I approach teaching via interactive storytelling. Computer science is abstract and complex; translating the concepts we teach and study into a narrative makes them easier to reason about and understand. This is not just for students: my research process is often driven by narrative simplifications of the literature. I want to teach, at all levels, to hone cognitive empathy for researchers and students alike. I seek to create an academic environment where *everyone* is enabled to build their own meaning on the foundations of theoretical computer science (TCS).

Experience

I have experience both in the classroom and in mentoring student research. During my six years as a PhD student at UCSD, I served as the Teaching Assistant for six courses. I deliberately "taught through" the entire undergraduate TCS curriculum so I could tell a coherent story linking each course. While working on applied machine learning for Mars Curiosity with Professor Dyar at Mount Holyoke, I provided weekly research and coding support for undergraduate students writing theses.

I was evaluated as "Overall: Excellent" (highest possible rating) TA by each instructor I worked with at UCSD. When I was the TA for "Introduction to Graduate Complexity Theory," 85.7% of responding students (55% response rate) "Strongly Agreed" that they would recommend me to other students as a TA. I am invested in continuing to improve as an instructor: I participated in an extra-curricular "Engaged Teaching and Learning" seminar at UCSD, a once-weekly course that trains attendees in evidence-based teaching practices.

I served as a Mentor TA for three sessions of the UCSD TA training class. This consisted of running weekly practica to develop teaching skills. Activities I ran included sample five-minute lectures, discussions of how to grade fairly and efficiently (with concrete examples), white-boarding practice, and office hours role-play. Additionally, I served on the final panel for this TA training, answering questions related to grading, work-life balance, difficult experiences with students and professors, and other topics of concern for new TAs and grad students.

I facilitated many reading groups and seminars on topics such as: complexity lower bounds, boosting algorithms, generalization in computational learning, and minimum circuit size problems. Usually I will prepare the first few sessions, along with a list of suggested papers and topics for future meetings. Nearly all of my collaborative research projects have emerged from these groups, where I "teach to learn."

Philosophy

Narrative Pedagogy I want to empower students to tell their own stories using TCS. If they can meaningfully "cast" formal objects into concrete roles, then I have succeeded as an educator. Appropriate casting allows them to both (1) solve problems with algorithmic tools and (2) leverage intuition from other domains for theoretical understanding.

Students who learn to develop and criticize correspondences between formal objects and reality become *authors* of mathematics. They can see that TCS is a lively and interactive set of *literary communities*, each with its own conventions, norms, and aesthetics. At every level, students should be invited to become active participants in these communities. Thus, I model metaphor-production in the classroom and invite students to engage, via examples and questioning. This construction of facility with TCS as *literacy* is the foundation of my teaching practice.

Inclusive Spaces I am committed to ensuring that *all* students can access departmental resources, regardless of their background. My spouse has epilepsy, a chronic seizure condition. Because I have worked with her to manage this condition and secure accommodations, I am aware of and sensitive to the needs of students who require accommodations for any disability, invisible or otherwise. I often volunteer to proctor Office of Disability Services exams to ensure students are

treated respectfully and professionally. I solicit requests for accommodations at every contact by reminding students that they can reach me at office hours, over email, or at the end of discussion to discuss "any issues they are having with the course."

Careful attention to language is as important for my teaching as it is for my research. I was raised in a bilingual household. This helps me recognize the difference between misunderstanding concepts and misunderstanding English. I use "plain" English that is more easily understood by an international audience with *all* students, and encourage other course staff to do the same.

Language barriers and ability status are just two factors that we must consider to make *all* students feel welcome in computer science departments. Academic and industrial computer science has systemically excluded and marginalized women, people of color, and queer people. This while I have been welcomed, presumed competent, and accorded many other privileges simply because of my race, background, and gender. So, my responsibility is to be an effective ally for students and colleagues from under-represented groups. Our literature and community will be richer and more effective for including and respecting their voices.

Courses

Undergraduate Courses. I am best suited to teach courses in Theoretical Computer Science, Data Science, and Machine Learning. I prefer to "rotate" through the curriculum, to help ensure that these courses tell a coherent story for the students. For example, I treat the introductory **discrete mathematics** courses as intensive language instruction: students are learning the *formal languages* they need to describe and reason about computer science. So I would embed specific examples from later courses in **databases** and **machine learning** to help students practice their translation skills.

My advanced courses would focus on facilitating engagement with research, and students' decisions of whether to apply to graduate school. I benefited enormously from supervised undergraduate research, so in these courses I would provide the same extensive support I was afforded: assisted reading, approachable preconceived projects, and workflow guidance.

Graduate Courses. The key difference between graduate and undergraduate education is the immediacy of the research frontier. Undergraduates need to learn that the research frontier *exists* and have some exposure to research activity. Graduate students must be taken to the *very edge* of what is known, and given the tools to push forward. My graduate courses will include readings from the research literature, and develop the kinds of writing skills required to produce original research. Luckily, TCS has a broad selection of "classic" papers that are self-contained and well-written.

Theory also has the "advantage" that many open problems are easy to state but difficult to solve. In all my advanced classes, I would close key sessions with a "tractable" open problem related to the day's topic — problems that are not famous, but still important and interesting. The ability to identify such problems is a crucial research skill that I can demonstrate in a classroom environment, and explicit meta-cognition around their selection can help students get started in their own research.

Research & Experimentation

Improved Grading via Adaptive Rubrics For large UCSD classes, I still gave rich feedback on assignments. I always graded with a two-pass system: (1) Identify key features of each student's response (steps, definitions, mistakes, etc). (2) Develop a detailed rubric from the most common features. (3) Apply the rubric to each assignment. Crucially, the rubric is a *function* of student responses; it is defined by the first pass. I saw each assignment twice and this helped ensure consistency. I used a histogram of response-features to decide which topics to cover in future discussions. Even in smaller classrooms, this rigorous method can improve *justification* for grades.