I have been fortunate to engage in teaching at a scale that is rare for Ph.D. students. During the first two years of my Ph.D., I redesigned Stanford's primary undergraduate computer vision course and have been instructing it for the past three years [16]. Last academic year, I also began co-instructing Stanford's graduate computer vision course with Professor Fei-Fei Li [14], which has grown to become Stanford's second largest course. In total, I have instructed nearly a 1000 Stanford students and expect to be teaching another 600 students in this academic year. Courses at other universities are already using the assignments I designed [13] and some have even adapted my research papers [11] into their own assignments (e.g. MIT's HCI course [7]).

Aside from teaching, I have been consistently running a research group for undergraduate, Masters, and Ph.D. students who are interested in the intersection of computer vision and human-computer interaction. In total, I have advised 23 students in research, 7 of whom have been accepted to top computer science Ph.D. programs and 8 to industry research labs. One of them has co-founded a computer vision startup using ideas from our research together. Ten of my students have published 11 first-author papers to computer vision and human-computer interaction conferences with me as an advisor [1, 2, 4, 5, 6, 8, 10, 18, 19, 22, 3]. Another 6 students have contributed to my first-author publications as co-authors [11, 12, 15]. I am proud of having advised 2 honors thesis for Stanford undergraduates, one of which won the Ben Wegbreit Prize for the best honors thesis [17].

Cultivated by my experience, my teaching philosophy focuses on providing students with opportunities to actively learn by asking questions and by teaching others. These two goals have defined how I organize course materials, structure assignments, incentivize grades, and scaffold research.

TEACHING EXPERIENCE

I started my doctoral studies with a strong belief that the growing ubiquity of "black-box" neural networks made it important to expose students with a counterbalance — a historical perspective of “white-box” methods. It is desultory to expect students to ask questions that critique large black-box systems without a historical perspective on alternative options. In computer vision, for instance, modern approaches predominantly use convolutional neural networks and yet, most courses do not explain what characteristics we should expect from these systems. To provide such a perspective, I re-designed Stanford's undergraduate computer vision class [16] to open by characterizing which properties of human sight we want to mimic in machines — namely shift-invariance, i.e. the ability to identify objects even if they move. The course derives discrete convolutions by restricting linear dynamical systems to produce shift-invariant outputs. Students then learn to hand-craft convolution operations to extract gradients, modify image properties, and even stack multiple convolution operations to detect lines and objects. Later on, when they learn about convolutional neural networks, students are armed with the experience to question whether modern architectures are shift-invariant and are capable of concluding that they are not [21].

Aside from designing course content, I have attempted to incorporate my teaching philosophy in how I design mechanisms to challenge top students while allowing a reasonably hard-working student to successfully develop the fundamentals, and attain a satisfactory grade. Designing opportunities was particularly pertinent because during the three years that I have taught the undergraduate course, it doubled in size. Meanwhile, the graduate computer vision course I teach is now the second largest course at Stanford. To make my classes accessible, I created recitation sessions, led by my teaching assistents during the first weeks of class, which expose students to the pre-requisite material. To stretch the top students to think deeply, I developed extra credit opportunities in every assignment, in the form of difficult theoretical proofs and even competitions where students competed for the best empirical results on a test set [13]. I also initiated an extra credit option for students to teach one another by co-authoring publicly available course notes [9].

TEACHING PLANS

While it would be natural for me to teach future courses in computer vision, I would also eagerly teach undergraduate- and graduate-level courses in artificial intelligence, human-computer interaction, and natural language processing. I would also be particularly interested in developing new courses at the intersection of these fields: vision-language grounding and human-AI interaction.

Recent advances have showcased how linguistic signals can improve visual representations and similarly, how visual inputs can improve task-agnostic language models. Given their mutual benefit and their reliance on a common set of models, I would like to develop a course that explicitly brings multi-modal visual-linguistic grounding to the forefront with the goal of developing agents that can act in the real world while communicating their intentions and following others' instructions.

Artificial intelligence is instantiated by human data, utilized to solve human goals, and ultimately a tool to improve human experiences. I aim to design another course centered primarily around human-AI interaction. This course would articulate design decisions for model interpretability, human agency, minimal machine learning bias, mixed-initiative systems, human augmentation, user trust, and pro-social societal implications.

MENTORSHIP

Good research is pedagogical. Although research is often viewed as distinct to teaching, effective scaffolds can quickly teach anyone how to conduct impactful research. Over the last few years, I have been developing scaffolds that apply my teaching techniques to research mentorship.

One such scaffold, the *SPICE* technique (state-problem-insight-challenge-execution), helps students structure their ideas by asking themselves "what previously unsolved problem their idea tackles?" and "why is it a difficult technical problem that merits exploration?". The first three letters describe steps to answer the first question: describe the *state* of related work, explain a *problem* that is unsolved given that state, and introduce their ideas as an unique *insight* to tackle the problem. The last two letters answer the second question: articulate the technical *challenges*, and plan out the *experiments* that justify the utility of the insight. My students have found this technique to be effective when writing manuscript introductions, giving elevator pitches, and brainstorming.

