Teaching and mentoring are the most fulfilling and rewarding experiences as a professor. I am excited by the opportunity to nurture the next generation of computer scientists.

Teaching Philosophy

My teaching philosophy is helping students develop critical thinking and communication skills so that they can have a solid foundation to pursue any discipline in the future. Over the course of my research, I have realized the importance of communication skills but they have been long ignored in our education. Critical thinking and communication skills are not separated, but fundamentally related. Professor Suresh Venkatasubramanian once said "Writing is thinking". On one hand, critical thinking can lead to improved communication and writing. On the other hand, improved communication and writing can help sanity check the motivation (explaining why the problem is important) and polish the research idea critically.

Concretely, I will incorporate critical thinking and communication skills into my classes. When I teach, I will first regularly ask students to practice a small task (e.g., a small coding project that implements a technique they just learned) and then present their results in class. I will provide detailed and customized feedback on their presentations and how they should improve. Gradually, the students will be asked to practice bigger tasks (better building upon the previous small tasks) and then present their results, and I will provide detailed and customized feedback again. With increasing difficulty and multiple iterations, it will be practically useful and efficient to help student develop critical thinking and communication skills.

Teaching Experience

I was a teaching assistant for four machine learning courses during my PhD at University of Chicago: two for undergraduate levels and two for graduate levels. Although they were all machine learning courses, they targeted different audiences (undergraduate vs graduate), application domains (e.g., public policy), and depths (theory vs application). For all these courses, I was responsible for preparing problem sets, conducting recitation, grading, and holding lab sessions and office hours.

To become an effective teacher, it requires time to prepare a course and teach with a holistic perspective. The most memorable TA experience was when I was a TA for CMSC 25025 (also STAT 37601, a PhD level machine learning course for statistics students) in Spring 2017. This course is an introduction to machine learning and the analysis of large data sets using distributed computation and storage infrastructure. This course was taught by Professor John Lafferty and extremely popular across the campus. The class was over registered, over 80 students ultimately, which was a large amount of students for a private school. We had undergraduate and graduate students majoring in computer science, statistics, mathematics, economics, physics, and chemistry. The professor, leading TA, and I co-designed class materials, problem sets, and exams. To mix machine learning theory with practice, the leading TA and I would pick a real-world problem for each topic covered in this course. The students obtained hands-on experience by using machine learning methods they just learned to solve real-world problems on the distributed computing platforms, during which the leading TA and I would hold lab sessions and office hours to help them. I have learned that this practice can interest students, and allow a student who wants to delve into the topic further with enough information to pursue his or her interests.

Teaching Interests and Approach

My teaching, research, and work experience has covered a wide variety of subjects, including machine learning, causal inference, computer architecture, and software engineering. Given the need, I would readily commit the effort required to effectively teach undergraduate classes in these subjects. In addition, I would love to teach graduate level seminar courses focused on special topics such as machine learning for systems and data science for cloud computing. For these seminar courses, I will cover research areas such as resource management and scheduling, neural network pruning and training, and hardware accelerators. I will also invite guest speakers who are well-known researchers on related topics to my classes to introduce their most recent research, which will help student improve their communication skills by networking with these researchers. I would also like to create hands-on courses where students can build systems for emerging computing problems, and develop or apply machine learning methods for problems that have societal ramifications. As a part of such a class, students will read and discuss recent papers in depth and get research experience through sizable course projects. Since these seminar courses focus on interdisciplinary research, I would expect students from different backgrounds such as machine learning and computer architecture. When students are looking for group mates to work on course projects, I will encourage them to look for people from different disciplines. I believe students from diverse backgrounds can learn from each other, expand their skill sets, and improve networking and communication skills.

Mentoring Approach

I always enjoy mentoring students and seeing them grow. I have helped advise one master thesis at MIT (paper under submission), one undergraduate research project at University of Chicago (accepted to be an MLSys 2022 paper), and I have formally and informally advised multiple undergraduate students (one in chemistry major) inside and outside of my institutions such as University of Pittsburgh and University of Pennsylvania. The most important lesson I have learned so far is that every student is different. Some can be independent and complete a goal without extra guidance. Some may need more interactions and hands-on engagement. It would be great to know the style of each student in the beginning, but sometimes even the students have not figured out what types of the interactions work best for them. My approach is starting with frequent interactions to help students and myself figure out the pacing that we are both happy with, and then adjust the frequency of interactions gradually.

Another lesson I have learned is to respect students' research interests because interest is the best teacher. I am willing to adjust the original goal of a research project if the student is interested in pursuing a slightly different direction and I also think that change is doable. Ultimately, what I can offer to a student is how to do research rather than what research to do. Helping students pursue the directions they are interested in may also discover new research opportunities that I did not identify in the beginning, and I am happy to expand my skill sets and grow with the students together. As the master student I mentored at MIT, who graduated as a part of Tau Beta Pi and Eta Kappa Nu (two national honor societies for EECS and engineering), wrote in her master thesis:

> "To my mentor and supervisor, Yi Ding, I would like to say thank you for her patience and guidance during our countless discussions and struggles. Yi has always been there to answer any questions or offer feedback, and I am deeply appreciative of the time and passion she dedicated into the project."

The last but not the least lesson I have learned is the importance of **mental health**. During my PhD, I saw some of my peers struggled and even left the PhD program. The Huixiang event [1] in the community of computer architecture was also a shock to me. I was lucky enough to switch to a great advisor Henry Hoffmann during the latter half of my PhD and now I still stay in academia. As an advisor, I am committed to the mental health of my students, which I believe is much more important than academic successes.

[1] `https://twitter.com/vhuixiang?lang=en`