Teaching Statement

Rishabh Iyer

The opportunity to teach, mentor, and learn from students is a major reason why I am pursuing a faculty career in academia. I find teaching especially fulfilling and have discovered that it deepens my understanding of the subject by requiring me to explain concepts in the simplest way possible. I also derive great satisfaction from mentoring students, particularly as I see them take increasing ownership of their research and shape it with their unique perspectives. In the rest of this document, I first discuss my experiences and approach to teaching before doing the same for research mentorship.

1 Teaching

Teaching experience. I have been a teaching assistant (TA) on *eight* occasions for courses ranging from introductory math and physics, to undergraduate computer science courses, and finally advanced graduate courses on computer systems.

My most relevant classroom teaching experience comes from Principles of Computer Systems (POCS), the most advanced graduate systems course at EPFL. In POCS, the instructor introduces a principle (e.g., modularity or redundancy) during the first lecture of the week, with the TAs leading a discussion of seminal systems that exemplify the principle in the second lecture. My approach during these lectures was to have students discover the nuances of the system *by themselves* and through *guided interaction*. For each system, I would have the students start with a simple set of constraints and design goals, then proceed through a series of strawmen, understand the problems with each, and gradually arrive at the final design. In my experience, this approach significantly enhances students' critical thinking skills and allows them to better extract the key ideas underlying a system's design.

I have also had the opportunity to TA for Software Engineering and Software Development Project (SDP) at EPFL, a unique pair of courses in which students first spend one semester learning the skills required to write production-quality code (via lectures interspersed with exercises) before spending another semester learning how to engineer software products by building an Android application from scratch using the Scrum development methodology [7], with TAs acting as product owners. The second course—which consists of no lectures—made me realize how much better students *understand an idea when they see it materialize in a real system*. Going forward, I plan to have a significant hands-on component in every systems-related course I teach.

Courses I can teach. I am qualified to teach undergraduate and graduate courses in computer systems, as well as undergraduate courses in networking and computer architecture. I would also be glad to teach introductory computer science courses such as introduction to programming, algorithms, and discrete mathematics.

I am keen to introduce two new courses. The first will be a seminar for graduate students focused on techniques that enable building systems with well-understood performance properties. This seminar will span work in computer systems, networking, architecture, formal methods, and real-time systems to give students a holistic view of the field. As outlined in my research statement, I believe building such systems will be increasingly crucial in the future, and this course will provide graduate students with a comprehensive view of the state of the art.

The second course will focus on how techniques from program analysis and formal methods can unlock new possibilities in networked systems. This hands-on course will be aimed at senior undergraduates and junior graduate students and will not only introduce them to the state of the art but also provide them with the skills necessary to apply these techniques to their own research. I believe that such an interdisciplinary course, if taken early, can shape a student's research outlook by enabling them to make connections and pursue directions they were previously unaware of.

2 Research Mentorship

Mentorship experience. I have been fortunate enough to mentor a total of **eleven** students: five undergraduate students during summer internships at EPFL and six junior graduate students during my time as a senior student and then postdoc. Several of my summer interns are now PhD students at top universities, including UC Berkeley, Stanford, and EPFL.

I am particularly proud of the research output of the students I mentored. Three of my summer interns made significant contributions to projects I led, which ended up appearing at top conferences. Musa Unal played a pivotal role in getting the Concord project (SOSP'23) [3] off the ground, Ayoub Chouak helped extend

the CFAR project (OSDI'24) [2] to find timing violations in cryptographic libraries, and finally, Daneshvar Amrollahi improved CFAR by integrating first-order logic-based loop summarization techniques.

I have also guided five graduate students to their own first-author publications at a top venue. At EPFL, I mentored Jiacheng Ma and Kartikeya Dwivedi, whose work appeared at OSDI'24 [5] and SOSP'24 [1], respectively. During my time at Berkeley, I mentored Alex Krentsel, Ziming Mao, and Rathin Singha, all of whom successfully published their first-author papers at HotNets'24 [4, 6, 8]. My resume contains more details about each of these students and their work.

Mentorship approach. My approach to mentorship is informed by the following insights gained through my experiences working with students; however, I acknowledge that there remains much for me to learn and refine.

First, I believe students are most productive when they are *excited* about their work. To foster this, I make it a priority to communicate the broader vision of each project at the outset and emphasize why the student is well-equipped to realize this vision. I have found that this is particularly valuable when working with undergraduates and interns. For instance, Musa had significant expertise in low-level Intel hardware, which made him enthusiastic about tweaking the OS scheduler in Concord, Ayoub was passionate about security, which made him eager to extend CFAR to cryptographic code, and finally Daneshvar was primarily interested in formal methods, which made him keen to learn about automated loop-summarization techniques. I believe that this alignment between the students' interests and the project topic ensured that the students were excited about their work, and was a key reason for them being so productive.

Second, I recognize that each student is unique and benefits from a mentoring approach tailored to their strengths and preferences. For instance, some of the graduate students I mentor at Berkeley, such as Ziming and Rathin, benefit from frequent meetings and structured feedback at each stage, while others, such as Jiacheng and Kartikeya, thrive when given greater independence to explore ideas. I believe it is crucial for an advisor to adapt their approach to each student, to ensure that the student realizes their full potential.

Finally, I encourage students to gradually take ownership of their work. Early on, I offer close support through regular meetings and collaborative whiteboard sessions. Over time, I guide them to form and test their own hypotheses, helping them grow into their roles and take full ownership of their projects.

In summary, I am excited about all aspects of being a professor, including conducting research, teaching courses, and mentoring students. I look forward to guiding students to achieve their goals and teaching them the engineering and communication skills required to solve problems they are passionate about.

References

[1] Kumar Kartikeya Dwivedi, Rishabh R. Iyer, and Sanidhya Kashyap. Fast, Flexible, and Practical Kernel Extensions. In *ACM Symposium on Operating Systems Principles*, 2024.

[2] Rishabh R. Iyer, Katerina J. Argyraki, and George Candea. Automatically Reasoning About How Systems Code Uses the CPU Cache. In *USENIX Symposium on Operating Systems Design and Implementation*, 2024.

[3] Rishabh R. Iyer, Musa Unal, Marios Kogias, and George Candea. Achieving Microsecond-Scale Tail Latency Efficiently with Approximate Optimal Scheduling. In *Symposium on Operating Systems Principles*, 2023.

[4] Alexander Krentsel, Rishabh R. Iyer, Isaac Keslassy, Sylvia Ratnasamy, Anees Shaikh, and Rob Shakir. The Case for Validating Inputs in Software-Defined WANs. In *Hot Topics in Networks, HOTNETS*, 2024.

[5] Jiacheng Ma, Rishabh R. Iyer, Sahand Kashani, Mahyar Emami, Thomas Bourgeat, and George Candea. Performance Interfaces for Hardware Accelerators. In *USENIX Symposium on Operating Systems Design and Implementation*, 2024.

[6] Ziming Mao, Rishabh R. Iyer, Scott Shenker, and Ion Stoica. Revisiting Cache Freshness for Emerging Real-Time Applications. In *Hot Topics in Networks, HOTNETS*, 2024.

[7] Scrum Development in Software Engineering. https://en.wikipedia.org/wiki/Scrum_(software_development). [Last accessed on 2024-10-23].

[8] Rathin Singha, Rishabh R. Iyer, Charles Liu, Caleb Terrill, Todd D. Millstein, Scott Shenker, and George Varghese. If Layering Is Useful, Why Not Sublayering? In *Hot Topics in Networks, HOTNETS*, 2024.