

Teaching and mentorship are central to my vision as a faculty member. I have found both to be deeply rewarding and hope to continue teaching and mentoring students across diverse backgrounds and career paths.

Teaching

My teaching experience spans a wide range of contexts and student populations. As an undergraduate at Stanford, I served as a teaching assistant nearly every quarter, primarily in the Section Leader Program where I led weekly discussions for 10-12 students in the introductory CS sequence. In my final year, I served as head TA for the second-level introductory CS course (CS106B), coordinating dozens of other teaching assistants and managing course logistics. One of my most rewarding TA experiences was teaching an experimental course for non-majors (CS101). Several students from that class have gone on to stellar careers outside CS—one recently contacted me about their transition into cybersecurity and intellectual property law, while another serves in a public policy role, assisting members in the U.S. House of Representatives. These outcomes underscore my belief that engaging students of all backgrounds is crucial; and, teaching non-majors allows me to shape how the broader world understands and uses technology.

At Berkeley, I have continued teaching as a graduate student TA for the undergraduate data engineering course, Data 101. These experiences have shaped my teaching philosophy, which emphasizes **contextualizing computing within its historical and intellectual landscape**. When teaching query optimization in Data 101, I found that walking through the historical evolution of join algorithms—from early nested loop joins used in simple relational systems, to sort-merge joins optimized for sequential disk access, to hash joins that became practical once main memory grew large enough to hold partitions—helped students understand why optimizers choose different strategies under varying workload and hardware constraints. Instead of memorizing that “hash joins are efficient for equijoins,” students could reason about when memory constraints or data sizes would favor different algorithms. This approach makes the material more engaging and less rote, while building the intuition students need to make real engineering decisions.

Beyond the classroom, I also care about teaching practitioners who are building AI systems in the real world. After finishing my undergraduate degree, I worked as an ML and data engineer in industry before starting my PhD, and I have maintained close ties with that community. Over the summer, I created the syllabus and course content for an AI evaluation course aimed at industry engineers and technical PMs. **The course has reached over 3,000 practitioners**¹, and the accompanying course reader will appear on arXiv soon and later as an O’Reilly book. Building this course was deeply rewarding because it required me to think critically about what skills I used daily as an engineer 5–6 years ago and how those skills need adaptation in the era of LLMs. I adapted methodologies from my PhD work—e.g., grounded theory for error analysis, LLM judges for evaluation, statistical correction for measurement, and interface design for human review—into practical frameworks that engineers could immediately apply. This experience enriched both my teaching and research, showing me how academic methods can solve real-world problems and revealing new research directions from practitioners’ challenges.

Based on my research and teaching experiences, I am prepared to teach courses spanning databases, data management, data engineering, and human-computer interaction at the undergraduate level. Beyond these established courses, I would be excited to develop a new and experimental undergraduate course on *AI Engineering* that brings the practical skills from the AI evals course I wrote into an academic setting, while also incorporating emerging topics like software engineering with AI-assisted tools. **At the graduate level, I look forward to teaching seminars on AI-Powered Data Systems**, which would couple readings and discussion of recent research with an open-ended project where students build novel systems or apply techniques to advance their own research in other domains.

Mentorship

Mentorship is one of the most rewarding parts of my work. **I have mentored 15 students for six months or longer, spanning high school through early-stage PhD levels.** All of them have co-authored a paper with me or co-presented

¹<https://testimonial.to/ai-evals-course/all>

a poster at a lab research event. About half are women or underrepresented minorities in computing. My mentees have gone on to have strong outcomes: one undergraduate, *Parth Asawa*, received PhD offers from both MIT and Berkeley and is now pursuing a PhD at Berkeley; another is at Columbia and received the NSF Graduate Research Fellowship. Three of my undergraduate mentees have been nominated for the CRA Outstanding Undergraduate Researcher Award. Three others are applying for PhD programs this cycle with highly competitive research portfolios. I have also mentored PhD peers; for example, *Bhavya Chopra*, who as a first-year PhD student co-led two UIST papers—an uncommon achievement for someone in her first year of graduate study.

Illustrative Mentorship Examples. With Parth, an undergraduate who first reached out after reading my blog posts from my pre-PhD days in industry, mentorship evolved over several years as we worked through a sequence of projects on data quality systems. We began with ML-powered data cleaning for relational data, where he contributed to experiment implementation. From there, we moved into LLM-powered data cleaning and later LLM-powered data validation, extending these techniques to unstructured settings. When we started, he had no plans to do research beyond one semester, but he became increasingly engaged with the process of framing questions, running experiments, and interpreting results. After helping run experiments for two of my papers, he began developing his own research ideas. **He is now a second-year PhD student at Berkeley, has transitioned to a different research area, and already has two first-author papers**—a trajectory I am very proud of.

While Parth’s mentorship spanned several years, my work with Bhavya in her first year illustrates a different focus: helping an early-stage PhD peer develop her footing. We worked on two projects together: DocWrangler, an interactive interface for authoring unstructured data pipelines, and a collaboration on AI-assisted dataset discovery. Across both, I helped her develop intuition for what makes compelling insights—probing deeper than surface-level analysis to uncover findings that genuinely advance understanding. In DocWrangler, we worked closely as I shared my thinking about user study design. By the dataset discovery project, she was running studies and identifying insights independently. We also worked on communication skills: giving talks, distilling ideas for different audiences, practicing presentations. **She is now leading her own follow-up work based on insights from DocWrangler**, with the confidence and toolkit to drive it forward independently.

Mentorship Philosophy. My approach emphasizes two things: (i) helping mentees build concrete research skills *step by step*, and (ii) helping them develop the confidence to define problems *on their own*. Early on, I focus on establishing a structure—e.g., clear goals, small parts of projects, regular feedback, and visible progress. As mentees grow, I gradually shift responsibility toward them and emphasize independent decision-making and reflection. Over time, the goal is for each person to move from execution to ownership: identifying problems, proposing solutions, setting their own research direction, and most importantly—having a good time.

Based on my positive experience in my PhD, I would like for my future group to have the following qualities:

- **Diverse and complementary.** I will recruit members with different technical strengths and perspectives, spanning data management, systems, AI, and HCI. Projects will be designed to be complementary rather than competitive, so that students can learn from each other while developing distinct areas of expertise.
- **Tight-knit and supportive.** I want to foster a close-knit culture where students enjoy hanging out with each other, modeled after my own advisor’s group, with shared appreciation and respect for each other’s contributions.
- **Collaboration-forward.** I plan to continue the co-mentoring model I already practice, where students work closely with each other and with collaborators across disciplines. I have co-mentored four mentees with postdocs who are more theory-oriented or AI-oriented, and I enjoy feeling like I’m always learning something new through these collaborations; they keep the work fresh and dynamic.

Ultimately, what I value most in teaching and mentorship is seeing people grow into confident, curious researchers who enjoy what they do. The best part of my job is learning alongside others—when their ideas challenge mine and push our work forward. As a faculty member, I hope to build a group and classroom culture that captures that same energy: thoughtful, collaborative, and fun.