Teaching Statement

Li (Lilly) Wu

My academic journey was inspired early on by a course I took during my graduate school, which left a lasting impression on how I view teaching. Since then, I have meticulously observed and reflected on both in-person and online courses, learning from excellent educators and evaluating the impact of different teaching methods. This continuous process of observation and reflection has not only deepened my appreciation of teaching but also guided me in developing my own teaching philosophy.

1 Teaching Philosophy

My teaching philosophy is grounded in the belief that students learn best when they are actively engaged, intellectually challenged, and supported in a welcoming learning environment. I strive to spark curiosity, encourage critical thinking, and help students develop practical skills that extend far beyond the classroom. By connecting theory to real-world applications and research-driven content, I want to prepare students for not only understanding the theoretical foundations of computing but also applying these concepts to solving complex, real-world problems.

Inspiration and Active Learning. My own academic journey was shaped by moments when teachers inspired me with cutting-edge research and challenged me with projects. I strive to create those moments for my students. I believe educators play a vital role in helping students uncover their passion, cultivate enthusiasm, and begin shaping their purpose. I want my students to discover *what truly excites them*. To foster this, I prioritize inspiration by exposing students to real-world innovations through recent papers and techniques, and emphasize active learning through group discussions and hands-on projects.

Bridging Theory and Real-World Application. The fields of computing evolves rapidly and influences nearly every aspect of modern life. It is essential to show students how abstract concepts can be applied to practical problems, helping them recognize *what they can accomplish* and why it matters. Drawing on both my research and industry experience, I bring in examples and case studies that highlight how the concepts they are learning apply to real-world applications.

Developing Technical and Soft Skills. I believe the best computer scientists and engineers are not only technically proficient but also effective communicators, collaborators, and leaders. My role as an educator is to help students learn *how to solve problems with technical depth while upholding high professional standards*. To foster this, I design collaborative assignments that emphasize both technical mastery and professional growth. In these assignments, students rotate leadership roles, present their ideas, and work with peers from diverse backgrounds.

Inclusive and Supportive Environment. Most importantly, I am committed to creating an environment where *every student feels seen, valued, and safe to express themselves*. I believe an inclusive and supportive environment sparks bold, creative ideas and cultivates a sense of belonging. To foster this, I plan to set shared classroom values, encourage diverse perspectives, and adapt my teaching based on anonymous feedback. I aim to build an engaging and interactive classroom where students connect openly with one another and with me, where laughter and learning go hand in hand.

2 Teaching Experience

My teaching experience spans China, Germany, and the United States across lecture courses, research seminars, capstone projects, and industrial research labs. I have taught courses, supervised independent studies, guided thesis projects, and mentored students at the undergraduate, master's, and Ph.D. levels. To date, I have mentored 16 students (fourteen undergraduate/master's and two Ph.D.), with outcomes including coauthored peer-reviewed publications, multiple patents, industry placements (e.g., Microsoft), and graduate study (e.g., UC Santa Cruz).

University of Massachusetts Amherst. In the summer of 2024 and 2025, I had the privilege of co-directing the UMass Turing Summer Program [4], an outreach initiative designed to broaden access to computing for high school students. I handled the program's logistics from start to finish, including creating the website, recruiting students, designing a three-week curriculum, and coordinating a team of 15 professors and 15–20 instructors/TAs. Each year, we enroll 22 students.

Beyond my co-director responsibilities, I also served as an instructor. I was constantly thinking about *how to make the program enjoyable and accessible to high school students*. To that end, I took time to learn students' names and backgrounds in advance and worked with TAs to cultivate a supportive environment. I carefully designed the schedule to balance lectures with labs, show-and-tell sessions, lab tours, and capstone projects. The lecture topics were carefully selected and related to their daily lives, including robotics, healthcare, and climate change. In particular, to give students a tangible understanding of computing, we began with a hands-on demo of a single physical server, then explored large-scale data centers through videos, culminating in a campus tour of server rooms housing hundreds of servers. By the end of the program, we were proud to receive 100% positive feedback from students.

I also had the opportunity to guest lecture in COMPSCI 230: Computer Systems Principles [1], a course with around 180 undergraduate students. My lecture focused on core system concepts, which I connected to my current research on resilient edge AI, showing, for example, how memory hierarchies can shape effective fault-tolerant system design. Beyond teaching, I have had the privilege of mentoring eight undergraduate and master's students and two Ph.D. students through summer research projects, theses, and independent studies. Many of them have gone on to pursue graduate programs or landed roles in industry, including at companies like Microsoft.

Technical University of Berlin. At the Technical University of Berlin, I worked as a Teaching Assistant for courses in cloud computing and research seminars on distributed systems and operating systems [2]. In the seminars, to engage students and maximize the learning outcomes, I carefully curated research topics with real-world relevance, particularly those aligned with the industry trends (e.g., Kubernetes). I also thoughtfully considered the learning curve and adjusted project goals according to their background, ensuring the projects remained both intellectually rigorous and accessible.

Other Institutions. Earlier in my career, I worked as a Teaching Assistant during graduate school. After completing my Ph.D., I joined Bosch Research as a Senior Research Scientist, where I mentored master's students and research interns. My experience in industry taught me that a good mentor puts their mentee's success first, helping them grow both technically and professionally, giving them space to think independently, and encouraging collaboration and leadership. As a result, this approach has paid off: my mentees have earned multiple patents and gone on to graduate programs and leading industry roles. Additionally, I have led and guest lectured at hackathons and workshops. One example is the LivingFog Hackathon in Las Naves, Spain, where I demonstrated practical applications of fog computing [3].

3 Teaching Interests

Undergraduate and Graduate Curricula I am comfortable teaching a diverse range of computer science courses across both undergraduate and graduate levels. For undergraduates, I can teach fundamental topics such as computer architecture, algorithms, programming, and software engineering. With some preparation, I am capable of teaching courses in embedded systems and information theory. For graduates, I am prepared to instruct advanced courses in distributed systems, cloud computing, advanced algorithms, and machine learning systems.

Looking ahead, I am eager to develop the following new courses:

- **IoT, Edge, and Cloud (advanced undergrad and intro grad)**: This course responds to growing workforce demands and the rapid technological development in these domains. It will cover system architecture, infrastructure, orchestration platforms, networking, and applications. I will use Kubernetes and Kubeflow as case studies and integrate research topics like edge AI.
- **Reliable AI Systems (Advanced grad)**: AI has become a central workload in both cloud and edge environments, but ensuring its reliable execution remains a major challenge. This course will explore the intersection of AI systems and distributed systems, covering the fundamentals of model training, inference, and pipelines, along with fault-tolerance and self-healing techniques.
- **Sustainable Computing (intro undergrad and intro grad)**: With hyperscale data centers creating new challenges for energy supply and carbon reduction, education in sustainable computing becomes increasingly important. This course will embed sustainability concepts into the CS curriculum, emphasizing energy-efficient design, carbon measurement and accounting, and lifecycle assessment.

References

[1] 2025. COMPSCI 230 Computer Systems Principles. https://sites.google.com/view/cs230-spring-2025/home. Accessed: 2025-08-25.

[2] DOS. 2021. Operating Complex IT-Systems. https://distributedsystems.berlin/courses/. Accessed: 2025-08-25.

[3] FogGuru. 2021. Hack the fog. http://www.fogguru.eu/living-lab/hackthefog/. Accessed: 2025-08-25.

[4] LASS. 2025. UMass Turing Summer Program. https://lass.cs.umass.edu/turing. Accessed: 2025-08-25.