The topic of the day's lecture was the topological notion of continuity ($f\colon X \to Y$ is continuous if for each open set V of Y, the set $f^{-1}(V)$ is open in X). However, the students already encountered the ε-δ definition of continuity in calculus ($f\colon \mathbb{R} \to \mathbb{R}$ is continuous at p if, for each $\varepsilon > 0$, there exists a $\delta > 0$ such that $|p - x| < \delta$ implies that $|f(p) - f(x)| < \varepsilon$). So, I wanted to assure the students that these two seemingly different definitions coincided. I began my lecture by asking the students to recall the ε-δ definition. Then, two students began to debate whether the definition was *for each* ε *there exists a* δ or *for each* δ *there exists an* ε. Deviating from my plan for the day, I told two students to come to the board to explain the definition as they understood it. This led to a friendly heated discussion over the definition of continuity. I sat down and watched, prompting occasionally to ask them to draw a picture or to consider an example. As a result, the students discovered the definition themselves.

The above incident is representative of the topology seminar that I led at the Institute for Science and Technology Austria (IST Austria). In this seminar (as well as in other courses that I have taught), I encouraged students to think critically about what I teach, whether it is a hard theorem or a definition. After enough time has passed, students forget most of the lecture material in a class. However, the retention of materials that a student discovered, rather than was told, is much better. For this reason, I strive for an **active learning** classroom. In mathematics and theoretical computer science, an active learning classroom is one that encourages the students to attempt a proof before listening to someone else present it, or to ask questions about transposing the order of *for each* and *there exists* in a definition.

Teaching Experience

When I teach, I involve the students by asking simple questions, by pausing for a minute after I present an important theorem to allow the students to digest the statement, and by allowing the lecture to be guided by student-asked questions. I assign homework that balances understanding definitions with synthesizing knowledge learned throughout the semester. My teaching style emphasizes that the material should be understood, not memorized. I achieve this by promoting learning as an interactive process, and by making myself available for discussions with the students. Moreover, in all classes that I teach, I incorporate skill-building activities such as technical writing and communication.

Below, I briefly describe the experiences that I have had while teaching or TAing various classes, with more topic-specific courses first and lower-level classes last. Each course has helped me to improve my teaching style.

Computational Topology (S-16, S-18). In Spring '16, I taught Computational Topology at MSU as a topics course. In this course, we covered topics ranging from knot theory to Brouwer's fixed point theorem to persistent homology. This course did provided a basis from which another faculty member and I proposed two new courses: M 476 (Introduction to Topology) and CSCI 535 (Computational Topology), which are now offered every other year (Fall odd years and Spring even years, respectively). *Steve Cooper (University of Nebraska - Lincoln) observed a lecture in S16.*

Topics in Topology (Seminar, 2011–12). Recognizing my enthusiasm when speaking about topology, a student at IST Austria asked me to teach him topology, and I initiated a seminar on point set and algebraic topology at IST Austria. We met once or twice a week for 1.5 hours, where I would lecture from Munkres' *Topology.* I had four to eight students at each lecture. The students who attended were excited to learn the material, often asking challenging questions that sparked great discussions (see above).

Computational Geometry (S13). I co-instructed a small undergraduate Computational Geometry class with Gary Miller at Carnegie Mellon University. We broke the semester into two main parts: classical computational geometry (convex hulls, Voronoi diagrams, Delaunay triangulations, arrangements, linear programming, etc.) and modern topics (including the computation of the Fréchet distance, the approximate nearest neighbor search, and an introduction to persistent homology). One of the first lectures I taught was on oriented projective geometry. A graphics student approached me after the class to let me know that he finally understood some of the computations in graphics after that lecture.

