Teaching Statement

Brian Plancher
brian.plancher@gmail.com

Inspired by my interdisciplinary research interests and past professional experiences, I approach designing and teaching courses with three main goals. First, I aim to create courses that not only teach the core course material but also knowledge and skills that students can use in their future courses, research, and careers. Second, I aim to create courses that promote diversity, inclusion, and belonging, understanding that all students and colleagues enter new courses and jobs with different backgrounds, expectations, preparations, and life contexts. Third, I aim to develop course materials that benefit learners both within and beyond the walls of my classroom. To achieve these goals, I strive to design and teach courses that are **foundational**, **project-based**, **accessible**, and **inclusive**.

Figure 1: Starting the final race for the project-based robotics summer program for high school students.

During my PhD I have gained valuable experience designing and teaching courses across a variety of topics and settings to a diverse range of students. From running Harvard's introduction to artificial intelligence course with over 150 student and 10 teaching assistants, to designing advanced seminar courses on robotics hardware-software co-design and tiny machine learning (TinyML), to developing a large-scale MOOC course series on TinyML that has had over 55,000 students from 176 countries enroll, to helping launch both a tuition-free, project-based robotics summer program for high school students, and the Tiny Machine Learning Open Education Initiative, I have found that teaching has brought some of my most enjoyable and meaningful moments in graduate school and has inspired me to pursue a career in academia.

Foundational

Whether teaching an introductory undergraduate course or an advanced seminar course, my goal is that students will gain a foundational understanding of the topics covered in the course. I hope that students, regardless of their background, will leave my courses with core knowledge that they can build upon in future courses, in future research, and in their careers.

In introductory courses this manifests in a focus on teaching the concepts upon which the discipline is based. In the introduction to artificial intelligence (AI) class, my goal was for students to come away understanding the core algorithmic concepts that appear throughout the computer science curriculum and the trade-offs of using the various approaches. Therefore, I structured my lectures and sections around teaching recursion, local approximation, complexity analysis, etc., using AI algorithms like the value iteration equation as examples. I also developed an active learning exercise for the start of the robot motion planning lectures to bring to life the trade-offs of using various algorithms in the real-world. In the exercise, students are randomly given a slip of paper describing, at a high level, the strengths and weaknesses of an algorithmic approach, and are asked to discuss in groups which of their approaches would be most appropriate for a given real world robotics problem. Each group then shares their selected algorithm, and rationale, with the class. Inevitably, the class launches into a lively debate and I end the exercise by noting that there is not one "correct" answer and that it depends on the context in which the algorithm is deployed (e.g., it is not an issue if an autonomous vacuum cleaner often crashes into a wall, but it would be for an autonomous car).

In advanced seminar courses the level of detail is increased but the concepts stay the same. By carefully choosing papers for students to be read and analyze, and by asking probing questions during group discussion, I reinforce the foundational themes of the specific sub-field in every class. In a course on optimization algorithms for robotics, I gave an example paper presentation using one of my published papers. I had initially intended to focus my presentation on how our constraint handling approach compared to other common methods. However, early on in the presentation, it became clear that there was confusion over our optimization approach. As such, I re-focused my presentation to discuss the differences and trade-offs between gradient descent, Newton's method, and the method used in our paper. This led to a discussion of where, when, and why the various approaches should be applied in research and industry.

Project-Based

I have come to realize that while it is important for students to learn all of the facts presented in a given course, some of the most valuable skills I learned in the classroom were hands-on experiences, soft skills, and collaborative problem solving. As a teacher, I aim to center learning on projects and assignments that help students build and refine these skills alongside their technical knowledge.

We developed the Tiny Machine Learning course around 3 project-based assignments and a collaborative final project. To connect the assignments to real-world challenges, we had guest lectures from industry experts at leading companies such as Google, ARM, Microsoft, and Qualcom. In the first assignment, students trained a custom Keyword Spotting model (think "OK Google") on their own data and deployed the model onto an Arduino microcontroller. By closing the loop with hardware tests, students not only got to see the direct impact of their design choices, but also gained valuable experience with the entire TinyML flow. This assignment was graded not on the final model accuracy, but on the students' explanation and design of their data collection and testing scheme. This ensured that students understood the foundational challenges of TinyML and focused not only on technical knowledge but also on critical writing and communication skills.

