Teaching Statement

Shawn Shuoshuo Chen

Teaching is a key feature that sets a faculty job apart from other careers in the technology sector. It is also one of my reasons for becoming a professor. I enjoy teaching students and find it rewarding to see students apply the knowledge and skills acquired in the classroom to real-world challenges. My previous experiences have prepared me for the teaching and mentoring aspects of an academic career.

Teaching

I am interested in teaching undergraduate and graduate-level courses in computer networks and operating systems. I am also interested in teaching introductory courses such as introduction to programming. In addition, I would like to develop a graduate-level course relevant to my research. This course will focus on the architecture and design of high-performance warehouse-scale computers, with topics ranging from the design of warehouse-scale interconnects using emerging photonic technologies to the exploration of cost- and power-efficient topologies for modern AI clusters.

I have participated in teaching in two roles: (1) as the multiple-time head teaching assistant for undergraduate computer networks courses at CMU and UVA, and (2) as a guest lecturer for both undergraduate and graduate-level systems courses at CMU. Through these experiences, I have developed a teaching philosophy rooted in experiential learning and inclusive teaching. Below, I discuss them in detail.

Aligning the curriculum with industry practice. The computing industry is always evolving. Even the most recent textbooks can be disconnected from what is widely used in the industry. Given my extensive experience working at Google and Microsoft, I have brought my knowledge of modern data centers and AI supercomputers into the classroom. For instance, I gave lectures on Google's optical switching-based data center network and Microsoft's specialized AI accelerator cluster[1] design in three systems courses.

The computing industry has a great tradition of publishing "postmortem" summaries after major outages. This is an excellent way to learn how some complex production systems work. As an example, to help students understand how the BGP routing protocol and DNS operate on today's Internet, I created homework problems and exam questions based on the 2021 Facebook outage[2]. In these assignments, students were given a timeline of key events during the outage and were asked to derive the final routing and DNS states. Many students doubted their answers, as their results indicated that the Facebook hostname was inaccessible. However, they were astonished to find out this was exactly what happened and simultaneously felt gratified that they truly understood BGP and DNS.

Fostering both know-why and know-how. In systems courses, I believe it is important to reinforce conceptual understanding (know-why) with practical skills (know-how). For example, students typically learn IP routing by acting as a "human router": they manually convert IP addresses from the dot-decimal notation (e.g., 192.168.1.1) to a binary format using pen and paper, and then decide where an IP packet goes according to a set of routing rules. I designed a class project to help students deepen this understanding with hands-on exercises. The project required students to implement their own routing and load balancing functionality using C++ in the Click software router[3]. If they implement the functionality properly, students will see the quality of a live video stream improve from low (due to network congestion) to its original high quality in a web browser. As part of the project, students also learned to troubleshoot networking problems by performing packet capturing using Wireshark[4] and creating virtualized network environments using Kubernetes containers[5]. According to feedback from some students, the tools and debugging skills they learned from this project also helped them pass job interviews.

Providing students with personalized learning experience. A student's background, time commitment, and interests all contribute to their ability and pace to learn. My instructor and I used several techniques to accommodate these differences, one of which was the flipped classroom model. Some core networking concepts, such as distance vector routing, can be time-consuming to comprehend. To adapt to each student's pace, we supplemented traditional lectures with flipped classrooms. In this model, students

[1] https://azure.microsoft.com/en-us/blog/azure-maia-for-the-era-of-ai-from-silicon-to-software-to-systems/

[2] https://en.wikipedia.org/wiki/2021_Facebook_outage

[3] https://github.com/kohler/click

[4] https://www.wireshark.org/

[5] https://kubernetes.io/docs/concepts/containers/

reviewed lecture notes on their own and then worked on homework problems in class. This allowed me to provide one-on-one guidance to those who needed extra help. This technique proved effective in helping many students master the concept. As a result, I observed that students posted fewer questions about distance vector routing on the online Q&A forum compared to previous years.

Mentoring

I have mentored three junior PhD students and seven undergraduate/masters students. I worked with some of them on my research projects and guided others on their own course projects. My approach to mentoring, shaped by both my own advisors and my mentees, is to be flexible and to foster independence. I believe mentoring is a mutual learning experience in which I continually adapt to each student as much as they adapt to me.

Each student has their unique strengths and preferences. I have learned to be flexible in project planning so that each student can work at a pace that maximizes their productivity. One student was strong at execution but relatively new to systems research. I met with her twice a week to help build her background knowledge, review code, and plan next steps. She successfully ramped up after a semester and is now on track to publish her first paper. Another student was self-motivated and very independent; he liked to present milestone results with polished plots and slides. We collaborated through less frequent meetings, scheduling ad hoc discussions whenever there was major progress. This way, he could focus more on the actual work and reduce the overhead of providing updates.

I foster independence by focusing on building a reusable “essential skill set” rather than providing single-use solutions. An example is how I approach teaching literature review, a foundational skill for any researcher. Instead of just assigning papers and asking for a summary, I first provide the students a concrete analysis framework—a set of key questions guiding them to deconstruct the paper’s problem statement, core insights, methodology, and evaluation. We apply this framework together for the first two or three papers, with me modeling the process of critique and probing for unstated assumptions. After they are comfortable with this scaffold, I deliberately shift my role. The students are then responsible for finding relevant papers, applying the framework independently, and leading our discussion. This method ensures they have not just read a paper, but have learned how to read any paper, empowering them to confidently and autonomously explore new research topics.