CS Town Hall 2021

Slides
Video Recording

Student Organizers
ACM: Rucha Patki, William Chern, Katherine Chong
UPE: Chloe Jiang, Kevin Zhang, Rishan Girish, Kendrake Tsui, Jessica Cheng

Faculty/Staff in Attendance
Omid Abari, Guy Van den Broeck, Michel Moraga Campbell, Kai-Wei Chang, John Cho, Jason Cong, Paul Eggert, Eliezer Gafni, Quanquan Gu, Alina Haas, Miryung Kim, Rich Korf, Todd Millstein, Baharan Mirzasoleiman, Carey Nachenberg, Tony Nowatzki, Nanyun Peng, Peyton Reddick, Veronica Santos, David Smallberg, Yizhou Sun

ACTIONABLES

This is a list of actionables that has been compiled from Town Hall responses as well as student responses from our surveys.

Diversity

- Hiring **more diverse professors and TAs**
  - Highlight minority professors that we already have. This will bring more visibility and showcase the diversity within the department.
  - Use code examples where the central character is female
- Consciously thinking about **students with disabilities** when creating course material
  - Color-blind conscious powerpoints
  - Accessible homework
  - Currently, class size makes it difficult to ask for accommodations
- **Retire "weed-out classes"**
  - Instead of trying to push students out, they should simply be exposed to more practical aspects of working in CS/tech so students know if this is what they really want
- UCLA CS should be proactive working with the **Academic Advancement Program**
  - More CS PLFs would be extremely beneficial for retention
  - Professors should be reminding their students who are eligible to join AAP to do it, remind students that AAP is a resource for them, and encourage them to become PLFs after taking the course
- **Transparency**
  - Publicly release reports on our diversity / equity / inclusion, actions being taken on departmental level, results of those actions
  - Transparency about the role of the Diversity Dean, Standing Committees, Liaison panels, mentor professor program, etc.
- Encourage **feedback**
  - Incorporate more opportunities to provide anonymous feedback
  - Highlight changes made due to student input
  - Openly mention that professional feedback is encouraged
  - Low-friction mechanisms to provide feedback
- **Outreach**
  - Better efforts to reach groups along the pipeline

    - UCLA Engineering can come to Title 1 schools in LA -- do more direct outreach, emphasize availability of financial aid, tell students they don't need to already know how to code, etc.
- Being aware of **microaggressions**
    - Acknowledging and explaining the historical roots of certain concepts/terminology such as “master-slave” and electing to use different terminology that has industry acceptance
    - Similar comments for the matching algorithm, etc.

Academic Curriculum

- Possible revamp of the curriculum: **change the ratio of required courses to electives** to allow students to take more elective courses.
    - Students want more flexibility in choosing classes more applicable to their industry or research interests (many students voiced that the requirement of CS M152A in particular was not of relevance)
    - Rebalance major requirements to include potentially fewer required courses and more electives
- Make **USIE courses more accessible** to students
    - List out specific expectations, instead of official prerequisites, to expand breadth of USIE courses that require certain prior knowledge
- Work with the university to **improve availability of recorded in-person lectures**
    - Professors with slides may be able to record audio over those slides
    - Other larger classes taught on the whiteboard require human operator; work with university to expand this capability to more lecture halls
- Make petition process for creating **custom sci-tech and tech breadth** more transparent
    - Approved pathways in the past: Sociology, Communications, Digital Humanities, and Political Sciences, etc.
    - Make it clear to students that Prof. Korf and an OASA counselor review the petition and are willing to talk to students to understand their interests better
- Have conversation among instructors about **unreasonable workload in some CS courses**, such as CS 111 and 131
    - CS field has grown, whereas CS curriculum hours have not grown
    - Overwhelming workload gates certain students from the major, affects students’ mental health and well-being

  - Reevaluate curriculum structure (e.g. break up a heavy course such as 111 over two quarters to help students absorb the important material better)
  - Encourage all professors to incorporate checkpoints and grading scripts in their projects to help minimize academic dishonesty in these classes
- Encourage professors potentially to include **office hours right after class**
  - OH right after class seem to be better attended, as students are already there and might as well stick around if they can
  - Encourage students and professors to broaden office hours not just for class help but also to chat about outside things, such as industry, imposter syndrome, etc.
- Establish **more formal dialogue with alumni from industry** to get input on updating course material to keep up with the times
  - e.g. CS 133 was created in response to demand from industry
  - Some professors update projects based on industry happenings and talking with previous students and alumni
- Encourage professors to send out **mid-quarter feedback** to get input from students

