Teaching Statement

Hashim Sharif

Throughout undergraduate and graduate school, I have drawn significant inspiration from my teachers and mentors. Courses and mentorship opportunities played a crucial role in exposing me to a variety of exciting topics in computer science, and eventually helped shape my research interests in compilers, systems, and program analysis. I would find it extremely rewarding to exert a similar positive influence on new students through teaching and mentorship. I am deeply passionate about teaching both core and specialized computer science topics. Through teaching, my goal is to share my technical expertise with students and help them in their research and career development.

1 CLASS TEACHING

Experience. At UIUC, I was one of two teaching assistants for **CS 498 Digital Forensics** in its first offering. The goal of this course is to teach students computer networks and computer security techniques related to investigating evidence from digital data. I was responsible for leading a weekly lab; I led all lab discussions, assigned tasks to students, and graded students on their lab assignments. As TA duties, I also designed and graded a midterm and the final exam and graded student assignments. I also had the opportunity to give three guest lectures. These include two guest lectures in **CS 526, Advanced Compiler Construction**, and one guest lecture in **CS 598 Approximate and Probabilistic Computing**.

I learned valuable things from these teaching experiences. Through weekly office hour sessions for CS 498, I was able to identify concepts students found challenging. These close interactions helped me understand how one can explain complex concepts in multiple ways and how using analogies and examples can make it easier to understand challenging topics. Multiple students personally mentioned that they found these one-on-one tutoring sessions as an opportunity to brush up on course material. In the CS 526 guest lectures, I discussed compilers for heterogeneous systems, and got an opportunity to present my work on ApproxHPVM. Presenting my research work in a class setting, I learned the importance of covering more background material. In CS 598, I gave a lecture on compilers for deep learning and compiler systems for approximate computing. My lectures engaged the students in discussions; I received a lot of questions and was pleased to find students were making connections to their research.

Teaching Philosophy. I view teaching as a channel for long-term impact. I like to keep my lectures interactive and engaging. I strongly focus on creating an environment where students don't feel intimidated to share their ideas. After class, I also like to hang around for a few extra minutes to have one-on-one student interactions. For graduate courses, I want to make my lectures be discussion sessions where students and instructor(s) debate the pros, the cons, and the implications of research papers discussed in class. I strongly believe in project-oriented courses where different project components focus on different concepts. To encourage collaboration among students, I am an advocate of doing projects in teams.

2 RESEARCH MENTORING

Experience. I am fortunate to have closely mentored **seven undergraduate** students (including 2 women) and **four PhD** students; and co-authored papers with all of them. All the undergraduate students who have graduated are now pursuing PhD programs at top schools. I started mentoring Nathan Zhao and Peter Pao-Huang while they were high-school seniors. They are both now pursuing their undergraduate programs at UIUC. I have regular weekly meetings with Nathan and Peter, and we actively work on research projects. I have co-authored 3 conference papers [3–5] with Nathan, and one paper with Peter [5]. I worked closely with Yasmin Sarita (Cornell undergrad student) and Elizabeth Wang (UIUC undergrad student), and have co-authored two papers [3, 4] with Yasmin, and one paper [4] with Elizabeth. These experiences encouraged Yasmin to apply for graduate school and she is now pursuing a PhD at UIUC.

On the TRIMMER project [1, 2] that focuses on code size reduction techniques, I mentored Abdul Rafae Noor, Mubashir Anwar, Shoaib Asif, and Usama Hameed while they were undergraduate students at the Lahore University of Management Sciences. Through this project, I closely worked with the students on a challenging research problem, helped them learn LLVM, and helped them improve their writing and presentation skills, while we worked together on the journal version [1] of the TRIMMER project [2]. These research experiences motivated them to apply for grad school. Abdul Rafae and Mubashir are pursuing their PhD programs at UIUC, Shoaib is currently pursuing his PhD at UT Austin, and Usama is pursuing his PhD at UCLA. I continue to mentor Abdul Rafae on the HPVM project. Mubashir and I are collaboratively working towards releasing our TRIMMER project source code.

On the ApproxTuner [4] project, I closely worked with and mentored Yifan Zhao (PhD student) and Akash Kothari (PhD student); Yifan is co-advised by Vikram Adve and Sasa Misailovic, and Akash is advised by Vikram Adve. Yifan and I also collaborated on ApproxCaliper [5]. I have co-authored two papers with Yifan [4, 5], and one paper with Akash [4]. For field experiments with production-ready agriculture robots, I closely worked with Vatsin Ninad Shah (UIUC undergrad student), and we have co-authored one paper [5].

I am fortunate to have had the pleasure of mentoring exceptional students who have made valuable contributions to our research projects. By mentoring students, I have gained valuable experience in guiding students working on research projects. I feel fortunate to have had the opportunity to play a positive role in the career development of my mentees. I try to be there for them whenever they need my advice on research or any career-related issues. I have strong working relationships with all my mentees, and I am very proud of their accomplishments.

