Teaching Statement

Xudong Sun (UIUC)

I am excited to teach, advise, and work with students as an educator in academia. My philosophy as an educator is to focus on students and help students learn in a self-driven and effective way. I also connect my research to my classroom teaching to help students learn basic knowledge and cutting-edge research through practice.

I have prepared myself through my classroom teaching experience during graduate school. I have been the teaching assistant (TA) for the Advanced Operating Systems course. I have regularly given guest lectures to courses on computer systems and software engineering at UIUC. I have also given invited lectures at the University of Virginia and Brown University. I have regularly co-organized and helped to organize research seminars on computer systems and software engineering at UIUC. In addition, I have created and co-organized a new seminar on systems verification at UIUC.

In addition to classroom teaching, I have mentored five junior PhD students, one master's student, and three undergraduate students in different research projects; four students are from underrepresented gender and racial groups. Six students co-authored research papers published at top conferences (e.g., OSDI, SOSP, NSDI) with me, the other three students are currently leading exciting projects. Two of the undergraduate students are pursuing their PhD at UC Berkeley and master's degree at Harvard, and the master's student is working on cloud infrastructures at Apple. During the mentoring process, students learned how to define and solve research problems independently.

1 Classroom Teaching

I **incorporate my research into my classroom teaching**. I believe that the impact of research is not limited to industrial adoption, follow-up projects, and citations—research should also benefit students through teaching. Many of my research projects have been incorporated into courses at UIUC. For the Advanced Operating Systems course, my research Sieve and Acto are used to teach students how Kubernetes (the most popular cluster manager) works, what are the critical reliability problems in Kubernetes, and how to detect them. For the Topic in Software Engineering course, my research Ctest is used to teach the configuration problems in cloud systems and how to address them using testing. I have regularly given guest lectures in these courses to teach students about my research projects and how they can apply them in practice. Both Acto and Ctest are used to design programming assignments and course projects thanks to the efforts made by the course instructors and TAs; the course projects teach students how to use my research to find new bugs in real-world, widely-used cloud systems, and how to fix them by working with the system developers.

I create **new resources** to help students learn about **interdisciplinary research**. When I was working on a project on systems verification—an interdisciplinary area that combines formal verification and computer systems research, I found that there was no course on this topic at UIUC which created some learning barrier. After talking to a few professors and graduate students, I realized that I was not the only one who encountered this challenge. So I created and co-organized a new research seminar on systems verification: a forum to discuss cutting-edge techniques and research on systems verification. The seminar was held weekly throughout the semester and attracted many students and professors with a background in computer systems, formal verification, and their intersection. Several students later told me that the seminar was very beneficial to their research.

I make sure students are **well motivated** before they learn. To help motivate students, when designing my lecture I make sure to explain to students why the lecture is **relevant** and **exciting**. For one lecture I gave to the Topic in Software Engineering course, my goal was to let students understand my research on preventing configuration-induced failures. I started the lecture by asking students about their personal experience of configuring computer systems, and my questions triggered students to talk about the mistakes they made when configuring systems and think about how to prevent such mistakes proactively. I then connected students' personal experience to large companies like Google and Meta that also suffer from misconfigurations, and explained how my research addresses this problem. Several students told me that they enjoyed my lecture and felt excited to work on this topic after my lecture.

I design my lecture following a **backward** design by first deciding the learning objectives and then preparing the lecture material accordingly. For example, when preparing for a lecture that covers two research papers on operating systems (OS) reliability, I planned the learning objectives following Bloom's Taxonomy [1]: Students should *understand* the techniques from the two papers, but more importantly, students should learn how to *create* new techniques to address new problems. For that lecture, I focused on delivering the high-level long-lasting messages from the two papers such as "device drivers are a significant cause of system failures" and "isolation is the key to tolerating driver failures." After talking about the techniques presented in the two papers, I encouraged students to generalize the knowledge from the papers, think about other reliability problems in OS, and propose their solutions.

