Teaching Statement

Tianyin Xu
Department of Computer Science
University of Illinois Urbana-Champaign

I enjoy teaching as an essential component of research and education. I am committed to inspired teaching, and it is rewarding to know that students enjoy it. I was selected to the List of Instructors Ranked as Excellent (one as Outstanding) *seven* times, three for CS523, two for CS598TXU, and two for CS591(SN/DS). I also enjoy mentoring students through research and working with a diverse group of excellent students.

1 Lecturing

I enjoy giving lectures in classrooms. I have taught the following courses (a few of them were done in virtual classrooms due to the COVID-19 pandemic):

- CS 523: Advanced Operating Systems (five semesters)
- CS 598TXU: Reliability of Cloud-Scale Systems (three semesters)
- CS 423: Operating System Design (three semester)

CS 523: Advanced Operating Systems. The Advanced Operating Systems course was my favorite course when I was a Ph.D. student at the University of California San Diego (UCSD). It was a course that walked me to the door of computer systems research and helped me build a foundation and develop a taste. I want my students to have a similar learning experience as I did. Therefore, when I took over the course, I revamped the original CS-523 course design—I took my favorite parts of the UCSD course, namely the lectures and in-class discussions, and adjusted the course materials to fit the CS curriculum at the University of Illinois.

The objective of CS 523 is to teach operating system design from a research point of view. Concretely, the course covers key topics of operating systems, walks students through the evolution of many different systems designs and techniques, and examines their usage in important historical and modern systems. The course materials come from seminal, noteworthy, or representative research papers (typically describing the design and implementation of a seminal system). The lectures discuss the key insights and techniques of those papers. Based on the needs of the students (e.g., preparing Ph.D. qualification exams) and their research interests derived from the research groups in the CS department, I developed course materials over time to cover papers that are more useful for students to read and discuss.

Learning by doing systems research is an important component of the course. During the course, the students conduct a mini research project, from proposing an idea, to performing a feasibility study, and to designing, implementing, and evaluating the proposed techniques. Many course projects were further developed into full-fledged research projects, with research papers published at top venues (several won paper awards), as listed at `https://cs523-uiuc.github.io/pub.html`. It is rewarding to see CS 523 playing a role in students' initial research experience and career development.

An emerging challenge of teaching CS 523 in the past two years is the rapid increase of the class size. In Spring 2024, the class size doubled to about 60 students compared to Fall 2019, driven by the growth of the CS department and our graduate programs. With a large class of junior Master's and Ph.D. students, a few teaching practices, originally designed for a 20-30 Ph.D. student course, are no longer effective. I have been experimenting with new designs to accommodate the scale of the class, with the principle of providing more research guidance. For example, I changed the paper discussion from a chronological order where classic papers (such as THE OS, Tenex, and Mach) are discussed first, to directly discussing recent seminal papers to provide students with a modern view of systems research and help them understand emerging research problems, with classic papers discussed as related work in context.

CS598TXU: Reliability of Cloud-Scale Systems. I created the Reliability of Cloud-Scale Systems course in Fall 2018 and taught it for three semesters (Fall 2018, Spring 2019, and Spring 2022). It is a course that closely connects to my research, so I am always passionate about teaching it.

The objective of this course is to teach the principles and practices of reliability engineering in modern "cloud-scale" systems, and expose students to the research of software and system reliability. The students

study how large-scale systems fail in the real world, and then learn reliability techniques and practices, including those widely adopted in industry and new ideas proposed by academia. I designed the course based on my research and industry experience (I worked at Facebook Core Systems on production system reliability). The syllabus contains both industry practices and research proposals—every lecture first explains the state of the industry practice and then discusses state-of-the-art research. The format of CS 598TXU is similar to CS 523 described above, including discussion-based lectures and a semester-long research project.

In the longer term, I plan to (1) turn CS 598TXU into a regular 500-level course and (2) design a corresponding 400-level course on Reliable Software Systems. It will better connect my teaching with my research. Certainly, it would need thoughtful planning—with the current teaching arrangement of the Systems and Networking area, I would like to focus on CS 523 and CS 423, which are more fundamental, must-have courses. Fortunately, with the successful recruitment of new faculty members (see my Service Statement), I plan to rotate the teaching of CS 523 and CS 423 with my colleagues, and develop CS 598TXU further. My syllabus of CS598TXU is used by colleagues to create similar courses at other institutes, including Purdue University, Rutgers University, the University of Illinois Chicago, and the University of Virginia.

CS 423: Operating System Design. I taught three semesters of the Operating System Design course, an upper-division undergraduate course. I largely reused the course design, including the syllabus and machine problems (MPs). By design, the course uses the Linux kernel for MPs, instead of educational OS kernels such as xv6 and Nachos. The design, on the one hand, provides a real-world kernel development experience; on the other hand, it poses significant challenges for students who do not have experience with OS kernel programming (which is the majority of the class). The main change I made is to replace the last MP (which was moved to a system security course) with a new MP on implementing kernel extensions using eBPF and Rust. eBPF and kernel extensions are emerging techniques of the Linux kernel and operating systems (OS), which are important to cover in this course.

