Teaching Statement

Yuhao Zhu

The opportunity to nurture future computer scientists and engineers both in a classroom and in a research setting plays a key role in my pursuit of an academic position. Over the past six year, I have enjoyed several teaching and mentoring experiences, from which I formed my teaching and mentoring philosophy.

Teaching Experience and Philosophy

Teaching Experience. My primary teaching experience comes from three semesters as a teaching assistant (TA). The courses that I have TA-ed for cover different levels of the curriculum spectrum, ranging from lower-division Introduction to Embedded Systems, upper-division Computer Architecture, to graduate-level Dynamic Compilation. In the two undergraduate courses, I led weekly discussion sessions, designed exams, and helped assign final grades. In the graduate course, I co-designed the syllabus, overhauled the programming assignments, and gave guest lectures.

Apart from teaching in a classroom setting, I have also co-delivered a tutorial on infrastructures for mobile computer architecture (MobiTools, co-located with ISCA 2016). We presented both software and hardware methodologies that allow computer architects to conduct studies on real mobile platforms. The tutorial was so well-received that, afterwards, we established a collaboration with Google's consumer hardware team whose lead was among the attendees.

Teaching Philosophy. My teaching experiences have shaped my teaching philosophy, which is centered around *balancing principles and practices*. I believe in the power of practice in computer science education, i.e., to let students explore real-world technologies with tangible software/hardware artifacts. However, it is important to balance principles and practices. Technologies and systems might come and go, but the core principles that guide the design of successful systems do not. For instance, the cache system design in microprocessors has changed dramatically with different technology trends, application needs, etc. However, the fundamental principle of locality exists forever. In a classroom, I will always teach a principle first before introducing designs that practice that particular principle, and explain the trade-offs between different practices. My goal is to equip students with core principles that enable them to come up with best practices for problems that are not covered in a classroom.

Course Offerings. As a faculty member, I will be qualified and excited to teach any courses related to computer systems and computer architecture as well as introductory computer science and engineering courses. I would also enjoy giving seminars on emerging topics in computer systems (cognitive systems, edge computing, crowdsourcing and human computation, etc.). My experience of co-founding and moderating the Emerging Topics in Computer System Design seminar at UT Austin gave me valuable perspectives of engaging junior students and cultivating new ideas. I plan to transfer that experience to teaching seminar courses.

Motivated by my research interests in systems and mobile computing, I am particularly interested in teaching a hands-on mobile computing course. The course will have two tracks: an application track and a system track. Students taking the application track would design and implement an innovative mobile application of their choice, and deploy it on an actual mobile platform such as a smartphone or an IoT-oriented development board. Students taking the system track would design system software that makes intelligent trade-offs between various system metrics such as performance, energy, and privacy. By targeting both senior undergraduate and junior graduate students, I expect that this course will prepare students for software engineer positions, and help me identify potential students to do research with.

Mentoring Experience and Philosophy

Graduate Student Mentoring. My longest and greatest mentoring experience was with Matthew Halpern, who joined me on the event-based scheduling project when he first joined our group. I gave him specific tasks, but also encouraged him to think about the big picture of the project. Through our frequent interactions, he made an astute observation that eventually became a critical piece of the final paper. As the mentorship continued, Matt came up with his own idea on mobile CPU evolution, which I continued to mentor him on. While he was driving the project, I consistently offered my feedbacks and worked closely with him on the paper writing, which led to his first major conference publication at HPCA 2016. Matt is now fully capable of leading independent research, and has started mentoring junior students.

I also mentored another fresh graduate student, Wenzhi Cui. I suggested a project idea about understanding tail-latency in event-driven servers to Wenzhi, who then quickly took ownership of it and made significant progress. I acted as an advisor by pushing him to think about the project direction as well as discussing technical details. In the end, I was very proud that Wenzhi submitted his first tier-one conference paper as the lead author.

Mentoring Philosophy. As my advisor's first graduate student, I get to mature my own research style while mentoring junior students, which allowed me to form a research methodology with mentorship built in mind. I am excited to share with students the **IDEAL** research methodology: Identification, Definition, Exploration, Assessment, and Learning.

- IDENTIFICATION. Identifying a research topic is the first step for graduate students. I believe good research topics should have both real-world impact and scientific merits. One effective approach to identifying such research topics that I would like to share with students is to spare some time everyday reading technology websites and news articles. The focus of such readings is to identify issues that the general public care about (and thus have broad impact) and to think about what scientific breakthroughs are needed to address those issues.
- DEFINITION. Clearly defining the problem scope, after identifying a research topic, is vital to any research projects. In my observations, the *single most expensive mistake* that graduate students tend to make is skipping the scope definition step and jumping to the solution exploration step too quickly. Often, it leads to months of time wasted on searching solutions to an "undefined" problem. I learnt two principles about defining problem scopes: if the problem identified is too specific, generalize it and scope the problem in a broader context; if the problem identified is too big, break it down into multiple smaller problems where each has a clear scope.
- EXPLORATION. Exploring *creative* solutions to a well-scoped problem is what we all enjoy as scientists and engineers. I believe creativity can be trained. Many psychology studies have shown that creativity comes from the reuse of ideas, or in other words, to take ideas that have worked well in one context and apply it by analogy to another context. A creative solution is nothing more than the juxtaposition of a particular idea in a particular context that people have never though of before. To freely "reuse" ideas, however, one must master a lot of ideas. I will encourage students to extensively read papers, particularly those that are not directly related to their current research, to create an idea repository that future projects can borrow from.
- ASSESSMENT. Assessing the practicality of a solution is particularly important to computer system research. I encourage students to assess solutions not only under the constraints of today's technology, but also to factor in future technology scaling to create systems that have a long-term impact.
- LEARNING. The research loop is never closed without learning from the success and failure of the past. I believe delivering good research talks is an important way for learning because a good talk usually means positive and insightful feedbacks, which motivate better research.

Finally, I am a strong proponent of long-term investments on students. In my first semester working with my advisor, he spent over a thousand dollars to send us to a half-day course on presenting data and information. He later sent us to other venues such as training mental agility. Although some of these investments might have not paid off immediately toward a paper—some never will—it gradually shaped my mindset beyond just becoming a better graduate student to becoming a better person, which I forever benefit from. As a faculty member, I expect to offer no less to others.

Fostering Diversity

I am committed to improving diversity because I believe it is through diverse groups and different perspectives can we come up with ingenuous ideas. I have been actively involved with UT Austin's Women in Engineering Program where I mentored two female sophomores on their academic development. One of them, Hannah Peeler, won a merit-based scholarship at the end of our mentorship based on her demonstrated potential. I was also a speaker at a high-school summer camp to encourage female students to pursue STEM careers because I believe fostering diversity is a long-term process that must begin at the K-12 stage. Apart from mentoring and outreach activities, I also contributed to diversity through service. As the PC Chair of Tiny Transactions on Computer Science, I intentionally created a diverse program committee with people from different countries and with different genders.

The diversity issue will become even more important as our society is increasingly relying on artificial intelligence (AI) and machine learning (ML) technologies. ML and AI require massive data sets, which, if are not sufficiently diverse, will lead to systems that have inadvertent biases. For instance, learning form data sets containing only behaviors from men will produce systems that are likely bias toward men—an issue dubbed "a sea of dudes." I plan to take on research projects in collaboration with crowdsourcing and social computing colleagues to develop methodologies that integrate and manage diversity in computing systems, and thus contribute to diversity from a technological perspective.