Teaching Statement

Justin Thaler

I am excited to embark on a career in which teaching and advising play central roles. I am also committed to exposition in both oral and written form, and regard the effective communication of concepts and ideas as one of the most important and rewarding aspects of a researcher’s job.

Teaching Philosophy. My approach to teaching has been informed by several experiences. As a graduate student, I regularly delivered guest lectures in my advisor’s classes, and was twice a Teaching Fellow, for CSCI E-210 (Algorithms at the End of the Wire) and CS 124 (Algorithms and Data Structures). I learned a lot about teaching through guest lecturing, teaching weekly sections, holding office hours, grading, and interacting with students. More recently, I delivered a 15-hour course on Verifiable Computation at the 14th Bellairs’ Crypto-Workshop that was attended by over a dozen professors and graduate students from universities throughout North America and Europe. This was an intense and rewarding experience that taught me a lot about how to effectively present a large body of material in a limited amount of time, and also taught me about effective expository writing (I compiled lecture notes in advance, and am in the process of turning them into a formal written survey).

I believe that the aspect with the biggest influence on the effectiveness of a course is the clarity with which the material is organized and presented (this holds as well for written surveys, research talks, and papers). A close second is the exercises, as these force students to engage actively with the material and to follow lines of reasoning to their logical conclusion.

Clarity of Exposition. Clarity of exposition requires careful and often time-consuming planning (at least, it does for me). Planning out the precise order in which concepts are explained allows me to ensure that each idea or logical step follows easily from the preceding one, and helps me to anticipate questions and potential sources of confusion.

A crucial pitfall that I have learned to pay careful attention to is background material. Even the best-planned presentation will be rendered ineffective if it assumes familiarity with a concept that students may have actually never seen before, or are not yet comfortable with. This is a surprisingly easy trap to fall into, especially in (common) settings where students’ backgrounds are diverse, and not all class members have seen concepts that are easy to take for granted as a presenter.

Exercises. Effective exercises are by far the most effective way to get students to engage deeply with the material and internalize concepts. There are many kinds of exercises that can accomplish these goals, such as asking students to prove theorems related to or extending concepts covered in class, asking students to apply concepts or techniques on specific examples, or asking students to implement algorithms. The best kind of exercise will depend on the material being covered.

I have personally found programming assignments to be particularly effective at ensuring engagement – there are few better ways to ensure that one understands an algorithmic concept in all of its details than to actually code it up. Programming assignments can also be used to ensure that
students understand an algorithm’s limitations, or to understand challenges that arise in practice. Of course, such assignments are not appropriate for all courses, especially theoretical ones.

The courses for which I have been a teaching assistant have included a mix of programming assignments and proof-based problem sets; I have found this combination to be highly effective and plan to use it in the courses I teach whenever appropriate.

**Advising.** I have sought out both formal and informal opportunities to work with younger graduate students throughout my career so far, and have greatly enjoyed these interactions. For example, in my final year of graduate school I started working with Mark Bun, who was then a first-semester Ph.D. student, and I have worked closely with him ever since. I had the opportunity to introduce Mark to an entire area of research, and convey both big-picture views of areas as well as crucial ideas that are often obscured in the formalism of research papers. Watching Mark grow into a mature and independent researcher has been a pleasure, and I look forward to the important responsibility of playing a similar role in the development of my future students. While at Yahoo Labs, I have also worked closely with Samira Daruki (a Ph.D. student at University of Utah), serving on her qualifying committee, and co-hosted Christopher Musco (a Ph.D. student at MIT) for a successful internship. I am eager for more opportunities to advise students as they embark on their research careers.

**Future Teaching.** As my research spans several areas of theoretical computer science and mathematics, I am qualified to teach a broad array of courses at both the undergraduate and graduate level. A brief summary of these courses follows.

*Algorithms and Data Structures.* My formal teaching experience to date has been in both an introductory algorithms and data structures course, and a graduate research topics course that focuses on algorithms and data structures. I would like to teach similar courses in the future, and am also qualified to teach other related courses such as Randomized Algorithms.

*Data Analytics.* I would also be happy to teach a more applied class on data analytics, covering topics including clustering, regression, classification, essential elements of matrix analysis that are often skipped in introductory linear algebra classes (PCA, SVD, etc.), and techniques for addressing problems found in real data such as errors and missing values. This would be a hands-on course, with the goal of imparting to students the tools necessary to identify structure in data and use it to make predictions.

*Data Streaming and Algorithms for Massive Data Sets.* I am also eager to teach a course covering data streaming algorithms, and algorithms for massive data more generally, such as compressed sensing and property testing. I believe that this is an ideal class to offer undergraduates and beginning graduate students as a follow-up to an introductory algorithms course, as it is a well-motivated and active research area in which many fundamental algorithmic ideas are displayed very cleanly. This is the subject that caused me to get excited about theoretical computer science, and I look forward to sharing this passion with future students.

*Computational Learning Theory or Communication Complexity.* Two of my major research interests are in computational learning theory and communication complexity, and I would greatly enjoy teaching an introductory course for undergraduates and beginning graduate students on either subject, as well as an advanced research topics course.

*Computational Complexity Theory.* I am qualified to teach introductory and advanced courses on computational complexity theory, as well as an advanced graduate research topics course covering
Algebraic Methods in Theoretical Computer Science, which have played a pervasive role in my own research. The latter would be a relatively broad course, covering algebraic methods within the areas of verifiable computation, learning theory, and concrete complexity.

Mathematics. As a consequence of my extensive experience with algebraic methods in my research, I am well positioned to teach courses on Linear Algebra and Matrix Theory, Abstract Algebra, and Approximation Theory. I am also qualified to teach an introductory course on Discrete Mathematics.