2 Research Mentoring

My mentoring philosophy focuses more on helping students become mature and independent researchers through **experiential learning** [2]: Students learn by doing, and I closely work with students and provide feedback timely.

To help students get started on doing research and build confidence, I encourage students to start by making small, concrete contributions to a project. I implement this approach by being **hands-on** and carefully designing **concrete onboarding tasks** when starting to work with a student. When designing onboarding tasks, I ensure that solving each task does not require too much background knowledge, and each task is incrementally harder than previous ones so students need to constantly innovate new solutions. For example, when working on the Anvil project (formally verifying cluster management systems), to help Wenjie Ma, an undergraduate student who had never worked on cluster management systems, get started, I designed a series of tasks including fixing a small broken lemma, writing a new lemma for proving a similar theorem, and combining these lemmas to prove a top-level theorem. After this process, later Wenjie managed to finish thousands of lines of proof independently.

I strongly believe in the importance of **feedback**: I give feedback early and regularly to students, and get feedback from them on what I could do to improve their experience. For example, when mentoring X (anonymized for the student's privacy), a third-year PhD student who is still looking for an academic advisor, I designed a project that fits X's interests and skill set to help X accumulate research experience and get through this stressful phase, and provided early and regular feedback along the way by having one-on-one weekly meetings to concretize the plan for each week. However, X was not very productive initially, and after a conversation I realized that X was too shy to ask for technical help. So I adjusted my mentoring style according to X's feedback by having multiple on-demand meetings each week and doing pair programming with X when X was blocked, which helped improve X's productivity. Since then X has contributed thousands of lines of code to the project.

To help students become independent researchers, I will be **gradually hands-off** and encourage students to **take leadership** by thinking independently and exploring their own ideas after they have accumulated enough experience and progress in the project. For example, when mentoring Wenqing Luo, a master's student, on the Sieve project (testing cluster management systems), I let Wenqing be in charge of several important system designs, and Wenqing designed and implemented some novel testing policies and testing oracles, which all become the essential parts of the project. The experience later inspired Wenqing to lead a follow-up project which finally turned into his master's thesis.

Given my experience, I am prepared to advise students and help them succeed. I will encourage students to get started with concrete, low-risk projects initially to learn basic research skills and guide them to eventually identify the research problems they care about. For each student, I will provide feedback regularly throughout the semester and hold a review each semester. Besides research, I will also help students to figure out their long-term career goals, prepare them for their goals, and reach out for their potential career opportunities. I will also adjust my advising style according to students' feedback so that I can do my best to improve their experience, learning, and research.

3 Curriculum Ideas

My research improves the reliability of computer systems using verification and testing techniques. Given this background, I can teach undergraduate and graduate courses in these areas, such as **distributed systems**, **operating systems**, **program verification**, and **software testing**. I also plan to incorporate my research into my teaching.

I am interested in developing a graduate course on **the reliability of cloud systems** that covers techniques to prevent, detect, tolerate, and diagnose failures in cloud systems. I plan to incorporate my research into my teaching. For example, I plan to develop course projects to teach students how to automatically and effectively detect bugs in modern cloud infrastructures based on my previous testing research including Ctest, Sieve, and Acto. The projects will also teach students how to communicate and work with developers to fix the detected bugs.

I am also excited to incorporate **systems verification** into the curriculum. This could start with either a seminar or a graduate course on this topic. I plan to integrate my previous research Anvil into my course. Anvil is a framework for verifying safety and liveness of cloud system implementations. I plan to use Anvil as a platform to teach students how to perform Hoare logic reasoning, state machine reasoning, and temporal logic reasoning to verify safety and liveness of real-world cloud system implementations.

References

[1] Lorin W. Anderson and David R. Krathwohl. *Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Longman, 2010.

[2] David A Kolb. *Experiential Learning: Experience as the Source of Learning and Development*. Prentice Hall, 1983.