The project-based robotics summer program for high school students was structured with weekly team-based challenges to build core competencies in motion planning and computer vision, culminating in a final autonomous car race. To build excitement and community, we not only gave out awards to teams that won the challenges, but also to teams that came up with the most creative solutions, and to teams that were the most effective at collaboration. We shuffled the teams between weeks, and in the second year we added mid-week challenges to both give students early feedback on how well their new team was working together and to provide clear, achievable checkpoints to build confidence and confirm understanding. Collaborative exercises with actionable feedback like these were instrumental in helping students from diverse backgrounds collectively solve problems, and gain confidence in the material. I spent much of my time helping teams realize that all team members, regardless of their technical proficiency, could be valuable contributors. To aid in this effort, I created a task that required less coding, but was equally important to the final race performance, autonomously triggering the race start when a "traffic light" turned green. This required careful manual tuning of color parameters to both recognize the green light and not accidentally trigger on someone in the crowd wearing a green shirt. At the conclusion of the course, students self reported, and the staff observed, significant increases in collaborative problem solving techniques, soft skills, and technical skills. In fact, the course had such a positive impact on the students then many of them, including some of the less technically proficient students, came back for multiple additional summers as course staff.

Accessible and Inclusive

By teaching with a focus on foundational and project-based learning, my hope is that students are able to build a toolkit that they can use in their futures. In order to ensure that all students are able to build that toolkit, it is important to structure courses and teach in a way that is accessible and inclusive for all of the students in the class, across their diverse backgrounds. I also strive to, where possible, enable other teachers and learners globally to benefit from, and build off of, my courses and course materials.

Within the Classroom

One way I have tried to address these issues is by ensuring that students see the same topic through different lenses of learning. For example, in the introduction to AI class I used multiple teaching tactics including:

1. Contextualizing the algorithm through videos of it in action in the real-world and/or in active research
2. Deriving the algorithm, inviting students to ask questions at each step

3. Providing a graphical explanation of the algorithm to aid visual learners
4. Connecting, comparing, and contrasting that algorithm to other algorithms learned in the class
5. Assigning theoretical questions about the algorithm's fundamental properties in problem sets
6. Assigning coding assignments to code up the algorithm for experiential learning

By using multiple approaches in my teaching, students who are visual, mathematical, and/or experiential learners get exposure to the topic in the way they learn best. And, by approaching the topic from multiple angles and through lower stakes assignments, students have more chances to evaluate their understanding and signal confusion which often raises helpful collective problem solving discussions for the entire class.

At the same time, I strive to create an inclusive environment in the course and make my office hours a safe space to ask any question. I set the tone that office hours are meant to be a space for shared learning and try to ensure that the teaching staff's office hours are available to all students by spreading them out across different days and times, and then offering additional times by appointment. I am also committed to 360° feedback and ask for anonymous student feedback after every lecture to improve the course in realtime. For example, student reflections on the pace and depth of material covered in class have enabled me to make real-time teaching adjustments, such as providing more detailed or complex examples or, alternatively, digging deeper into core concepts when needed.

Around the Globe

I want to enable the next generation of global innovators and leaders by improving access to high quality educational materials and programs on cutting edge topics. For example, to expand global access to the Harvard Tiny Machine Learning course, I worked to co-develop a four course series on TinyML on edX. The series can be accessed for free, assumes no prerequisites beyond basic programming, and includes hands-on labs that leverage both Google Colaboratory's free compute for all model training and low cost Arduino microcontrollers for deployment. We have also open-sourced all of the course materials and have a preprint under review describing our pedagodgical approach to assist others in developing similar courses [1]. Finally, based on requests from the community for further support to develop locally specific courses, seminars, and workshops on TinyML, we have excitingly launched the Tiny Machine Learning Open Education Initiative (TinyMLedu) [2], which I co-chair. The initiative has already enabled some very successful collaborations and outreach efforts.

At Your College or University

I would be excited to apply these core principles of foundational, project-based, accessible, and inclusive learning to designing and teaching courses at your college or university. In particular, I would be excited to teach a wide range of introductory and mid-level courses, especially those that cover embedded systems and (parallel) algorithms for robotics, artificial intelligence, and machine learning, as well as advanced courses on robotics and optimization. I would also be interested in exploring opportunities to develop hands-on, interdisciplinary, project-based courses collaboratively across your college or university exploring walking robots, parallel programming, tiny machine learning, and other new areas in our growing and dynamic field.

References

[1] V. Janapa Reddi, **B. Plancher**, et al. Widening Access to Applied Machine Learning with TinyML. arXiv preprint. 2021.
[2] **B. Plancher**, and V. Janapa Reddi. TinyMLedu: the Tiny Machine Learning Open Education Initiative. ACM Technical Symposium on Computer Science Education (SIGCSE). 2022.