Ravi Netravali — Teaching Statement

The opportunity to work with students, both through teaching and research mentoring, is my primary motivation to become a professor. In addition to the gratification I derive from the "aha moment" that students feel when they first grasp a concept, I believe that teaching is an excellent platform to refine research tools and to articulate research ideas in their simplest forms, promoting new ideas and applications. I have been fortunate to have had teaching and mentoring experiences throughout my undergraduate and graduate education, and I am excited to contribute to computer science education in the future.

Teaching experience: As an undergraduate student at Columbia University, I was a TA for the Fundamentals of Computer Systems (CSEE 3827). This course explores basic processor design from the ground up, teaching topics like combinatorial circuit design, memory structures, and instruction set architectures. In addition to holding office hours, I assisted in creating and grading problem sets and exams.

As a graduate student, I was a TA for Digital Communication Systems (6.02) at MIT. This course introduces undergraduate students to EE and CS concepts, from the physical layer through the transport layer, which combine to create an end-to-end communication system. In addition to helping design and grade written assignments, programming labs, and exams, I performed weekly check-offs with 30 students (out of the course's 300 students) each week. These check-offs served as an excellent opportunity to understand how best to help students with very different backgrounds and understanding of course material. I created several questions each week to ensure that students could apply the key techniques described in lecture, and then met with students one-on-one to address their questions and to ensure that they were up-to-speed. Following each week's check-offs, I generated a set of notes which anonymously summarized the provocative questions and comments that came up during individual meetings. I found that these notes provided different perspectives on the course material, and helped increase the students' curiosity in the course topics.

Finally, I have had the opportunity to lead lectures as a PhD student. In 6.02, I delivered several recitation lectures which aimed to reiterate the key topics from the course, and extend them through theoretical and practical examples. I have also prepared and delivered guest lectures on my research in MIT's graduate networking course (6.829) and graduate seminar on Advanced Topics in Networking (6.888).

Teaching approach: In addition to delivering material coherently, I believe that the primary roles of an instructor are to keep students engaged and to provide continual support. Through my experiences as a student and as a TA, I have developed two insights. First, students are most engaged when they generate directions of exploration, rather than solely being lectured about conventional wisdom. The best way to achieve such engagement is by applying course concepts to concrete systems. Moreover, I think that assignments should include open-ended components that promote exploring interesting directions rather than trying to find the one "correct" solution. Such "hacking" not only allows students to generate, test, and refine ideas, but also illustrates the value of the underlying concepts which they are learning.

Second, I have found that courses are most beneficial when they leverage the diverse backgrounds and interests of students. Delivering a concrete course curriculum which lays the foundation for critical computer science topics is mandatory. However, integrating multiple perspectives into lectures is challenging and often feels unnatural. Instead, augmenting lectures with small group exercises and instructor meetings allows students to more easily and freely express their genuine thoughts and questions. Sharing knowledge from these sessions can be invaluable to students as they attempt to integrate the course material into their future endeavors; this approach was well received in the digital systems course for which I was a TA.

Though these techniques require more time from an instructor, and will become increasingly difficult to implement given the growing sizes of CS courses, they allow students to maximize their excitement and the insights that they draw from a course. Allowing such unbounded thought may also introduce previously unconsidered directions to instructors, potentially helping courses evolve over time.

Courses I can teach: I would be qualified to teach undergraduate and graduate courses in computer networking, computer/distributed systems, cloud computing, and introductory computer science topics. Additionally, I am interested in generating a new course on practical systems. Many of the techniques that are taught in systems and networking courses have apparent applications in services that students use daily. For example, data consistency guarantees are embedded into the protocols that drive web transactions. Additionally, components of these services are often open-source. While courses today do incorporate system building, I have found that students are often unaware of the powerful services that are publicly available; a priori knowledge of these technologies can significantly aid system design and implementation. Thus, I think that a course which breaks down how commonly used services (e.g., web applications) operate in practice by exploring how subsystems (e.g., databases) and new technologies (e.g., Amazon Lambda) cooperate to provide rich functionality and availability, and how tradeoffs are settled (e.g., security vs. performance), would be very useful. Understanding these systems by interacting directly with their code bases will help generate new research ideas (by identifying desirable features and the roadblocks preventing them), and will better prepare students to collaborate with and work in industry.

Mentoring: During my PhD, I have been fortunate to work with several excellent students at various levels. I provided guidance as a senior graduate student to Vikram Nathan [2] and Hongzi Mao [1] at MIT, and Vaspol Ruamviboonsuk [3] at the University of Michigan, as they worked through their first research projects in graduate school. I also supervised several undergraduate and masters students on their respective theses: Somak Das, Ameesh Goyal, Eeway Hsu, and Elizabeth Dethy.

Through these mentoring experiences, I have found that students perform best when they drive their own research, while seeing progress and high-level vision. Additionally, I believe that students excel once they understand that research is truly interdisciplinary, and that there are multiple skillsets and knowledge bases that are required to produce "good" research. This realization motivates students to identify their strengths and areas that require improvement, enabling them to actively contribute to research projects (boosting confidence) while developing breadth. Thus, my role as an advisor would be that of a facilitator and motivator, encouraging students to pursue directions of interest, without fear of exploring new areas and skillsets. Frequent meetings and other forms of feedback are invaluable to overcome intermediate roadblocks, and will help ensure that students make rapid progress towards their goals.

References

[1] H. Mao, R. Netravali, and M. Alizadeh. Neural Adaptive Video Streaming with Pensieve. In *ACM SIGCOMM*, 2017.

[2] R. Netravali, V. Nathan, J. Mickens, and H. Balakrishnan. Vesper: Measuring Time-to-Interactivity for Modern Web Pages. In *USENIX NSDI*, 2018.

[3] V. Ruamviboonsuk, R. Netravali, M. Uluyol, and H. V. Madhyastha. Vroom: Accelerating the Mobile Web with Server-Aided Dependency Resolution. In *ACM SIGCOMM*, 2017.