Many areas in Electrical Engineering and Computer Science (EECS), including autonomous systems and cyber-physical systems (CPS), are interdisciplinary and require a strong background in mathematics, control, programming, and physics. To attract more students to the research frontier, we need a well-structured training process to provide students with a solid foundation, keep them engaged, and encourage them to make breakthroughs. My goal as an educator is to strengthen the process of training creative young minds in EECS, especially in autonomous systems and CPS. After gaining experience as a teaching assistant (TA) and mentoring junior students at the University of Illinois at Urbana-Champaign (UIUC), I have developed an understanding of how I would like to approach this goal. This would be through (1) exploring innovative teaching technologies in accordance with students' aptitude, (2) connecting theory to practice in teaching, (3) designing project-oriented courses for advanced topics, and (4) helping students grow to conduct independent research. In the following, I will briefly summarize my teaching and mentoring experiences, discuss my plan for teaching as a faculty member, and reflect on my proposed approaches and methods for effective teaching.

Teaching and mentoring experiences: I grew up on a university campus in an underdeveloped part of China in a dual-professor household. The conviction that education is the chief enabler of personal growth and social progress is baked into my belief system and has motivated me to pursue an academic path.

At Illinois I was fortunate to have several opportunities to teach undergraduate and graduate students. I have been a teaching assistant for the undergraduate course *ECE 313: Probability with Engineering Applications*, with about 300 students. I have also been a regular guest lecturer for *ECE 584: Embedded System Verification*, with about 20 graduate students. Unlike most courses, for which TAs are mainly responsible for grading homeworks and exams, those courses gave me valuable experience in developing an innovative adaptive learning forum, designing materials for advanced topics, and incorporating my own research, as I will describe below in *Approaches and methods*.

As a Ph.D. student, I have had the pleasure of mentoring six talented undergraduate and graduate students with diverse backgrounds. Under my guidance, we developed software, participated in competitions, and published research papers. All the mentored students continued research in various graduate programs (including at UIUC) or self-driving startups.

Teaching plan: I am ready to teach senior undergraduate- and graduate-level courses on control theory, model checking, and random processes. With some additional preparation, I will be able to teach courses on robotics, algorithms, computer engineering, probability, machine learning, and distributed systems.

I am also eager to develop a new course on *Safe Autonomous Systems*, based in part on my own research expertise. This course will cover fundamental techniques for designing autonomous vehicles, including object detection and perception, localization, and human and environment modeling. It will also extend to advanced formal techniques for safe design and analysis of autonomous systems. There will be hands-on assignments using commercial simulation tools, and the students will be able to identify interesting research problems as course projects, which can potentially be implemented on various vehicle platforms, such as 1/10-scale race cars and real cars. My lectures for ECE 584 on verification would be a starting point for this course. I also plan to incorporate materials from a new course I am currently developing with Prof. Sayan Mitra at UIUC, called *Principles of Safe Autonomy*.

Approaches and methods:

1. **Adapting to students' aptitude with technology.** In big classes, students have wide distribution of backgrounds, learning styles, and goals. I believe it is important to respect and indeed nurture their differences. ECE 313 had around 300 students; some were gifted in math and eager to tackle challenging problems, while others struggled and found homework to be overly time-consuming. The large class size made a high-touch individualized solution impractical. We built an online adaptive learning platform called *Masterprobo* to mitigate this situation. Instead of using a fixed set of homework and quiz problems, Masterprobo provides a variety of ways to train students based on their own learning styles and progress. For example, if a student chooses the "adaptive" mode, the difficulty and number of problems in the homework will dynamically change based on the problems he or she has solved, the hints he or she has resorted to, and the time he or she has spent on each problem. The student can also choose the "challenge" mode to do a smaller set of more challenging problems. Moreover, when we held online office hours on Masterprobo, we could assign easier problems that could be solved using similar methods to better clarify the elements that confused

the student. As a side effect, Masterprobo also helped resolve the other important problem in big classes: plagiarism. This was because Masterprobo had a large pool of problems and the students would be assigned different problems tailored based on their studying progress. Our end-semester feedback showed that our students appreciated this new teaching technology because it provided them with a unique study experience and encouraged them to get the most out of the class.

2. **Connecting theory to practice.** When teaching *Engineering Drawing*, my father often made use of carrots and potatoes to serve as 3D models when the models were in short supply. This left me with a deep appreciation for the importance of practical, relatable examples. As part of the Masterprobo experience, I designed problems for the chapter on "probability for machine learning". Machine learning and inference models involve a lot of abstract concepts, and some students frequently complained and appeared lost. I found that using real-world examples really helped. For instance, to explain p-values for hypothesis testing, I developed more than two dozen problems adapted from recent research results and news items, e.g., how the FDA uses p-values in approving new treatments. I found it rewarding when the same struggling students thanked me for gaining a better grasp of the principles and more confidence in solving real-world problems.

3. **Project-based learning in advanced topics.** Teaching graduate-level courses should be different from teaching undergraduate basic courses, since the audience is a smaller group of students focused on certain topics. When teaching ECE 584 on verification algorithms, proofs, and tools, I created mini-projects using the verification tools I developed, and encouraged the students to build their own course projects based on these tools. I was amazed by the variety of problems on which formal verification can be applied, such as analysis of interesting properties of metabolic processes and planning of safe actions in reinforcement learning, which, in turn, broadened my own view of formal methods. For graduate-level autonomous system and CPS courses, such project-oriented design can enable collaborative teaching, which can augment research for both the students and the instructor.

4. **From the novice-director model to the colleague-consultant model in mentoring.** One way of collaborating with junior students is to identify their capabilities and assign modules of a research project to them based on their skill sets. However, I am more interested in building a sustainable relationship by guiding students to conduct independent research. I tend to agree with the view expressed in [1], that an ideal mentee-mentor relationship evolves from a novice-director relationship at the beginning, to an apprentice-master relationship, to a collaborator-guide relationship that can last a long time, and finally to a colleague-consultant relationship.

   I have had wonderful mentoring experiences with my junior collaborators following that pattern. As an example, I will mention my collaboration with Bolun Qi who started as an undergraduate research assistant and continued as a Master's student in our group. Initially, he took part in developing formal verification tools (C2E2 and DryVR) with me. However, his implementations often did not meet my expectations, since he always circumvented bugs by adding more "if-else" conditions. We later figured out that I had not clearly conveyed the basic principles of each function in the software. Then I changed our discussions from task assignments to mini-lectures to make sure he understood the algorithms and what we were expecting before he started to program. He soon became very passionate and a quick learner. By the time he graduated, we had published four research papers together in top conferences like CAV and HSCC, and he had independently developed an interactive version of DryVR for his Master's thesis! I could clearly observe Bolun's growth and that of other undergraduate students I worked with, and I benefited a lot by developing our mentee-mentor relationships through our collaborations.

Looking ahead: I am excited about continuing my journey as an instructor and mentor to disseminate knowledge and supervise younger generations engaged in autonomous systems and CPS. I will strive to create an effective teaching environment by using the above teaching and mentoring approaches, and I look forward to learning new ones as I refine myself as an educator.

References

[1] Renata A. Revelo and Michael C. Loui. A Developmental Model of Research Mentoring. *College Teaching*, 64(3):119–129, 7 2016.