Teaching Statement

My most enjoyable years are learning and teaching computer science in my graduate school. When I am learning, I feel happy and my mind becomes sharp. Students' insightful questions often give me different perspectives to understand the topics. What I really believe in is student-centered teaching philosophy. Teaching is two-way communication between instructor and students. Teaching a class is like guiding students on a wonderful trip to the best scenery. An excellent tour guide always knows where to go, what to do, and what tourists are looking for. To teach a class, my first step is to understand the students, their existing knowledge, learning styles, and their goal. It is key to get this information early, through consultation with experienced instructors, or set up survey and questionnaires with students. The syllabus will be designed considering their existing knowledge, interests and learning styles. During the lecture, I invite questions after delivery of each module. Doing so not only engages students to think but also get feedback from students. I also ask students questions in the lecture to get them to think about the content. Then delivery of lecture content will center around answering the questions. It is important to motivate students and provide roadmap from the beginning of the journey to the end. Let them be aware of the importance of the course (for example, it is a foundation course whose applications can be found in many other courses, or it is a course of advanced study). During the entire course, I believe it is key to know how students are doing at each step and prepare to adapt the teaching.

I gained my first teaching experience when I served as a teaching assistant (TA) in the Department of Computer Science at the University of Georgia. I was a teaching assistant for Introduction to Computing and Programming for three years. Before the course start, I met with the professor to understand the course design. I tried to put myself into professor's shoes and imagine it was me to design the course and teach it myself. This course was an introductory course for students whose major might not be computer science. Therefore, an interdisciplinary approach is taken. Specifically, the labs and projects drew many practical problems from a wide range of subjects, such as personal finance, biology, math, physics, economics and so on. During labs and office hours, I showed students that computer programming is a tool that can help us solve many problems. In addition to guiding students through labs and help students with questions about finishing the projects, I often briefly started with some background about formulating lab/project problems from real life. Once introduced to thinking and formulating a problem in computing way, most students got very excited knowing that they could solve many other problems in their majors. They also developed intuition about computers and programming languages with analogies to human language and get more natural feeling learning the course content.

Another course I have assisted to teach for two years is algorithms. I showed to students that algorithms can be applied everywhere, from operating systems, network, database design, AI to machine learning. I prepared the introduction to the common algorithms and their applications to help them to see the roadmap. For students that have strong interests and prior experiences,

I expanded their understanding with more related applications of each algorithm. When a student applies the same algorithm to a different but same category of problems, he gets better mastery of the algorithm. For students that consider algorithms course too mathematical and hard to understand, I presented them with a simplified problem and visualized procedures that are representative of the basic form of the algorithm. When their feedback was receptive, I started to show them how to improve the algorithm, for example, adding consideration for corner cases and finding ways to optimize its time and space complexity.

During my years of work at AT&T Labs, I have designed and provided a series of training on current topics software engineering, distributed system design, and machine learning

- Building large-scale distributed systems with Apache Hadoop and Spark, live demos are provided drawing work projects of building mobility network data systems and analytics
- Building software infrastructure for real-time streaming data processing with Apache Kafka, drawing application of Apache Kafka to AT&T mobility network event trace data platform
- Doing data science with python. Live demos have been drawn from work projects that use Jupyter notebook, pandas, matplotlib, seaborn, scikit-learn, keras and tensorflow

In addition, I have also mentored many new and junior employees at AT&T Labs, sharing my R&D experience and helping them finish projects as a team. Going back to academia from industry, I would love to bring industrial experience to college students to benefits their career development.

In summary, I believe my passion in computer science learning, teaching and industrial research and development experience setups a good foundation to teach students about computer science and advise their career development.