I work in theoretical computer science, specifically on designing algorithms for fundamental problems on graphs and networks. Traditionally, algorithms are studied in the *worst-case* setting: for an algorithm to be deemed "fast", it must run quickly on *all* instances of the problem. The appeal of worst-case analysis is its *robustness* and its emphasis on concrete, universal statements: algorithms must work well for *all* inputs. In an unpredictable world, we prefer to have such strong guarantees. However, designing optimal algorithms for the worst-case is difficult, and it is often easier to consider *average-case analysis*. Here, algorithms are designed and analyzed for typical instances, rather than for worst-case instances, often leading to faster, cleaner algorithms. However, these average-case algorithms lack the strong universal guarantees of the worst-case, trading off *robustness* with *optimality and simplicity*.

In my research, I seek to bridge these two fundamentally different mindsets in a way that achieves the best of both worlds. The core technique in my approach is *preconditioning*: transforming, or *conditioning*, an arbitrary input instance to *behave like an average-case instance*. In this way, I can focus my attention on typical, non-pathological instances, and then translate the results back to derive worst-case bounds. In other words, I combine the optimality and simplicity of average-case analysis with the robustness of the worst-case setting.

With the preconditioning technique, I have made substantial progress on basic algorithmic questions that have resisted many previous efforts. In the global minimum cut problem, we are given a network of nodes connected by links, and we want to delete the minimum number of links to break the network into two. A long-standing open problem is whether a *deterministic* algorithm can find this minimum in near-optimal time, i.e., in time close to the size of the network itself. In recent work, I answer this question positively using a preconditioning-based approach. I first consider typical instances—e.g., random networks, where each node connects to a small, randomly chosen set of neighbors—and apply techniques from average-case analysis. Then I transform a general network into one that looks random-like: I split it into connected clusters so that (a) within each cluster, the connections between nodes look random, and (b) there are very few connections between nodes of different clusters. I solve each cluster separately using average-case analysis, and then stitch the solutions together according to the decomposition. Another problem I solve using preconditioning is the network connectivity problem in the *dynamic* setting: given a network that changes over time, keep track of whether it is currently connected. While the technical details differ between problems, the overall strategy of preconditioning is similar.

In fact, preconditioning has become a central problem-solving *philosophy* guiding my research. Given a new research problem, one of my first instincts is to ask, "What if the input behaves like random?" This question often leads to valuable insights, even in projects that do not end up using preconditioning in the final result. This happens in my work on the minimum k-cut problem (a generalization of global minimum cut above) where we want to break the network into k pieces, not two. I show that the popular Karger-Stein algorithm, a staple in modern algorithms courses, actually runs faster than its classical analysis proves; moreover, the running time that I derive is in fact optimal. To achieve this improvement, I first study the Karger-Stein algorithm for random networks and prove that the original analysis can be improved, which was not at all obvious beforehand. Motivated by this insight, I then generalize my techniques to all networks, this time without the need for preconditioning.

So far, my successes with preconditioning are mainly in the context of network algorithms. In the future, I plan to broaden my scope beyond network problems and explore the full power and generality of the preconditioning technique. Ultimately, my vision is to demonstrate the power and versatility of preconditioning not only as a technical tool, but a general philosophy that unites the simplicity and optimality of the average-case setting with the robustness of the worst-case.

I believe UC Berkeley will be an amazing place to foster my scientific growth and make my research vision a reality. The faculty and students at UC Berkeley cover an immense depth and breadth of algorithmic topics, from network flow algorithms (Prof. Satish Rao) to algorithms for high-dimensional data (Prof. Jelani Nelson) to data science (Prof. Peter Bartlett). The Simons Institute only magnifies the depth and breadth of research at UC Berkeley, bringing together top researchers from around the world. This diversity is a great strength, and the varied research experiences and perspectives form the perfect breeding ground for new insights on preconditioning and beyond.