I find it challenging to teach CS 423. One main challenge is a lack of mature autograding infrastructure. We do have automated grading scripts; however, kernel bugs often lead to crashes of the entire system and make it hard to give partial credits. Moreover, the current grading scripts are not comprehensively implemented and cannot be passed to students to test their kernels locally. Consequently, the grading is laborious and time-consuming, leading to prolonged feedback. The second main challenge comes from the steep learning curve of Linux programming, which is difficult for students who have yet to take any OS course. For example, I observed that students who took ECE 391 Computer Systems Engineering (which teaches how to write a small OS from scratch) consistently outperformed students who only learned userspace programming; however, ECE 391 is not a prerequisite of CS 423, and most CS students may not be able to take ECE courses. Addressing these challenges requires careful rethinking and course development, which I would like to do, with the help of my colleagues, as the major teaching effort for the next phase.

2 Research Seminar

Seminars are essential platforms for faculty and students to present and discuss research, and build research communities. I have been actively organizing the following two research seminars:

- CS 591(SN/DS/IG): Systems Reading Group (eleven semesters)
- CS 591SE: Software Engineering Seminar (seven semesters)

CS 591(DS/IG/SN): Systems Reading Group. I created the Systems Reading Group in Spring 2019 and have been organizing it since then. Today, the Systems Reading Group is among the most actively attended seminars in the CS department, with a regular attendance of 20–30 faculty and students (including many students who do not take course credit). It is currently co-organized by four faculty members (Ramnatthan Alagappan, Aishwarya Ganesan, Indranil Gupta, and me).

The goal of the Systems Reading Group is to provide a venue for faculty and students working on systems research to meet and discuss research on a weekly basis. The seminar includes internal presentations, external talks, and special seminars. For internal presentations, students discuss their research projects and papers related to their research. External talks are invited by the organizers. We also encourage students to attend other related talks such as faculty job talks and department colloquium talks related to systems

research. We maintain an active website for the seminar and related talks, which has been a go-to resource of systems research at the University of Illinois: https://systems-seminar-uiuc.github.io/.

The organization of the Systems Reading Group has gone through a long way. My initial plan was to organize a research seminar for the entire Systems and Networking (SysNet) research area—I heard critical needs for research seminars and the Spring 2019 seminar was organized in that way. However, I realized that the SysNet research area is too diverse for a unified seminar. The SysNet faculty comes from at least six research communities and publishes papers at different venues. As a result, it is difficult to have in-depth discussions in the seminar due to diverse interests. Later, I made the seminar focus on operating systems and distributed systems to foster deep discussions and collaborations. Another challenge was to get faculty to attend the seminar. Faculty is busy, especially senior faculty. For faculty not working closely together, there is no strong incentive to attend a weekly seminar. By narrowing down the focus, we have successfully built a strong community of faculty and students who regularly attend the seminar and discuss research. We will continue running the System Reading Group to deepen and broaden the systems research community at the University of Illinois.

CS 591SE: Software Engineering Seminar. As a member of the Software Engineering (SE) research area, I also actively co-organized the Software Engineering Seminar from Fall 2019 to Spring 2023. Unlike the Systems Reading Group I created in 2019, the Software Engineering Seminar has been running for over ten years. It was one of the best-organized seminars, with very active participation. Organizing the seminar was a joy. In fact, I borrowed many elements from the SE Seminar to the Systems Reading Group, e.g., having senior students help with the organization, organizing writers' workshop to help with paper submissions, and encouraging practice talks. Since my research interest has gradually shifted away from Software Engineering, I stopped organizing the SE seminar after Spring 2023 to focus on the Systems Reading Group.

3 Student Mentoring

My philosophy of mentoring students is to respect and trust them as colleagues rather than treating them as teenagers or managing them as employees. This means respecting their working styles, trusting their work ethics, and focusing on *helping* them learning how to do research. Moreover, I encourage students to build teams and work together instead of working alone in isolation. In my experience, working together leads to more happiness and productivity. Students can learn from each other and avoid being bottlenecked by the advisor. This is particularly important for systems research—no big system is built by a single person.

Ph.D. Students. I am advising six Ph.D. students (including a co-advised student). My first Ph.D. student, Wenyu Wang, whom I co-advised with Professor Tao Xie, graduated in 2022 and joined Meta as a research scientist. My student, Xudong Sun, will propose his Ph.D. thesis in Spring 2024 and plans to pursue an academic job in 2025. I design a continuous research agenda for each student based on their background and research experience. For students who have less research experience and need more time to catch up, I engage them in existing projects to work with and learn from senior students. When students are ready to lead projects, I help them develop their own projects, build the team, and guide them through the projects.

Master's Students. I am advising four Master's students (including a co-advised student) and have graduated six MSCS students with thesis (including two co-advised students) and three MCS students without thesis. Two MSCS students, Qingrong Chen and Andrew B. Yoo, were recognized as Siebel Scholars in 2020 and 2021, respectively. I use the same approach of advising Ph.D. students discussed above for Master's students. Since the Master's programs are shorter term than the Ph.D. program (two years for the MSCS program and 1.5 years for the MCS program), I tend to design relatively less open-ended but more concrete research projects that can be accomplished in the expected time frame.

Undergraduate Students. I have worked with 15+ undergraduate students from the University of Illinois (both CS and ECE), the CS REU program, and the UIUC+ Summer Undergraduate Research in Software Engineering program. I engage them in my research projects to help them experience systems research and develop research skills. Most of these undergraduate students pursued further study at top graduate programs of CMU, Columbia, NCSU, Purdue, University of Illinois, UCSD, University of Michigan, etc. One undergraduate student, YiFei Zhu, received the Highest Honor at Graduation and the University Honor.