Teaching Statement

He Ye
hey@cs.cmuedu

Over the past five years during my Ph.D. study (2018-2023), I have served as a teaching assistant (TA) for five distinct courses, accumulating a total of 13 teaching experiences, including two undergraduate courses and three Master's courses. I have supervised four Master's theses at KTH and am currently supervising three Ph.D. students at CMU. All of these teaching experiences have helped me gain valuable insights as both a teaching assistant and a supervisor. I split my teaching statement into the following three sections: 1) teaching experience, 2) teaching philosophy, and 3) future teaching courses.

Statement of Teaching Experience

Teaching experience as a Teaching Assistant (TA): In 2019, I served as a TA for the undergraduate course DD1334 Database Technology (a total of 64 TA hours per year). This involved organizing weekly 2-hour Q&A lectures for over 100 undergraduate students, providing assistance during lab sessions, and grading assignments. I also served as a TA for the Master's course DD2480 Software Engineering Fundamentals (a total of 64 TA hours per year) spanning 2018, 2019, and 2020. In this role, I guided students through weekly 4-hour lab sessions, supported them in coding tasks, and provided grading and feedback on their project work. Additionally, I was a TA for the course DD2482 Automated Software Testing and DevOps (a total of 82 TA hours) spanning 2019, 2020, and 2021, conducted entirely on GitHub to simulate real-world team collaborations. My responsibilities included reviewing students' Pull Requests for various tasks such as research paper summaries, competitions, demos, and presentations. I managed the merging process, provided comments, and graded their tasks.

All of the above courses involve hands-on software engineering tasks with a heavily lab session setting. In my lab teaching experience, I observed many students getting panicky when they encountered error messages or faced issues. They immediately raised their hands to ask for help instead of taking a moment to think about where and how to solve the problem. Rather than providing direct answers, I would first guide them to carefully read the error diagnostics and encourage them to consider possible solutions. I consistently advise my students not to be afraid of making errors; errors are friends that come with helpful diagnostics for problem-solving.

Teaching experience as an advisor for undergraduate student project work: I have served as a project advisor for the undergraduate course DD1393 Software Engineering, totaling 64 TA hours, in 2018, 2019, and 2020. In this course, I annually supervised two groups of undergraduate students, each comprising 8-10 students, as they collaborated on real-world industrial projects (e.g., a map-based audio guide for Nordiska Museet and image recognition software for MSAB, etc.). My role included guiding them through the process of eliciting software requirements, both functional and non-functional, offering technical solutions using open-source frameworks, assisting them in managing project risks, and ensuring the successful delivery of their projects to industrial partners.

Teaching experience organizing Master's student seminars: In my teaching role for the Master's course DA2210 Introduction to the Philosophy of Science and Research Methodology for Computer Scientists (a total of 72 TA hours per year), spanning the years 2018, 2019, and 2020, I was responsible for organizing student seminars. This included conducting 10 seminar sessions, each accommodating approximately 20 students, with each

session lasting 45 minutes. The seminars covered fundamental topics related to mathematics, natural sciences, and computer science. My primary objective for these seminars was to foster discussions with diverse opinions and encourage interaction among as many students as possible. To facilitate varied engagement, I implemented various formats, including group discussions, group swaps, panel discussions, presentations, feedback sessions, and group competitions. In my experience, employing these diverse seminar formats has proven effective in enhancing interaction among students. More importantly, it helps that every student has the opportunity to express themselves and actively engage in listening to others.

Supervision experience of Master and Ph.D. students: I mentored four master's students at KTH through the process of selecting a research topic for their Master's theses until successful defenses. This included collaborating on two theses with Spotify and Ericsson to integrate our research into an industrial context. As a Postdoc at CMU, I mentored three Ph.D. students. Recognizing that students new to research often lack a comprehensive understanding of the field, I observed that the provision of clear and specific tasks significantly helps them to kick off the project. To enhance students' clarity of research goals, I employed three main approaches: 1) weekly one-on-one meetings, 2) a curated list of related works, and 3) an evolving list for idea development.

I organized weekly individual meetings with each student to review their study and research progress, jointly analyze models, and offer constructive feedback. Importantly, I also care about their personal and career plans, providing support to help them achieve these goals. I always recommend as many as related work to students and ask them to write a short paragraph to compare their current work with prior work. This helps students to have a comprehensive understanding of their research field at a higher level. Additionally, it benefits them in identifying the uniqueness and novelty of their own research contributions. I promoted the maintenance of an idea list, collaboratively refining and discarding concepts to nurture their own ideas and passions.

