I believe teaching is not a one-way street but an enjoyable two-way interaction. I am grateful for every opportunity I’ve had to teach and mentor students in the past and excited about making teaching a significant part of my career. My goal is not only to present Computer Science (CS/ECE)-related materials where our students can learn, but also to help them discover and grow their passions for learning and research in CS and ECE.

**Classroom Teaching:** Teaching will be a life-long learning process. It is encouraging as teachers to share in the student’s growth and achievements, and also challenging as the educator’s role is related to students’ future. From my past teaching and learning experiences, I’ve gradually developed a vision for how my teaching should be and how to self-improve to be a better teacher. These past experiences have inspired my teaching philosophy:

1. **Understand our students, respond to learners with different styles and backgrounds.** As Peter Honey and Alan Mumford identified in 1982, learners have four different learning styles due to their different backgrounds: Activist, Theorist, Pragmatist, and Reflector. As a teaching assistant at the Johns Hopkins University (JHU) for EN.600.463 Introduction to Algorithms (Spring 2015), I observed numerous students each with different backgrounds and learning styles. In addition to the students in CS, many students came from other departments such as Chemistry, Biomedical Engineering, and Applied Math, and they had slightly different learning goals and expectations. For example, some CS and Math students may already be familiar with how to analyze time/space complexity of an algorithm, but would like to learn more about how these algorithms are useful in real world applications. Although the course is designed to understand and analyze basic algorithms, I created a series of optional discussion sessions for the interested students to “practice” these basic algorithms and how to solve a problem based on the algorithms. At the very first session, I set up a theme to understand asymptotic time/space bounds: Implement the $O(n^2)$ and $O(n \log n)$ sorting algorithms and measure their computation and memory. At another session, I put up a problem to find the element that appears more than 1/3 times in an array that led students to understand and modify the Majority Vote algorithm. By a rough estimate, 30+ out of 73 interested students attended at least once or coded at least one problem. More than once, students came back with excitement saying the sessions were very useful in helping them better grasp the materials.

2. **Create an environment for engagement and practicing thought process.** I think it is more important for students to practice and improve their problem-solving thought process than just getting the answers. In Spring 2017, I was a teaching assistant and a guest lecturer for EN.601.633 Algorithms I with 92 students. With growing student enrollments, I tried to think about how to make a better environment for students to engage in this one of the largest CS courses at JHU. I was the only TA for this course but luckily the department hired a few graders to help me with the assignments. I was still worried about student engagement. Thus, in addition to my two 2-hour office hours per week, I asked the student graders who have available time to organize informal discussion sessions: We led discussions on selected prior assignment questions without knowing the answers. Anyone who had ideas on how to solve the problems was encouraged to share their thought process. Meanwhile, other students may experience some “aha moments” by thinking over others’ thought processes. In this environment, students had more chances to improve their problem-solving skills and cultivate computational thinking [1].

Some of the students even ended up forming a group to prepare for internship interviews together.

3. **Value the feedback and make self-improvement.** Naturally, teaching activities need to be tightly coupled with feedback, including feedback to the instructors or the students. As a junior researcher, I view classroom teaching as an excellent opportunity to practice the way of demonstrating and conveying content effectively, which is also a role of being a professor in the research community. When I was teaching EN.601.633 Algorithms as a guest lecturer in Spring 2018, I received feedback asking for more examples of how the covered graph algorithms (e.g., Tarjan’s algorithm for strongly connected components) are used in practical systems. I subsequently posted more materials to Piazza for an online discussion on how Neo4j and Spark GraphX use graph algorithms to process large structured graph data, and talked about the algorithmic issues and challenges from these systems in the next review lecture. For then, I also led a team consisting of TAs and student graders to provide timely feedback to students’ questions and assignments. I found that a combination of online and in-person discussion sessions is effective at giving and receiving feedback and encouraging self-improvement. I plan to strengthen such feedback channels and learn other effective teaching methods.

**Courses I can teach:** Given my background, I can teach graduate and undergraduate general courses in computer networks, algorithms, and computer/distributed systems. Beyond the general courses, I am also interested in organizing seminar courses such as *Advanced Topics in Networking* and *Advanced Topics in Systems Research*.
covering the latest advances in software-defined networking, programmable networks, ML systems, cloud and edge computing. I hope these seminar courses can boost research engagement in the field and spark new research ideas. I'm also excited about building on my research expertise to design a new course on *Algorithmic Principles of Computer Systems* to explore efficient algorithms (e.g., data streaming algorithms) that offer insights into the design of computer systems (e.g., datacenter networks, databases, and storage systems). It will be a hybrid course with hands-on and analytical materials to help students learn a useful set of algorithms, explore algorithms' system use cases, and evaluate how trade-offs are settled (e.g., memory, performance, and accuracy).

**Mentoring:** During my Ph.D. and Postdoc, I have been fortunate working with several students at different levels. I am mentoring or have mentored Ph.D. and master's students: Zhishuai Zhang (Johns Hopkins University), Anup Agarwal, Aqsa Kashaf, Tian Li, Hun Namkung, Yucheng Yin, and Mingran Yang (Carnegie Mellon University), and Renju Liu (University of California Los Angeles) on various topics that we share mutual interests, e.g., network telemetry, security and privacy for ML, and edge computing. When working with these brilliant junior students, my approach is to guide them to carefully formulate the problem first and encourage them to convince everyone (including themselves) that they are solving an important, meaningful problem instead of just telling them so. Then it is important to motivate students drive the projects. When students are facing difficulties, I found that it is effective to encourage students not to make immature judgments, seek intra- and inter-group discussions, and challenge students on the key issues. Some of these projects are finalizing towards submissions or already under submission. Through the Peking University-CMU summer internship program, I supervised an undergraduate student Zhuo Cheng on building a network analytics framework for IoT platforms, and I facilitated the collaboration between Zhuo and Mingran to actively prototype the system. I guided Zhuo on how to approach the problem via careful design process, and we worked together on extending sketching algorithms for better memory efficiency. After the program, I’m thrilled to continue mentoring Zhuo and Mingran on a conference submission and help their applications to CS and ECE Ph.D. programs.

Through these mentoring experiences, I realize that it is crucial for an advisor to form a warm, collaborative environment for students to be successful. As an educator, a professor bears extensive responsibility and shares valuable rewards in the meanwhile. I will motivate and guide students to pursue their goals and follow their passions to contribute to the CS and ECE communities. I sincerely believe team chemistry is a key ingredient in the formula for research success. As a teacher and an advisor, I hope I can be the catalyst to the team chemistry among students.

**REFERENCES**