Another scaffold of mine has been largely shaped by ideas surrounding agile research [20], a framework for scaling up research collaborations. I have been running three types of weekly community sessions: stand-ups, reading groups, and social meetups. My students come together during these meetings to share updates, present new ideas, share new research papers, and most importantly, ask one another for help. Students use these sessions to teach one another new concepts, pilot user studies, critique ideas, and develop meaningful support connections.

Mentoring students has helped me acquire a deeper grasp of technical concepts and research methodology. Following my students' interests has resulted in establishing dozens of collaborations with professors from both within and outside of computer science. I will strive to continuously improve how I mentor students and foster even better community dynamics in my future research groups.

[1] Vincent Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Re, and Li Fei-Fei. "Scene Graph Prediction with Limited Labels". In: *International Conference on Computer Vision*. 2019.

[2] Apoorva Dornadula, Austin Narcomey, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. "Visual Relationships as Functions: Enabling Few-Shot Scene Graph Prediction". In: *ArXiv*. 2019.

[3] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. *AGQA: A Benchmark for Compositional Spatio-Temporal Reasoning*. 2021.

[4] Kenji Hata, Ranjay Krishna, Li Fei-Fei, and Michael Bernstein. "A Glimpse Far into the Future: Understanding Long-term Crowd Worker Quality". In: *CSCW: Computer-Supported Cooperative Work and Social Computing*. 2017.

[5] Khaled Jedoui, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. *Deep Bayesian Active Learning for Multiple Correct Outputs*. 2020. arXiv: `1912.01119 [cs.CV]`.

[6] Jingwei Ji, Ranjay Krishna, Ehsan Adeli, Juan Carlos Niebles, Olga Russakovsky, and Li Fei-Fei. *Compositionality in Computer Vision*. `http://cicv.stanford.edu`. 2020.

[7] David Karger, Lea Verou, and Amy X. Zhang. *6.813/6.831 - User Interface Design and Implementation*. `http://web.mit.edu/6.813/www/sp18/assignments/rs1-implementation/`. 2020.

[8] Pranav Khadpe, Ranjay Krishna, Li Fei-Fei, Jeff Hancock, and Michael Bernstein. "Low Expectations Lead to Better Experiences: The Effect of Conceptual Metaphors on Human-AI Collaboration". In: *ACM Conference on Computer-Supported Cooperative Work and Social Computing*. 2020.

[9] Ranjay Krishna. *Notes for CS131 Computer Vision: Foundations and Applications*. `https://github.com/StanfordVL/CS131_notes`. 2017.

[10] Ranjay Krishna, Ines Chami, Michael Bernstein, and Li Fei-Fei. "Referring Relationships". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[11] Ranjay Krishna, Kenji Hata, Stephanie Chen, Joshua Kravitz, David A Shamma, Li Fei-Fei, and Michael S Bernstein. "Embracing error to enable rapid crowdsourcing". In: *Proceedings of the 2016 CHI conference on human factors in computing systems*. ACM. 2016, pp. 3167–3179.

[12] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. "Dense-Captioning Events in Videos". In: *International Conference on Computer Vision (ICCV)*. 2017.

[13] Ranjay Krishna and Juan Carlos Niebles. *Assignments for CS131 Computer Vision: Foundations and Applications*. `https://github.com/stanfordvl/cs131_release`. 2017.

[14] Ranjay Krishna, Danfei Xu, and Li Fei-Fei. *CS231N Convolutional Neural Networks for Visual Recognition*. `http://cs231n.stanford.edu/2020/`. 2020.

[15] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. "Visual genome: Connecting language and vision using crowdsourced dense image annotations". In: *International Journal of Computer Vision* 123.1 (2017), pp. 32–73.

[16] Juan Carlos Niebles and Ranjay Krishna. *CS131 Computer Vision: Foundations and Applications*. `http://vision.stanford.edu/teaching/cs131_fall1718/`. 2017.

[17] Junwon Park. "Cooperation Learning: Leaving to receive cooperation using social strategies". In: *Stanford University* (2019).

[18] Junwon Park, Ranjay Krishna, Pranav Khadpe, Fei-Fei Li, and Michael Bernstein. "AI-based Request Augmentation to Increase Crowdsourcing Participation". In: *AAAI Conference on Human Computation and Crowdsourcing*. 2019.

[19] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. "Generating semantically precise scene graphs from textual descriptions for improved image retrieval". In: *Proceedings of the fourth workshop on vision and language*. 2015, pp. 70–80.

[20] Haoqi Zhang, Matthew W Easterday, Elizabeth M Gerber, Daniel Rees Lewis, and Leesha Maliakal. "Agile research studios: Orchestrating communities of practice to advance research training". In: *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. 2017, pp. 220–232.

[21] Richard Zhang. "Making Convolutional Networks Shift-Invariant Again". In: *International Conference on Machine Learning*. 2019, pp. 7324–7334.

[22] Sharon Zhou, Mitchell Gordon, Ranjay Krishna, Austin Narcomey, Durim Morina, and Michael S Bernstein. "Hype: Human eye perceptual evaluation of generative models". In: *Thirty-third Conference on Neural Information Processing Systems* (2019).