Feedback from my students indicates positive experiences with my supervision, highlighting the efficiency gained by avoiding the valuable time in searching and struggling. For instance, one of the Ph.D. students I supervised decided to pursue a Ph.D. while he was working on his master's thesis, and I collaboratively aimed to develop his research mindset. Consequently, within the first month of starting his PhD, I mentored him to transform his Master's thesis into a research paper. In all my supervision experience, our collaborative approach fosters mutual encouragement, aiding us in achieving our collective goals.

Statement of Teaching Philosophy

I have always believed in the importance of good teaching. The students represent the future innovators who will shape advanced technologies for the betterment of humanity, and it is our responsibility it’s our duty as stewards of higher education to enable students to succeed regardless of the career path they choose. I believe that good teachers can not only help students acquire known knowledge, but also encourage them to explore the unknowns, help them spark their passion for learning, and deepen their understanding of real-world phenomena. This belief is the main source of my enthusiasm for teaching and my commitment to improving students’ learning experiences.

For undergraduate teaching, my teaching philosophy emphasizes two aspects: 1) *the establishment of a solid foundation in fundamental CS knowledge* and 2) *bringing practical examples into lectures*. Through my teaching experiences, I've observed that many courses provide high-level instructions, yet students often struggle to follow them due to a lack of foundational knowledge in areas such as compilers, operating systems, networks, and algorithms. I place significant emphasis on instilling these fundamental CS concepts as they are indispensable for aspiring software engineers. I value the integration of real-world examples into our courses as a means of emphasizing the practical application of the knowledge we provide. This demonstration of how and where to adapt knowledge serves to enhance students' comprehension of the techniques we teach by illustrating the real-world relevance and applications of the concepts covered in the curriculum. Comprehending knowledge holds greater significance than merely knowing and memorizing.

For PhD supervising, my teaching philosophy has largely evolved under the influence of my Postdoc supervisor and PhD supervisors.
Listen and be open-minded**.** All my supervisors possess open-mindedness toward research topics and encourage students to explore diverse subjects and methodologies. I am committed to adopting a similar approach as a supervisor, to carefully listen to what they want to do and assist my students in developing into mature researchers to find out their research interests and strengths.

Let students shine. Empowering my students and giving them due credit is essential for fostering independence and maturity. While I provide more guidance to junior students, as they progress to senior levels, I empower them to make their own decisions regarding research projects and career paths. I consistently advocate for students to independently engage with other research groups in academia and industry, attending conferences and seminars, and becoming sub-reviewers and program committees, etc.

Passion plays a pivotal role, especially given the likelihood of encountering more failures than successes in research. I believe that cultivating passion leads to sustained hard work, enabling both my students and myself to navigate and overcome challenges.

I value science more than publications (learned from my supervisors). I emphasize the importance of mental health in the work environment, recognizing that publication metrics alone do not adequately evaluate our scientific achievements. I provide sufficient encouragement for all the work that has not yet reached success, refraining from making judgments based solely on outcomes.

Statement of Future Teaching

I am interested in enhancing existing SE courses by incorporating a focus on Machine Learning Operations (MLOps). This addition aims to deepen students' understanding of architectures for deploying and managing ML or LLM models. The objectives of incorporating MLOps into these courses include:

1) Introducing practical skills in using MLOps tools and technologies, such as Git, Docker, Kubernetes, Jenkins, and other relevant platforms.
2) Familiarizing students with end-to-end MLOps workflows, enabling them to seamlessly integrate machine learning models into production environments.
3) Developing the ability to address challenges related to model deployment, monitoring, and maintenance in real-world project scenarios.

New courses. Moreover, I am considering the introduction of new courses:

- Prompting Engineering of LLMs
- Understanding and Hands-on Transformers: fine-tuning LLaMa and ChatGLM
- LLM Practice with langchain and knowledge graph
- Automated Software Testing and DevOps
- High Availability Software Architecture

These new courses aim to cover cutting-edge topics such as engineering LLMs, hands-on experience with Transformers, and the integration of automated software testing with DevOps practices. I hope these courses will provide students with valuable insights and practical skills relevant to the rapidly evolving field of software engineering and machine learning.