

I first began teaching as a 13-year-old black belt in karate, and within a year's time, I was teaching students from 5 to 55 years old. Since then, I have been a teaching assistant nine times as an undergraduate and three times as a graduate student. I have taught a class on my own, and I have created content for an online course. Helping others succeed through teaching has always been my passion, and I was drawn to ubiquitous computing (ubicomp) and human-computer interaction (HCI) because of it. I believe these fields are perfect for engaging people of all ages in computer science because they deal with relatable problems and applications.

Pedagogical Philosophy and Lab Culture

My pedagogical philosophy is derived from advice I received from my undergraduate research advisor, Romit Roy Choudhury. He told me, "Try to be around as often as you can, even if you aren't doing work related to research. Just by being around, you can contribute ideas, learn something new, or even help someone with a project that excites you more than what you are doing. You never know what can happen if you aren't around." I have learned throughout my career that success can come from many different opportunities, both planned and unplanned. As such, **my goal is to provide advisees and students with as many opportunities as possible for them to succeed.**

I recognize that progress can be made hourly, daily, or even weekly. As a research advisor, I want to complement my advisees' work rates as best as possible. I believe micro-management through explicit daily meetings can be burdensome when nothing has changed from day-to-day. On the other hand, I do not want advisees to be blocked waiting for feedback. My advising approach will be structured around frequent casual conversations and a welcoming lab culture where my advisees feel comfortable helping one another. Even when conversations are unrelated to research, I want my advisees to know they can reach me at their convenience. Of course, there will be cases when an advisee needs a nudge forward, so my advising approach will also include weekly one-on-ones and bi-weekly group meetings.

Teaching

During my first year as a graduate student, I served as a teaching assistant for CSE 331: Software Design and Implementation for undergraduates. My job was to deliver recitation lectures that summarized what had been taught in class and to cover new material related to the course assignments. When the content I was expected to deliver did not fill the entire time slot, I gave students the option to either leave or stay and ask questions about whatever topic they chose. The teaching evaluations I received from students often cited this practice with positive feedback. Some took the chance to ask for extra help on assignments, while others asked about how to get involved with research as an undergraduate.

While on internship at Microsoft Research in the spring of 2018, I taught EE590A: Advanced Topics in Ubiquitous Computing for the Electrical Engineering Professional Master's Program ([link](#)). The course introduced students to the diverse challenges associated with ubicomp and the skills that can be used to tackle them. Although the course had been offered in previous years, I streamlined the assignments so that they culminated into a final project that combined all of the skills the students learned: a wearable gesture recognizer that collected accelerometer data from a Circuit Playground board and classified the gestures on an Android smartphone. The project required 3D modeling, wireless communication, and programming for both Android and Arduino. The project also incorporated machine learning, a topic that was not included in previous course iterations. The students were allowed to select their own features and gestures, which led them to discover the tension between gesture expressivity and feature complexity on their own. I also gave students complete freedom in how they held their gesture recognizer. Some students went as far as creating wand systems with more complex gestures than I anticipated. Because the course was offered without prerequisites, students had various experience with programming and prototyping; this helped me learn how to encourage creativity while maintaining fairness across students.



An example of a wearable gesture recognizer created by a student in EE590A.

Mentorship

A strong lab culture provides a support structure for some of the most challenging times in a student's life. I have constantly upheld our lab's culture by making myself available to others as they brainstorm ideas, give practice talks, and

write their papers. I have read and edited the majority of the papers submitted by my lab and even gone as far as writing entire paper sections despite not being an author. I will continue to make these commitments for my future advisees.

I have also provided mentorship for both graduate and undergraduate students. I have acted as a proxy advisor for a graduate student on a project with the Bill and Melinda Gates Foundation, ensuring that he progresses his research while fulfilling the Gates Foundation's engineering expectations. The four undergraduates and three high school students I have mentored have made valuable contributions to various publications and study deployments. One student who particularly benefitted from my pedagogical philosophy had just finished another project in our lab and wanted hands-on experience with machine learning, but there were not any relevant projects at the time. I encouraged him to take an online course on machine learning and stay with the lab until a new opportunity arose. Eventually, he joined my PupilScreen project and developed a codebase that allowed me to quickly prototype different neural network structures. Both he and another student made sufficient contributions to warrant co-authorship on my IMWUT 2017 publications.

Outreach

An important part of mentorship is ensuring that students are equipped with the right resources to succeed. I served as a graduate student coordinator for the School of Computer Science & Engineering in 2016, which entailed serving as a liaison between faculty and graduate students and ensuring that the other departmental roles were filled. Starting that year, I also took it upon myself to run our School's annual workshop on the NSF Graduate Research Fellowship.

Part of the reason why HCI is so special at the University of Washington is DUB, the cross-departmental organization that brings together people who are interested in HCI. Being co-advised by faculty across departments, I feel that it is important to create opportunities for interaction. I served as a DUB seminar graduate student coordinator in 2017, but perhaps my biggest contribution to the DUB community was my idea of a DUB-specific doctoral colloquium (DUB DC, [link](#)). The DUB DC provides local students with the chance to get feedback on their research from academic and industrial panelists without cost or travel. I helped organize the first DUB DC in 2017, and it has been an annual event since.

Being a part of the greater HCI community has been a beneficial experience for me, but I did not even know that HCI existed as a field until I participated in a research program during my senior year as an undergraduate. I want to provide people with more accessible ways of learning about HCI and ubicomp. This is one of the many reasons for why I have led our lab's quarterly participation in the DawgBytes and Discovery Days programs—the University of Washington's K-12 CS and STEM outreach programs. As one of our lab's demonstration coordinators, I have also led presentations for policy makers like Senator Maria Cantwell and General Kevin Chilton, companies like Google and Intel, K-12 teachers, and countless other individuals. Finally, I have taught many lectures about ubicomp to students in prospective STEM major courses. These activities demonstrate my willingness to engage with the broader community.

Future Teaching Plans

My teaching experience has prepared me to lead a variety of undergraduate, graduate, and non-major courses. I would easily be able to offer my Advanced Topics in Ubiquitous Computing course again with updates based on student feedback from the first iteration. The course was designed without any prerequisites, so the content can be catered to a variety of audiences. For example, alternate versions of the course could require students to use more advance machine learning techniques or evaluate their smartphone gesture recognizer in a user study.

Gregory Abowd, a leading researcher in ubicomp, posits that ubicomp is disappearing because it has become interwoven with many other disciplines. In light of that belief, **I envision a number of courses that bridge the gap between computer science and practical applications within ubicomp.** For example, I could offer a project-based course that requires students to apply computer vision to image data provided by local clinicians.

More broadly, I feel comfortable teaching introductory programming, mobile phone programming, or computer vision. Regardless of the course I teach, I plan to design the syllabus with the same pedagogical philosophy I have used in the past. I will use fixed assignments so that students can practice the fundamental skills that they have learned, and then I will use open-ended assignments so that they can learn when and how to apply those skills.