Teaching and Mentoring

serena.l.booth@gmail.com
slbooth.com

I started contributing to computer science education in 2014. In the intervening decade, I have TA'ed classes from theory of computation, to robotics, to computing history; designed lectures, labs, and homeworks for a large 700-student machine learning class and two 150-student HCI classes; and taught small seminar classes. I have significant experience teaching and designing materials on responsible computing practices in different subdomains of computer science, and I view teaching students about the social consequences of their technical work as a great education obligation of our time.

Contributing to a Nationwide Course Review

The National Center for Women and Information Technology (NCWIT) researches best practices for retaining women in computing; with Google, they have distilled these practices into a suite of actionable recommendations and course materials (https://www.engage-csedu.org). As a sophomore, I scored syllabi and materials against these recommendations from hundreds of CS courses at two- and four-year undergraduate colleges across the nation. Some recommendations require overhauling courses but many are simple: Release homework scores with the mean and standard deviation to allay women's misbeliefs in their performance; avoid using stereotypes in lectures and problem sets; frame problems in student interests and connect them to practice. The perspective I gained from this experience helps me to use engagement best practices, peer programming, project-based learning, and flipped classroom techniques in my teaching.

Development of Responsible Computing at MIT

I have contributed extensively to the design of the responsible computing curriculum for three classes at MIT: the *Introduction to Machine Learning* course taken by 700 students each semester, and the 150-student *Software Studio* and *Software Construction* HCI courses. My materials help students reason about the fairness, bias, and societal implications of designing and deploying AI or other software. Some of my teaching materials are publicly available through MIT's OpenCourseware (here and here). OpenCourseware also interviewed Prof. Daniel Jackson and me about ethical decisions in software design; that podcast has reached ≈40,000 listens (here).

Introduction to Machine Learning is a behemoth class led by Prof. Leslie Kaelbling. As students might treat it as a turn-key class for scoring a job, creating engaging material that is tightly coupled to the technical content of the class is critical to incite an awareness of the social consequences of technical decisions. In particular, I designed the ethics component of the weekly mandatory in-person labs. For example, in the first lab, students learn about forming hypotheses with data. I use Simpson's Paradox to explain aggregation bias. Students must read and explain a scatterplot with real data that describes a surprising *negative* correlation between blood pressure and age. Then, students discover that this data is aggregated from both Vermont, which skews old and healthy, and Mississippi, which skews young and unhealthy. In a late-course lab on reinforcement learning, students must discover that a seemingly benign but incorrect reward specification leads an autonomous car to prefer crashing than not advancing. I once overheard a student say to her friend, "The ethics questions in Machine Learning make me feel like I belong in computing."

Software Studio (Prof. Daniel Jackson and Prof. Arvind Satyanarayan) and *Software Construction* (Prof. Robert Miller) center on how humans interact with software, so the average student is typically invested in responsible computing. As such, my approach is more direct. In *Software Studio*, students build a clone of X/Twitter iteratively throughout their weekly assignments. Each week, I designed ethics questions like "who are stakeholders who are not your immediate users?" and dilemmas like "how would you adapt your design to increase engagement among retirees? Among children?" Importantly, I guest lectured to introduce the Ethics Protocol [1]: a process developed by Abby Jaques and Prof. Milo Phillips-Brown to support reasoning about the social consequences of design decisions (similar to the protocol used by Cathy O'Neil in algorithmic auditing [2]). For their final project, students are asked to proactively conduct a full Ethics Protocol analysis, such that they avoid design decisions that retroactively they would have been ashamed to have made.

Classroom Experience

I have taught *Experiential Ethics*: a new standalone course at MIT that teaches ethics as a skill. After being taught the Ethics Protocol, studying codes of conduct for physical engineers, and reading and discussing authors like Profs. Safiya Noble and Ruha Benjamin, students use their new skills in a project to analyze a substantive piece of their own work, e.g., for a robotics student, the implications for a robot tutor for children. I have taught three iterations: one in Summer 2020 and two in Summer 2021. One of my students who went on to PhD study in machine learning at Cambridge emailed me: "Thanks again for having been such an awesome Experiential Ethics Teaching Fellow—your discussion sessions really shaped the trajectory of my research!"

