# Teaching Statement

Tom Silver

My passion for teaching is one of the main reasons that I left research in industry to return to academia. Some of my proudest experiences have been helping students in small ways ( “aha!” moments in office hours) and larger ways (multi-year mentorships). My teaching and mentorship philosophy is informed by my background in the following areas:

- **Course development.** I co-developed the syllabi for two new MIT courses (6.882: Structured Models for AI; and 6.s058 / 16.420: Representation, Inference and Reasoning in AI) and assisted in the creation of problem sets, projects, and exams. The latter course is now part of the new MIT undergraduate degree in AI and Decision Making (Course 6-4).

- **Teaching experience.** I delivered lectures at MIT as a co-instructor for 6.s058 / 16.420 and TA for 6.882. In college at Harvard, I served as a TA four times for theory of computation and discrete math classes. I taught weekly one-hour sections, held office hours, developed and graded assignments, and assisted in a flipped classroom. I earned four teaching awards (Harvard Derek Bok Certificate of Distinction in Teaching 3x, MIT EECS Hazen Teaching Award) based on student evaluations.

- **Mentorship experience.** I have supervised 13 students at the PhD, Master’s, undergraduate, and high school levels. I have worked with the majority of them for multiple years and usually have between five and seven active at a time. In addition to weekly one-on-one meetings, I organize “mini-group meetings” where we rotate between research updates, reading groups, brainstorming sessions, and social activities. I have co-authored papers with eight of the students (five conference, three workshop).

- **Outreach.** In college, I founded a large computer science outreach program called the Digital Literacy Project (DLP). We developed and taught an introductory programming course in under-resourced Boston schools. Over three years, my team and I created a 10-week curriculum, wrote and printed an original 67-page workbook for every student, organized a field trip for 120 students to a Harvard computer science fair, raised more than $20,000, recruited 50 teaching volunteers, and reached more than 800 middle school and high school students. We also traveled to Rosario, Argentina to train volunteers at a local nonprofit so that they could start a similar program.

- **Managing staff.** In addition to managing the staff of DLP, I led a teaching staff as Head Teaching Fellow for two introductory CS courses at Harvard (CS20: Discrete Math; and CS121: Theory of Computation). The courses had 60 students, 8 teaching fellows and 160 students, 10 teaching fellows respectively.

## Teaching and Mentorship Prerequisites

In my experience, effective teaching and mentorship starts with three prerequisites.

**Respect and compassion for students.** Student life comes with a variety of challenges, many of which are not readily visible to instructors and advisers. I believe in giving students the benefit of the doubt when they appear to be struggling and reaching out to offer support when appropriate. Even for students who are thriving, respect and compassion are crucial. I want all my students to feel that we are collaborating to increase collective knowledge and understanding.

**Organization and reliability.** As a researcher, I understand the technical challenges of partial observability and stochasticity. As a teacher and mentor, I try to mitigate these challenges for students as much as

possible. Whether I'm designing problem set questions or full research projects, I choose my words carefully to remove any uncertainty about my intentions. When working with students, I take pride in my punctuality, availability, personal organization, and communicativeness, and make every effort to clarify expectations.

**Responsiveness to feedback.** Power dynamics can make it difficult for students to volunteer critical feedback, so I try to directly invite feedback on my teaching and mentorship. In both running courses and advising research, I design online surveys and meet with students afterwards to debrief. These information-gathering exercises are invaluable in several ways. Sometimes I learn that the pace of our course or research project should be increased or decreased. Other times, I am surprised to find that a student perceives themself to be faltering, which gives me an opportunity to reassure them.

## Teaching and Learning with Abstractions

The primary goal of my teaching is for students to develop a coherent intellectual framework—to understand the material at multiple levels of abstraction. I tend to emphasize the higher levels of abstraction, while helping students develop skills to acquire the lower levels on their own.

**High-level abstraction learning: cumulative content, spiral review.** Even for advanced subjects, I believe it is very useful for course content to be cumulative: later lessons should build on earlier ones. Such accumulation is crucial for developing high-level understanding. The main risk in cumulative content is that some students may fall behind. Spiral review, where earlier material is periodically raised again, is a useful technique for combatting this risk and generally reinforcing concepts. I believe spiral review and other forms of interleaved practice are under-utilized at the college and graduate levels.

**Low-level abstraction learning: practice, exploration.** In computer science and math-adjacent subjects, there is no pedagogical substitute for hands-on experience. Problem sets and projects are the traditional modes and they are certainly valuable. I have also seen the benefits of a flipped classroom approach in my experience as a TA for Harvard's CS20 (Discrete Math). In CS20, students complete readings and watch short videos before class, then spend the class working together on exercises, with TAs standing by to assist. For larger and more advanced classes, long-term final projects are an excellent way to encourage students to practice and explore on their own. As a supervisor of long-term projects, I would work closely with students to make sure that their projects are feasible and well-aligned with the course's content objectives.

**Inclusive teaching and course design.** There is no one-size-fits-all strategy for education—every student, group of students, and course subject is different. When preparing to teach, I start by identifying where students are coming from. For example, in MIT 6.s058 / 16.420, we quickly realized that the class had two main populations: advanced undergraduate EECS students, and graduate Aero/Astro students. The former were more comfortable with programming and the latter were more familiar with probabilistic modeling. We adjusted the curriculum to make the material accessible to both groups. As a junior faculty, I would talk to my colleagues and students themselves to get initial context. Over time, I would develop courses that are not only well-aligned with typical enrollees, but also flexible enough that I can adjust on-the-fly.

## Future Teaching

Given my background, I would be very excited to teach (1) introductory courses in AI, machine learning, robotics, and discrete math for computer science; (2) advanced courses in decision-making for AI, robotics, and reinforcement learning; and (3) new courses that I develop on task and motion planning and neuro-symbolic program synthesis.