Graduate Algorithms (S17). In this class, we covered Kleinberg and Tardos *Algorithm Design*. For the most part, we covered this book from cover to cover, but skipped sections that overlapped with the complexity class. This class had a course project (which involved reading and understanding a research paper, with a presentation and final write-up). I also required the students to present an oral proof in a one-to-one setting. *Beth Burroughs (Department of Mathematical Sciences) observed a lecture in S17.*

Undergraduate Topics in Algorithms (F15,F16,F17). In this class, I teach the students about loop invariants and proofs of correctness and provide plenty of in-class time to work on problems in groups. We also cover topics such as runtime analysis, randomized algorithms, and recurrence relations. I have used two books to teach this class: *Introduction to Algorithms* (CLRS) and *Elements of Programming Interviews*. The former provides a good foundation of algorithms, and the latter keeps the interest of students who are more interested in jobs in industry than continuing to graduate school. At the same time, this course covers interesting algorithms that are aimed to pique the interest of students who might not have otherwise considered graduate school. This class has a semester-long project that requires students to make a short video on a new algorithm.

Discrete Mathematics for Computer Science (S09). I co-instructed this course with Herbert Edelsbrunner at Duke University. We created a set of Lecture Notes that we made publicly available online, and are now used by others teaching discrete mathematics. The class was well-received by the students; in fact, several of the students commented that I was easily accessible outside of class.

An Overview of Computer Science (S08). In preparation for teaching a future course (Discrete Mathematics), I observed teaching techniques in this introductory computer science class. I also acted as the lead TA for this course (with seven undergraduate teaching assistants) and was responsible for overseeing the labs that accompany this class. In addition, I taught two lectures. I dedicated one of these lectures to teaching the students debugging techniques, as I noticed that many students had difficulty tracing through loops and using a debugger.

Calculus (S04–S03). As an undergraduate student, I was a TA and recitation leader for various calculus classes at Saint Joseph's University. In this role, I learned how to redirect a student's goals from homework-solving to concept-learning.

Mentoring

I currently have four Ph.D. students: Robin Belton (mathematics, second year), Sush Majhi (Tulane mathematics, fourth year), Sam Micka (computer science, third year), and Anna Schenfisch (mathematics, first year). In general, I try to meet weekly with my Ph.D. students. In addition to their primary research, I encourage my students to be involved in outreach and to work on improvement of their written and oral communication skills. My graduate students also co-mentor the undergraduate students working with me.

Sam Micka is co-advised with Brendan Mumey (MSU CS). The theoretical / algorithmic research that Sam is working on deals with monitoring network flows; he currently has one conference paper and two conference presentations. At IHEEP 2017, he won second prize in the student presentations. As Sam is interested in teaching at a liberal arts college, he will also be involved in the NSF-funded CS Education project, and is involved with my outreach efforts in promoting CS education in middle schools and high schools throughout the state.

Robin Belton is working on understanding properties of the Erosion distance between persistence diagrams. She will use our theoretical work in the QuBBD project to analyze prostate cancer histology. She is also co-mentoring two undergraduate students who are working on applying techniques from topological data analysis to analyzing music. This work is an enhancement of some results from natural language processing.

Anna Schenfisch is just starting her Ph.D. studies. She started in Summer 2017, and has since become familiar with persistent homology and has investigated several research directions. In particular, we are starting to investigate inference results with directional persistence (filtering a topological space based on height in a given direction). Anna is interested in research related to the Gleason-grading project, and, in that direction, is running some simulations to study how to capture discrete curvature of glands given a set of nuclei locations.

As a post-doc, I started working with Tulane mathematics Ph.D. student Sush Majhi. I am currently co-advising Sush with Carola Wenk (Tulane, CS) and Rafal Komendarczyk (Tulane, mathematics). Sush and I have weekly one-on-one research meetings. Sush is currently working on a paper on manifold reconstruction; in particular, he is working on generalizing a result by Nyogi, Smale, and Weinberger.

Additionally, I have worked closely with graduate students during my three years as a postdoc. Mahmuda Ahmed and I have implemented the LH distance described in my research statement. Fabrizio Lecci, Jisu Kim, and I have developed an R package which implements techniques in statistical topological data analysis.

At the undergraduate level, I have quite a few mentees. To see the full list, please refer to my CV. I highlight six of the projects here:

- Mathematics major Parker Evans (Tulane University) is co-advised by Carola Wenk and myself. Under our guidance, he earned a Goldwater scholarship last year. He is working on extending the results of a former Ph.D. student (improving an exponential-time algorithm to polynomial time!). Parker will be graduating in Spring 2018, with a plan to pursue a Ph.D. in combinatorics.

