My teaching philosophy centers on two goals for my students: a rigorous theoretical understanding and the ability to implement and apply the material to real-world problems.

In my experiences as both teacher and student, I have found that the correct balance between theory and application is crucial for learning in any discipline, but particularly important for computational sciences. Each of the two perspective heightens the other, bringing students to a more complete understanding of the material and greater proficiency in application. These aims permeate all pedagogical tasks, including designing curricula and assignments, guiding classroom experiences, and interacting with individual students.

Classroom Dynamics My first undergraduate-level teaching experience was as an Assistant Instructor for *General Computer Science* (COS126) at Princeton University. During my discussion sections each week, I reviewed material from lecture with the students. The breadth and pace of the course was challenging—students asked confused questions that indicated that they had missed fundamental principles. In order to master the material, my students needed a more substantial theoretical background. To provide this background, I regularly re-explained material covered in lecture to reinforce the concepts. This process was interactive; sometimes I simply answered the question or connected new material to previous lessons, but I also encouraged students to answer with their best guess and build as a class to the correct answer—incorrect answers can often be harnessed as learning experiences.

With some theoretical knowledge in place, we then worked through practice problems, first together and then individually or in small groups. As the students worked independently, I circulated through the class and unpacked questions with those that struggled. Through this process, my students became comfortable with the material and learned to tackle new problems. They felt prepared for assignments and exams only with both theoretical knowledge and practice in application.

More recently, I have had the opportunity to guest lecture for other courses at Princeton, *Fundamentals of Machine Learning* (COS424, renamed) and *Foundations of Probabilistic Modeling* (COS513). Multiple students provided feedback that I "engaged with the class" and that my teaching style was "very interactive."

Moving forward, I hope to maintain an interactive atmosphere in my classrooms, no matter the size. While this can be challenging in a larger lecture environment, there are several strategies in structuring the material to encourage engagement. For example, the lecture narrative can be framed with questions or checks that prompt students to asses their current understanding of the material; this approach can similarly be used when presenting research. Additionally, I plan to use live classroom polls and "think-pair-share" activities in both small and large classroom environments. I hope to build up students' confidence in speaking up, especially in larger classrooms, by beginning with exploratory questions with no wrong answer, and transitioning to more difficult questions as the students' comfort increases. More intimate classroom settings allow for a wider range of teaching strategies, from solving a problem as a class (each student taking a turn in a "pass-the-chalk" manner) or diving into groups and having each group explain a concept.

Navigating Diverse Backgrounds As an introductory class, students in COS126 had a variety of skill levels, interests, and prior experience. With diverse groups of students, the challenge is knowing where to focus, as extremes could easily disengage a large portion of the class. One of my greatest challenges in teaching a diverse class was when I taught a free summer SAT course. I decided to design this class when I was asked about the SAT by a number of students that I mentored at church. The class had an extremely wide range of students—performance on the first practice test ranged from the first to 97th percentiles.

In both of these classes, going over problems revealed when students did not understand specific concepts—application exposed a lack of theoretical understanding. The first step in addressing these confusions was to ask if other students wanted to learn or review the relevant concepts; if not, I helped the student after class or pointed to external resources. If the class was receptive, we discussed the concepts and reviewed example problems. Students also worked in groups, either learning from each other, or preparing to show

everyone how to solve a problem. Group work allowed weak students to be helped by strong ones, and strong students deepened their understanding of the material by learning how to teach it. This structure allowed all students to strengthen both their theoretical knowledge and their ability to implement it.

Course Design My second formal academic teaching experience was as an Assistant Instructor for *Interacting with Data* (COS424), also at Princeton University. This course was well-structured in terms of the activities completed by students, but both instructors and students would have benefited from clearer course goals. Some students were distracted by mathematical details in the readings; clear objectives would have focused their study. As an instructor, these same objectives would have guided me in selecting readings and developing assignments. Learning from this, I created course goals when I designed the summer SAT class; these goals helped me filter through an abundance of materials on the subject. In any course I develop or teach, crafting clear course goals will be my first task.

In designing courses, I will also draw on the flipped classroom paradigm, where students engage with instructional content outside the classroom, and focus on application in class. This design allows the instructor to fill gaps in theoretical understanding while the students are engaged with the content.

Mentoring I find teaching and mentoring to be rewarding; it is one of the principal reasons that I am seeking an academic position. The year after I taught COS126, I encountered one of my students who told me that she had chosen to major in computer science and that I had helped influence her decision; this level of personal impact is difficult to find in other professions.

In my new role as a postdoctoral researcher, I have begun to mentor graduate students in the Engelhardt lab. This includes having detailed discussions about research directions, answering specific questions about statistical modeling and its implementation, and giving regular guidance on defining the next steps for particular projects. This last form of mentorship is the most productive, but also the most challenging. As a professor, I will seek to find projects for my students that both push scientific understanding further and are achievable for the respective students, with clear intermediate steps—this is yet another example of the importance of balancing both theory and application. In addition to mentoring graduate and undergraduate students in their research, I look forward to continuing to be involved in mentorship with the Women in Machine Learning organization, with church congregations, and through courses I teach.

Future Teaching I am seeking a faculty appointment at schools which value both research and teaching. Like the balance between theory and application, it is easy for institutions and faculty to focus on one of these concepts to the exclusion of the other. I believe that teaching provides a catalyst for collaboration and greater insights into fundamental concepts, and that collaborative research can similarly improve the skills needed for quality teaching.

I am happy to teach a wide variety of Data Science or Computer Science courses for undergraduates and graduate students. Within my field of expertise, I am particularly qualified to teach or develop courses in machine learning, probabilistic graphical models, text analysis, and computational social science. I would also enjoy teaching various introductory-level data science or computer science courses and guiding students through more independent and open-ended study and research.