Mentorship Philosophy. My mentorship style is very hands-on. I like to have regular meetings, brainstorming sessions, and code reviews. My mentorship goal is to actively help students refine their technical and non-technical skills. I strongly believe in Nico Habermann's statement: "Focus on the students, since graduating great students means you'll produce great research, while focusing on the research may or may not produce great students". I try my best to strike an appropriate balance in dividing my time between long-term student mentoring and short-term project-focused goals. I like to cultivate a vibrant research environment with open communication and active collaboration among team members. My project management strategy is to have clear role descriptions that align well with the expertise and research interests of students.

I like to take a slightly different approach when mentoring PhD and undergraduate students. When mentoring undergraduate students, my goal is to assign them well-defined tasks. To motivate PhD students towards becoming independent researchers, I give them more freedom in choosing their own research problems and encourage them to independently find new solutions and approaches. With both PhD and undergraduate students, I like to be very involved in project brainstorming, I like to conduct regular code reviews, and take active part in giving students feedback in their own research writing and presentations.

3 TEACHING INTERESTS

As an assistant professor, I would be interested in teaching courses across a broad range of areas. At the graduate and undergraduate level, I am qualified and interested in teaching **Compiler Construction, Program Analysis, Operating Systems, and Systems Programming.** At the undergraduate level, I would also be happy to teach **Computer Architecture, Parallel Programming, and Data Structures.**

I am excited to offer two new graduate-level courses on topics related to my research:

- **Compilers for Efficient Machine Learning**. This course will discuss compiler system development for efficient machine learning on the edge. It will cover the latest literature on machine learning compilers and domain-specific optimization techniques applied to machine learning workloads. I will also discuss neural network-specific approximation techniques such as quantization and pruning in the context of efficient system support. This will be a graduate-level course, and students will be assigned a significant final project. To prepare students for the course project, I will design programming assignments that focus on: i) machine learning optimizations, and ii) LLVM-based compiler transformations that focus on optimizing core workloads in deep learning (e.g., matrix multiplication).
- **Approximate Computing Techniques and Practices**. This course will get students acquainted with the latest literature on approximate computing techniques. This course will include three modules focusing on: a) software approximation techniques, b) approximations at the hardware architecture level, and c) compiler frameworks, languages, and tools for approximate computing. I will also discuss how these techniques are applied in various application domains such as machine learning, data analytics, and augmented and virtual reality (AR/VR). This will be a graduate-level course with the final project being the most significant portion of the class grade. I will also grade students for presenting research papers in class, and require them to submit paper reviews.

I am excited to start my journey as a teacher. Through teaching and mentorship, I look forward to helping my students grow in their careers, while also continuing to learn myself.

REFERENCES

[1] Aatira Anum Ahmad, Abdul Rafae Noor, **Hashim Sharif**, Usama Hameed, Shoaib Asif, Mubashir Anwar, Ashish Gehani, Junaid Haroon Siddiqui, and Fareed M Zaffar. 2021. TRIMMER: An Automated System for Configuration-based Software Debloating. *IEEE Transactions on Software Engineering* ***(TSE'21)***.

[2] **Hashim Sharif**, Muhammad Abubakar, Ashish Gehani, and Fareed Zaffar. 2018. TRIMMER: Application Specialization for Code Debloating . ***(ASE'18)*** .

[3] **Hashim Sharif**, Prakalp Srivastava, Muhammad Huzaifa, Maria Kotsifakou, Keyur Joshi, Yasmin Sarita, Nathan Zhao, Vikram S. Adve, Sasa Misailovic, and Sarita Adve. 2019. ApproxHPVM: A Portable Compiler IR for Accuracy-aware Optimizations. ***(OOPSLA'19)*** .

[4] **Hashim Sharif**, Yifan Zhao, Maria Kotsifakou, Akash Kothari, Ben Schreiber, Elizabeth Wang, Yasmin Sarita, Nathan Zhao, Keyur Joshi, Vikram Adve, Sasa Misailovic, and Sarita Adve. 2021. ApproxTuner: A Compiler and Runtime System for Adaptive Approximations. ***(PPoPP'21)*** .

[5] **Hashim Sharif**, Yifan Zhao, Peter Pao-Huang, Vatsin Ninad Shah, Arun Narenthiran, Mateus Valverde Gasparino, Nathan Zhao, Abdulrahman Mahmoud, Sarita Adve, Girish Chowdhary, Sasa Misailovic, and Vikram Adve. 2022. ApproxCaliper: Exploiting Application-level Error Resiliency for Optimizing Neural Networks. Under Review at *ACM SIGMETRICS 2022*.