- McNair Scholar Angus Tomlinson and REU student Kira Wencek (U. Rhode Island) are working with myself, graduate student Robin Belton, and faculty member David Millman, on using topological data analysis to compare musical compositions and to evaluate the complexity of a musician's work through time.

- The NSF-funded Storytelling grant provides amble opportunity for undergraduate research. Currently, architecture major Saurabh Tulsankar and CS-major Jachi Madubuko are working on developing 3D models that can be imported into Alice. Additionally, CS-majors Brendan Kristiansen and Kirby Overmann are working on developing outreach programs and middle school lesson plans that will eventually be made publicly available. Recently, Jachi has received a USP grant and been awarded an MSU Emerging Scholar award.

- Civil engineering major / CS minor Maia Grudzien is working with graduate student Sam Micka and myself on analyzing accident data throughout the state of Montana. She won first prize in student presentations at the IHEEP conference, and was named an MSU Emerging Scholar.

- Multiple undergraduate students have worked on the art-science project, NeuroCave (currently on display at the Holter museum in Helena). McNair scholar Justin O'Dea and CS-major Will Dittman collected data from pairs of people wearing EEG headsets. Additionally, CS-majors Brendan Kristiansen and Carie Pointer have worked with CS-faculty member David Millman on protocols for starting and closing the exhibit each day.

Training

I received the Graduate Aid in Areas of National Need (GAANN) fellowship while a graduate student at Duke. The aim of this fellowship was two-fold: to train me in both research and in teaching. Funded by the GAANN fellowship, I attended the SIGCSE Symposium in 2009 and 2010 (this is one of the two main conferences on computer science education) and the CARA-W Grad Cohort 2009. There, I attended many sessions on teaching techniques and curriculum development. At Duke, I took two teaching classes: Teaching with Technology and Introduction to College Teaching (the latter of which was taught by the Dean for Academic Affairs, Doug James). I also attended various Teaching Ideas workshops, including: Evaluating Critical Thinking: what every instructor needs to know, Using Acting Techniques in the Teaching/Learning Process, Academic Skills and Campus Resources for Duke Undergraduates, Responding Efficiently and Effectively to Student Writing, and Strategies to Teach Large Enrollment Classes Successfully. After attending these workshops, I always tried to test out a new technique in the classroom or in a public lecture. For example, the instructor of the acting skills class suggested having students get up and physically move half-way through the class. In the next Discrete Math lecture, I had the students do jumping jacks; I repeat this exercise in my classes occasionally.

As MSU, I have attended Center for Faculty Excellence (CFE) teaching and advising workshops: *Charting a Course with Learning Outcomes* (Sept. 2017), *Classroom Management Tips for Maintaining a Positive Learning Environment* (Sept. 2016), and *At-risk for University & College Faculty Training* (Sept. 2015). In addition, I have completed the Indigenous Mentoring Program (IMP), which is supported by PNW COSMOS to help faculty advise (and recruit) American Indian and Alaska Native students. These workshops have both provided general information that I bring back to my teaching, as well as have provided me with information regarding resources on campus.

External to MSU, I have also attended the 2016 CRA Career Mentoring workshop (for early-career faculty) and the Big Ideas in Big Sky workshop (focused on

Engineering education).

In January 2017, Connie Chang (CBE) and I attended *Improv for the Spirit*, facilitated by Katie Goodman. At the time I attended this workshop, I was teaching Graduate Algorithms, and I was struggling with the quality of student's writing. The similarities between good academic writing and good improv acting struck me, especially the 'setting the scene' part: let the audience know what you are doing. Sometimes (both in academic writing and in improv acting), you may think that you are stating the obvious; however, what is obvious for you is not always obvious for the audience. Thus, walking the audience through the proof/scene is crucial for the audience to follow the scene / writing. Back in the classroom, I had the students act out a simple improv game, which provided a nice distraction (that wasn't as arduous as jumping jacks), and also had a teaching moment. This exercise was well-received by the students.