Teaching Statement - Ziyang Li, University of Pennsylvania

I am deeply committed to teaching and fostering student learning. During my undergraduate and graduate studies, I have served as a teaching assistant for a wide range of courses, from foundational topics like data structures (CSE 12 at UCSD) to advanced subjects such as software analysis (CIS 5470 at UPenn), programming languages (CSE 130 at UCSD), computer graphics (CSE 165, 167 and 190 at UCSD), and large language models (CIS 7000 at UPenn). Beyond traditional courses, I have also translated my research on neurosymbolic programming into impactful educational opportunities, offering summer school courses (SSFT 2022 and SSNP 2024) and delivering tutorials at premier conferences (LOG 2022 and PLDI 2023). These experiences have equipped me with the skills and perspective to teach a broad range of topics, including programming languages, formal methods, and artificial intelligence. At the graduate level, I aspire to lead seminar courses on cutting-edge areas like neurosymbolic methods and AI-assisted software development, sparking discussions, new research directions, and collaborations.

Integrating Research into Education

During my participation in the *Summer School of Formal Techniques* (SSFT) 2022 and the *Summer School of Neurosymbolic Programming* (SSNP) 2024, I developed and delivered courses and hands-on labs based on my research on Scallop, a neurosymbolic programming language. These sessions covered foundational topics such as logic programming, relational databases, and probabilistic reasoning, as well as advanced concepts like designing and training neurosymbolic models. As a 3rd year PhD, I was the only student lecturer at SSFT 2022, standing alongside professors, which reflects both the depth of my expertise and my ability to translate cutting-edge research into impactful teaching.

The courses and tutorials I developed received wide acclaim. At SSFT 2022, where the audience primarily came from formal methods, many students approached me afterward to express their appreciation for my detailed explanations of logic programming and probabilistic reasoning. They were particularly inspired by how I introduced neurosymbolic programming into the domain of formal logic, broadening their perspective on the field. Teaching these advanced topics not only expanded students' views but also led to professional collaborations, including co-authored publications and a joint grant proposal on neurosymbolic methods with UC Berkeley. I also delivered invited lectures in graduate courses at UPenn on trustworthy AI, natural language processing, and large language models.

Mentoring Experience

My teaching extends beyond the classroom to mentoring undergraduate and junior PhD students, guiding them through their first research publications and fostering experiences that have significantly advanced their academic trajectories. During the Summer and Fall of 2023, I mentored undergraduates Jason Liu, Liam Dodds, Eric Zhao, and Felix Zhu from UC Berkeley, culminating in a publication on integrating foundation models into neurosymbolic systems (AAAI 2024). Through personalized guidance and support, I helped these Freshman and Sophomore students grow into promising researchers who are now continuing their academic pursuits and planning to apply for doctoral programs.

Writing a first research paper is often a transformative yet challenging experience for new PhD students. I mentored Alaia Solko-Breslin and Paul Biberstein during their first year as PhD students, helping them

navigate their first research projects in the area of neurosymbolic methods. My mentorship focused on refining their problem-solving skills, developing their technical writing, and instilling confidence in their ability to conduct research.

Teaching Philosophy

In recent years, the computer science curriculum in academia has had to adapt to the rapid industry advancements in areas such as machine learning, software engineering, and interdisciplinary applications. My teaching philosophy emphasizes bridging theory and industry-level practice. As a TA for Software Analysis (CIS 5470, UPenn), I designed new homework assignments by requiring students to detect simple vulnerabilities in real-life programs rather than traditional toy example programs. This approach gave students a tangible sense of creating solutions with real-world impact. While not every vulnerability was expected to be uncovered due to the complexity, I introduced a leaderboard system that sparked enthusiastic competition and encouraged collaboration and skill-sharing among peers. While serving as a TA for VR technology courses (CSE 165 and 190, UCSD), I helped students to work on final projects that reflect diverse interests—from gaming and music to healthcare—making the material both accessible and engaging for a wide range of students.