Emma Tosch · Teaching Statement

The core principle of my teaching philosophy is to establish a **supportive and respectful relationship** with students, with the goal of guiding them to a level of **mastery** of the material that comports with **students' individual goals and values**. For early computing courses that are more skills-focused, I favor **mastery grading**. For more advanced and elective courses, I favor **labor-based grading**.

The specific teaching methods I use vary on the basis of the subject matter, students' backgrounds, and class size. That said, I have observed that the values the instructor communicates are perhaps more important than any specific teaching techniques. I believe a precondition for optimal student outcomes is *compassion*, rather than *empathy*, for students and their situations. Compassion requires the instructor lessen their own importance, while empathy can run the risk of perpetuating inequality, due to a lack of shared experiences.

I try to establish an environment that promotes **psychological safety**, while also emphasize that **learning can be uncomfortable**. When teaching early computing courses, I try to make connections between computer science and language learning, or computer science and physical exercise. I encourage my students to think about **computing as a language** for which they need to learn vocabulary and grammar, and practice every day. I encourage them to think about learning as an exercise of the mind, and to be mindful that some amount of discomfort is a sign of growth. When teaching more advanced courses — especially those that include graduate students — I try to emphasize that there is likely **heterogeneity in students' backgrounds** and that all students should feel empowered to ask questions or make observations or connections with other material. I generally try to include **interdisciplinary reading** in order to emphasize connections between whatever course I am teaching and other disciplines. I believe this serves as a counterbalance to the sometimes siloed nature of specific computing disciplines, which can cause students unfamiliar with those disciplines to feel like imposters.

Finally, I try to emphasize to students the need for **intrinsic motivation** in course work, research, and life. The **emphasis on passion in computing is exclusionary and counterproductive** for both increasing diversity, and for the pursuit of science. I have seen this emphasis on passion lead to performative behaviors from students. I also believe it increases student anxiety (e.g., via "imposter syndrome"), and as a consequence reduces learning outcomes and research output. Only individuals can decide whether their motivation is intrinsic. However, an emphasis on intrinsic motivation from instructors and mentors, who have social capital, can **legitimize the experience of students** who are pursing computing for its financial benefits (i.e., sends the message: there is no shame in wanting to support yourself in a comfortable life), for familial obligations (i.e., sends the message: a desire to please parents is a valid cultural expectation, provided it does not do harm to the student), and other motivations students may feel pressured to hide.

Instructional Experience

During my employment at UVM as an Assistant Professor, I **developed three new courses** that simultaneously target **both undergraduate and graduate students** — *Reinforcement Learning*, *Systems for Knowledge Discovery*, and *Artificial Intelligence*. As neither *Reinforcement Learning* nor *Artificial Intelligence* had been taught at UVM for some time — and did not have existing syllabi — I designed these courses to cover gaps I perceived students had in the current curriculum. *Systems for Knowledge Discovery* is a new course I developed around my research area that doubles as a research methods course.

My UVM courses were on the smaller side (enrollments of 25, 12, and 37 respectively). I designed all materials and employed self-assessments and automated grading where I could, due to having limited course support (5, 0, and 10hrs of TA time respectively, no graders). Since my courses doubled as graduate

and undergraduate courses (i.e., they were effectively two courses in one setting), I had to balance teaching topics at a depth befitting the graduate level while not alienating undergraduates. Since all of my courses were new, not pre-requisites for other courses, not required for the major, and typically filled with terminal-semester students, I took this as an opportunity to experiment with course design and pedagogy.

Project-based courses. *Reinforcement Learning* (*RL*) and *System Design for Knowledge Discovery* (*SD4KD*) I ran as **project-based courses**. *RL* fulfills the capstone project requirement of UVM's engineering education, while *SD4KD* serves a dual purpose as a research methods course and mainly attracts graduate students. In both courses, I used a **labor-based grading** approach that clearly conveys the mapping from the amount of work students do (via "earning points") to their final grade. Neither course had exams — a choice I made in light of the challenges students faced when having to take proctored timed exams during the pandemic. I taught *RL* as an **online asynchronous course** with **frequent synchronous meetings**, including mandatory one-on-one meetings with individual students, for the first two-thirds of the semester and **switched to online synchronous** for the final third of the semester. I made this switch **in response to student feedback** on the challenges of the hybrid-flex asynchronous modality.