As a teaching assistant in Prof. Harry Lewis' *Great Ideas in Computer Science*, a course described by students as "a finishing school for computer science," I helped students to read influential papers from the history of CS (mostly pre-1995) and assess how these ideas have shaped the field. I am eager to draw on this approach: Especially in AI and Machine Learning, students often incorrectly view the relevance of the field as constrained to the past few years; but learning about the Mark I Perceptron, Eliza, and early reinforcement learning work like TD-Gammon will challenge their interpretations of the meaning of intelligence (and encourage them to reflect on the enduring hubris of AI researchers).

I was also an undergraduate teaching assistant for three classes at Harvard: *Introduction to Theoretical Computer Science* (Prof. Harry Lewis), *Introduction to Computer Science* for non-majors (CS1), and *Autonomous Robot Systems* (Prof. Radhika Nagpal). I extensively contributed to *Autonomous Robot Systems*: I redesigned the course materials to use TurtleBot robots instead of E-Puck robots. While E-Pucks are a table-top platform, TurtleBots are 3 ft tall, so I adapted each assignment to consider this larger scope, e.g., navigating across the CS building. For the final project, I designed a robot party wherein each student team was assigned a role for their robot to play—host, waiter, bartender—and the larger Harvard community was invited to ingest and imbibe.

Mentoring

I have mentored seven undergraduate students at MIT and two more externally, at UT Austin and USC. These efforts have led to two accepted papers (at AAAI 2024), four accepted workshop papers, and one full paper currently in review. One student has started a PhD and three other students are applying for PhD study this cycle. I highlight two students below.

Yiming Zheng (MIT) was my mentee from 2020–2023, starting as a junior undergraduate. I helped Yiming to study the evaluation of interpretability methods for machine learning. Yiming published a first-author workshop paper at NAACL as a senior and went on to write his master's thesis with me. Tiffany Horter (Wellesley; MIT cross-registrant) has been my mentee since her sophomore year. I have helped her to study human concept learning for interpreting robot behaviors. She is writing her senior thesis on this topic, and she has first-authored a workshop publication at HRI. I am eager to help her grow into an HRI academic in the years to come; hopefully she is accepted to graduate school at the same place where I hopefully land a job!

Teaching Plans

Beyond introductory computer science, I can teach upper-level undergraduate classes on AI, reinforcement learning, machine learning, and robotics. For graduate students, I can teach classes on Human-AI Interaction (focusing on both machine learning and reinforcement learning, and inspired by Prof. Matt Taylor's syllabus [3]), Human-Robot Interaction (inspired by Prof. Anca Dragan's syllabus [4]), and Explainable AI (inspired by Prof. Hima Lakkaraju's syllabus [5]). I also wish to bring Experiential Ethics to my future institution. Finally, one of my passions is computing and AI policy; I wish to teach a class informed by my work in the U.S. Senate. My framing is inspired by Prof. Daniel Weitzner's course at MIT, which is co-taught at Georgetown Law, where technology students argue Internet privacy cases within a simulated courtroom. In my context, students representing bodies of government like Congress and the agencies (NIST, FTC, State) would negotiate the design of guardrails for new technologies.

References

[1] Ethics protocol lecture. https://ocw.mit.edu/courses/res-tll-008-social-and-ethical-responsibilities-of-computing-serc/resources/mitrestll-008f21-6170lec-1/. Accessed: 2023-11.

[2] Cathy O'Neil and Hanna Gunn. Near-term artificial intelligence and the ethical matrix. *Ethics of Artificial Intelligence*, pages 235–69, 2020.

[3] Interactive machine learning. https://sites.google.com/ualberta.ca/cmput656/home. Accessed: 2023-11.

[4] Algorithmic human-robot interaction. https://people.eecs.berkeley.edu/~anca/AHRI.html. Accessed: 2023-11.

[5] Explainable artificial intelligence. https://interpretable-ml-class.github.io/. Accessed: 2023-11.