fan Zhang, a first-year Ph.D. student at UCLA, developed strong interests in AI security and trusted hardware. With the approval of her advisor, I guided her through the background of my recent work, TEESlice, helped her design a research plan, and provided continual feedback throughout the project. Her first paper is now under submission.

Across all these experiences, I have learned that effective mentorship requires both structure and empathy, setting clear goals while allowing students the freedom to explore. Watching them grow into confident and independent researchers has been one of the most fulfilling aspects of my academic journey.

Future Teaching Plans

Looking ahead, I am eager to contribute to both **core undergraduate education** and **advanced graduate training** in computer science. My background in systems and security has prepared me to teach a wide range of courses that bridge theory and practice, from foundational computer systems to specialized topics in AI and security.

At the undergraduate level, I am especially well-prepared to teach Operating Systems, having served for three consecutive years as the teaching assistant for the Operating Systems (Honor Track) course at Peking University. That course adopted the MIT 6.828 JOS curriculum, which gave me deep familiarity with both the technical content and pedagogical structure—from designing labs to guiding students in building a complete operating system from scratch. In addition to systems courses, I am also comfortable teaching other fundamental computer science courses, such as Data Structures and Algorithms.

At the graduate level, I plan to design and teach specialized courses closely aligned with my research expertise, including AI Security, System Security, and LLM Security. These courses will integrate frontier research with hands-on exploration, exposing students to emerging challenges at the intersection of deep learning, software security, and trusted execution. By combining real research papers, open-source projects, and guided experiments, I aim to help graduate students build not only technical competence but also the critical thinking and creativity required to conduct impactful research.