I have taught students in various environments. In my teaching, I strive to establish a structured trajectory to help students take ownership of their learning. Students do so by developing their own interests in the course and creating artifacts collaboratively.

Teaching Experience

I have served as a graduate student instructor (GSI) five times during my doctoral study, for the following courses: EECS 493 *User Interface Development* (3 times), EECS 487 *Interactive Computer Graphics*, and EECS 498 *Intelligent Interactive Systems*. The scale of these courses ranges from 30+ to 110+ students. I received a letter of appreciation from the department for achieving exceptionally positive feedback from students (4.75/5) in EECS 487 (40 students). I also participated in redesigning all of the exams and homework assignments for a senior-level UI development course (EECS 493) when a new faculty member began teaching it for the first time.

Outreach Experience

I have volunteered for any opportunity to reach out to K–12 students in the local community. For example, I created and led a workshop that the POSSE Foundation hosted for a group of underrepresented high school students. In these workshops, I taught programming concepts through algorithmic music composition. Students were eager to learn the concept of programming and think computationally to be able to write a musical piece of their own. It was a great moment for me when the workshop participants left a thank-you letter in my mailbox, confessing that they "hated code" until they took the workshop, but were excited to write code more to publish their music online. Currently, I am participating as a mentor in the Wolverine Pathway program, which provides learning experiences and allows underrepresented minority high-school students from the greater Detroit area to explore and contribute to ongoing research projects.

Mentoring Experience

I have been mentoring students and leading collaborative research projects throughout my academic career. I have mentored a total of six undergraduates and two master students at the University of Michigan. Half of them are male and the other half are female students. All of them co-authored and published papers in top-tier conferences in the field (CHI, UIST, and NIME).

Accessibility Experience

Beginning in the first year of my doctoral studies, I worked with a visually impaired faculty member for five years as a teaching assistant for an *Acoustics and Psychoacoustics* course. Through that experience, I have shared the common challenges that this faculty member faces everyday: grading answers to written exams, inserting auditory cues that indicate video replay in the slides, struggling to enter scores into a newly developed course website that lacks alternative text (alt-text) with my eyes closed. This valuable experience helped me to better empathize with this group. I also worked as a tutor at the Office of Accessible Education at Stanford to teach programming to students who need special accommodations.

Collaborative, Live Programming in Music for STEAM Education

With growing interests in making programming education inclusive of diverse populations, I have been interested in leveraging music to teach programming and facilitate these values. Music is a universal language for all age groups and it can attract various populations. My research interest in computer music enables me to develop unique pedagogical practices, facilitating an interdisciplinary mode of STEM + Art (STEAM) education. In particular, the use of programming languages in collaborative, live

music making can cultivate the participation of underrepresented groups in computer science education. I have explored this idea through a series of workshops as stated above, teaching high-schoolers algorithmic composition. I believe this will be an effective means of teaching computational thinking with musical motivations. I suggest developing a course and a programming environment that can be used to reach out to both university and K-12 students.

Teaching Philosophy and Proposed Courses

I have been inspired by my students and the enthusiasm they show when I help them take initiative in further exploration of their creations. To that end, in my teaching environment, I emphasize collaboratively creating one's own artifacts as part of the class. These artifacts can then be further developed as business ideas, research projects, or open-source software. The process of collaborative creation with hands-on technical experience will help students solve real-world problems when they pursue future careers either in the industry or in academia.

I am excited by the potential opportunities to teach any of the introductory-level courses in computer science and engineering. I can also teach various topics in human-computer interaction both in undergraduate and graduate levels. Furthermore, my background in human-computer interaction and computer music will immediately allow me to cover the following courses:

User Interface Development (Undergraduate, Graduate)

This course covers concepts and techniques for designing user interfaces in computational systems. The course lays the technical foundation of user interface development and teaches the methods required in the design process, such as task analysis, interaction design, and usability evaluation.

Web Application Programming (Undergraduate, Graduate)

Today's web applications are increasingly replacing native applications due to their many advantages: cloud storage, their cross-platform nature, and the abundance of third-party libraries that are readily available. In this course, students will learn how to build interactive and dynamic web applications using modern front-end techniques (jQuery, AJAX calls, CSS/bootstrap), back-end programming concepts and methods (PHP, MySQL), and interactive components (WebGL, Web Audio, D3.js).

Human Computation and Crowdsourcing (Undergraduate, Graduate)

This course will cover topics and research in the field of human computation and crowdsourcing. Students will learn the implications of crowdsourcing and obtain hands-on experience through various applications of human computation.

Computer-Mediated Collaboration (Graduate)

This course introduces the technological and theoretical foundations of designing interactive systems that can facilitate collaboration and cultivate participatory culture in scale through the use of computational systems. Using advanced full-stack web frameworks (Node.js, Angular.js), students will be guided as they develop web applications that effectively mediate collaboration among users.

Programming for Computational Arts (Graduate)

This course exposes students to the computational arts. It allows students with backgrounds in various disciplines (e.g. computer science, design, art, music) to get programming experience in the areas of generative art, creative coding, and physical computing. The students will create two different kinds of artworks (interactive installations and live performances) using text-based programming environments such as Processing, Supercollider, and Arduino.