1 Philosophy

My role as a computer science teacher is to give students practical problem-solving skills while also equipping them with the empathy and critical eye necessary to use those skills for good. With this in mind, I aim to develop courses that not only provide rigorous background but also directly engage with problems of ethics and current events.

My teaching and mentoring philosophy centers around three main strategies. The first is using Universal Design for Learning to create an inclusive environment by providing multiple means of engagement, representation, and expression. By removing the hidden curriculum from the classroom and making norms, outcomes, and expectations clear, I make content accessible to students from diverse backgrounds. This guides me in designing lectures, activities, and assessments that cater to different learning styles and needs, thereby leveling the playing field for all students. Second, mastery grading emphasizes the importance of understanding core concepts and skills before progressing. This approach not only reduces stress and anxiety around grades but also encourages a deeper understanding of the material. Students can revise and improve, aligning the grading process more closely with real-world scenarios where iterative improvement is the norm. Finally, I provide scaffolding for students by breaking down intimidating problems and providing students with the support they need. As they gain confidence and skills, the scaffolding is gradually removed, encouraging independence and critical thinking. This method ensures that students are neither overwhelmed by the complexity of the subject nor under-challenged, which can lead to disengagement.

Continual growth and adaptation are essential aspects of my teaching practice. Whether it's attending professional development workshops through the Center for Teaching and Learning or staying updated with developments in computer science education, I strive to be a lifelong learner. I also aim to expose students to transformative technologies, such as generative artificial intelligence (ChatGPT), to prepare them for the rapidly evolving tech landscape.

2 Experience

2.1 Teaching Assistantship

I had the pleasure to serve as a Teaching Assistant (TA) in Spring 2021 for **CSCI 3202: Introduction to Artificial Intelligence**, a course teaching the basics of search, Bayesian inference, supervised learning. Based on my teaching work, I was honored to receive the **David T. Spalding Graduate Teaching Fund Fellowship Award.** In this role, I facilitated a remote instruction model during the campus lockdown by creating study groups, providing individual and group office hours, and answering student questions via email and Piazza, an anonymous question platform. In addition to my direct interaction with students, I also developed quizzes, exams, and projects that included both conceptual and coding elements.

As a TA, my goal was to **improve student confidence in an advanced course by providing a variety of entry-points for the content**. For example, if students did not understand an example from the lecture, I recorded a video of myself showing a complete step-by-step solution and transferring the target skill to a new problem. Additionally, I emphasized critical skills like debugging and reading code by pairing students together in office hours to discuss their questions. Beyond course content, I made an effort to normalize struggling with a concept by sharing my own experiences and acknowledging that we are covering difficult material. During my end-of-semester evaluation, a student noted: *Emily* ...

makes students feel less intimidated and more comfortable to talk to her about the issues that they have. She wants to hear the problems that we have so that she can help. She also asks for feedback on what she's been doing so she knows where she can improve.

2.2 Teaching Outside the Classroom

Beyond my TA experience, I had the opportunity to teach in several informal settings. For the last three summers, I have developed and led a workshop for undergraduate research interns at the University of New Mexico; in this workshop, I covered the basics of data manipulation in Python and an introduction to data visualization as a tool for developing research questions. After the workshop, one student reached out to me and said: *I am already using some of the tools that you introduced me to in my own project. That intro was very useful.* During my fellowship with the U.S. Army Research Laboratory, I developed a series of lectures on the fundamentals of machine learning, targeted towards psychologists and neuroscientists. In the summer of 2022, I developed and led a seminar targeted to graduate students without a computational background. It covered basic concepts like control flow and functions while also introducing tools like data management documentation. One participant in the seminar noted: *This work is invaluable*. Finally, I have given back to my community by serving as a Tech Help volunteer at the Boulder Public Library for four years. In this role, I have empowered primarily elderly patrons to connect with their loved ones through video calls or social media and access community resources such as employment or legal help online.

2.3 Mentorship

I have been fortunate to mentor three undergraduate and one masters student. My mentorship focuses on **scaffolding the research process** so students can be involved in the entirety of a project. For example, I am working collaboratively with my current mentee, to harness large language models to give learners feedback. By assisting with an annotated bibliography, they are learning essential skills like searching for relevant publications, summarizing key points of previous work, and distilling what elements are helpful to our project. I have structured the process by providing search strategies, a template for summarizing the readings, and meeting face to face to explain technical concepts. This format **provides low-risk opportunities** for them to practice scientific writing and critical thinking, preparing him for graduate school or another technical career. I previously worked with a student on using semi-supervised methods to label a sparse dataset of student affect. To maximize their involvement with the project, we met to discuss the basics of machine learning as well as developing a dataset with SQL queries. As they became more comfortable with the project, I assisted in framing results by iteratively developing more sophisticated graphs and writing a final technical report. They have since been accepted into the computer science PhD program at the University of Colorado Boulder.

3 Courses

I am uniquely qualified to teach a breadth of undergraduate courses as a result of my broad research experience and interdisciplinary background in computer science and cognitive science. The following are courses I would love to teach:

3.1 Existing Courses

1. **CS ###: Introduction to Computer Programming**. This is the course I would be most excited to teach because it is often students' first exposure to computing.

2. **CS ###: The Art of Data Visualization**. I would integrate relevant topics such as ethics, journalism, and community science initiatives such as Tidy Tuesday.
3. **CS ###: Introduction to Machine Learning**. I would balance this course with the necessary mathematical foundations while also developing critical thinking around where data comes from (see Data Feminism) and the limits of these algorithms.

3.2 Future Courses

1. **Mind and Machine**: An introductory and interdisciplinary course exploring the intertwined history of cognitive science and computer science. Discusses topics such as biologically-inspired algorithms (leading to neural networks), computational linguistics, and using machine learning to understand the human brain. This course will expose students to diverse scholars and teach them to engage with public scholarship. Can be co-taught with faculty from Psychology or Cognitive Science programs.
2. **Intelligent Tutoring Systems**: An advanced course following any AI/machine learning course. Covers the history of adaptive educational systems, domain and learner modeling (such as Performance Factors Analysis and Bayesian Knowledge Tracing), data sources, interaction modalities, and ethics of using such technologies in educational settings. Will be paired with regular programming assignments that scaffold students into developing a system prototype as a final project.
3. **Introduction to Robotics**: An advanced course following any AI/machine learning course. Teaches foundations of kinematics, sensors, control systems, software, and ethics. Lectures and assignments will introduce a wide variety of robotics domains as well as open-source software projects (OpenCV, ROS). Can be combined with a lab section for hands-on experience with using hardware systems or simulation environments.