Innovative pedagogy in a traditional exam-based course. I initially designed *Artificial Intelligence* (*AI*) to be a more traditional course, with the exception that it employed a points-based grading system (rather than percentage) with many possible paths to achieving a particular grade. The course included **exams, programming assignments, problem sets, in-class quizzes, and blogging** as options to earn points. I designed the points system to allow students to play to their strengths while lowering the cost of working on their weaknesses. Several weeks into the semester, Covid policies were in flux and many students were officially requesting leniency. Given the extraordinary circumstances we were all in, I announced that I would be **running an experiment**: I would give all students A's in the course, decoupling their course performance from their final letter. However, I would continue to evaluate them as usual; they would just all receive an A at the end. I shared with them that my hypothesis was that they would do at least as well in the course as they would have, were they to get their "earned" grade in the end. I plan to write up and distribute an experience report, but my general positive findings were that: (1) **attendance at my office hours increased**, relative to other semesters and relative to my peers' courses, (2) students showed **increased engagement** not only in course material, but also in discussions of their own pedagogy (i.e. via **participatory course design**), (3) students who had disengaged with the course in the middle of the semester **re-engaged without prompting** — something I'd never experienced before as an instructor. The main failure of this experiment was in the demographics of students whose engagement dropped and never recovered — I found this set of students contained a disproportionate number of student athletes. I consider this an exploratory experiment into designing grading schemata that (1) reduce student stress levels, (2) increasing engagement, and (3) disincentivize cheating.

Innovation in a large required course. Prior to my employment at UVM, I co-taught[1] one large class: in Spring 2018, I was invited to teach our introductory course on **topics in discrete probability**, *Reasoning under Uncertainty*. The course had approximately 180 students enrolled, and is a required course for the major. I had the opportunity to engage with many aspects of course management during this experience, including training and mentoring TAs and undergraduate course assistants (UCAs).

My main approach when teaching this course was to communicate frequently about the goals of assignments (e.g., **evaluative vs. diagnostic**), the expected time to complete assignments, and to periodically solicit feedback from students about their preferences. My past experience with this course led me to realize that there was quite a bit of heterogeneity in students' backgrounds, and that some students found

[1] Luis Pineda, now at Facebook AI Research

the intensity of the undergraduate major sequence distressing. One of the major changes we made to the course from past iterations was to have a **dual-track grading rubric**. Our objective here was to ensure that students who tested poorly had a chance to do well, and to ensure that students who tested well did not feel the need to make up "busy work" in order to maintain their grade.

Additional experience in a variety of class sizes and composition. I had been a teaching assistant for **fundamental topics** in computer science throughout my graduate studies: *Data Structures*, *Theory of Computation*, *Structure and Interpretation of Computer Programs*, *Reasoning under Uncertainty*, and *Advanced Logic*. All were listed exclusively at the undergraduate level, except for *Advanced Logic*, which also included a graduate section. The three courses I TAed while at Brandeis University were in a liberal arts setting, with class sizes that ranged from 20-50 students.

Mentoring and Advising

At UVM, I advised one PhD student, Ratang Sedimo, who is pursing his graduate degree in computer science, but whose prior degrees are in Economics. Over Summer 2022, I supervised two additional graduate students on paid summer research projects: Mako Bates, a UVM PhD student, and Erin McBride, a recent UVM graduate who is heading to University of Wisconsin Law School in Fall 2022. I was also academic advisor to 24 undergraduate students.

I have additionally served as an informal mentor to several graduate students at UVM and have regular research and career mentoring meetings with students at University of Massachusetts and Carnegie Mellon University. With Dr. Brittany Johnson-Matthews, I originated and ran an annual[2] PhD recruiting event that has a mentorship component and have been serving as a mentor for the NSF's CSGrad4US program.

During my PhD, I mentored two summer research students who participated in the REU program: Molly McMahon and Rosario Huamani Carpio. Both students worked with me on research related to the SurveyMan project. I also mentored a Google Summer of Code student, Prakhar Srivastav on a SurveyMan interface.

Summary

I have considerable formal and informal teaching experiences across a range of institutions, class sizes, and educational paradigms (e.g., liberal arts vs. engineering). I have both **taught alone** and as **part of a team**, and I have both **managed a large hierarchical course staff**, as well as **taught without TA or grader support**. Because my research and background are interdisciplinary and span several areas of computer science, I have been called on to teach a variety of courses where there is institutional need. *I hope in the future to have the opportunity to* ***teach courses closer to my research domain****, which could help* ***train potential student collaborators.***

[2]We did not run the event in 2022, but intend to run it in 2023.