Academic Honesty

- Set **transparent expectations**
  - Make it very clear is what collaboration is allowed and what is not allowed
- Re-evaluate **grading schemes**
  - Grading curves encourage unhealthy, toxic competition and cheating
  - Suggestion from Dean Santos: Implement a grading rubric based on foundational points and not against other students. Have students offer more written explanations of understanding and opportunities to do custom projects.
- Provide **more support** for projects
  - Incorporate skeleton code so that students have an idea of where to start
  - Incorporate checkpoints and grading scripts in projects so students have an idea of what their grade will be
  - Provide enough resources and examples in class to complete the project
  - Ensure that enough time is given for students to complete the projects (given both the pacing of the class and workload/responsibilities of students)
- Incorporate student feedback on ensuring **course units** are reflective of workload
- Ask students regularly for **feedback on assignments** to gauge difficulty
  - Anecdote from Professor John Cho:

- I did something along this line this quarter because I haven’t taught 143 for a while. I put out a Google Form and asked how they did. Then after project 3, some students came by and asked how many more projects are there and how many hours are we supposed to spend. And I got this feeling that there’s too many projects and saw some of these sentiments on Piazza as well. Then I sent out a Google Form asking how long they spent on the first 3 projects and if they want more/fewer. Then I realized students are taking more time than I expected so I cut off the last project and extended the last 2.

To Share with Students

- Professor Quanquan Gu: There are some students, including my own grad students, who are building models for COVID and we put all the code on our project GitHub. If that might be an interesting resource, I’d be happy to point you to the resource so students can play with the code that’s updated on a weekly basis and you can see how ML/AI can be used to solve some real world problems like COVID forecast and prediction.
- Alina Haas: New Student & Transition Programs gives one day of New Student Orientation for transfer students and three days for incoming freshmen. If students want to leave feedback, they can connect with her office, and her office can get students connected with them.

CONTENT SUMMARY

Diversity

- What actions or initiatives have been or are currently being taken on a **departmental level** to address diversity, equity, and inclusion in CS? What is the impact?
    - Faculty recruitment on "mentor-professor" positions
    - Fiat lux on imposter phenomenon
    - "Awareness to action" modules on gender and race bias (deployed to 100+ students so far, more regular in earlier classes)
    - CS 30 in response to some of these issues, part of application to join BRAID
- How have you personally taken action to learn about diversity and equity issues and ensure inclusion **in your classrooms**?
    - Courses / Trainings / Workshops
        - Workshops by Center for Teaching
        - Courses on implicit bias training
        - Trainings on suicide prevention
    - TA training seminar: mental health awareness, refer students to CAPS
    - New professors: empathy training and resources
    - Special program for CAPS for engineers: expedited appointments
    - No demographic info on freshman admissions: Prop 209
    - Summer institute for high school students
        - Pipeline: opportunities for underrepresented high school students
    - Partner with CEED for summer research program, admitted high school students with diverse backgrounds
    - Engineering School level: new standing committee on diversity
- The UCEE report for this year indicated that **Computer Science regressed in minority and female enrollment**. We are below other UC schools of similar calibre and other engineering majors. What is the computer science department doing to remedy the gap in the pipeline besides just department wide efforts like CEED?
    - Can look at trends, but there may not have direct levers in the admissions process to remedy. Needs to be done in terms of outreach.
    - Focus on pipelines through middle and high schools (high school counselors)
    - UCLA doesn't require SAT anymore, this helped

- How will the computer science department address the Gafni incident? Why is our department chair not here today, especially considering his absence from town hall last year as well?
    - Little known info
- How is the department supporting **LGBTQIA+ students**?
    - Not aware of any specific programs addressing this group
    - Prof. Santos and Prof. Nachenberg have open door policies
    - QWER Hacks support
- What are some ways the department can help **support students or include LGBTQ+** in efforts of inclusion and diversity?
    - Worthwhile that faculty is aware that everyone has different experiences and should not focus on one particular normative, even in technical aspects
    - Celebrate and encourage more of what student groups do, we can do more
    - Would like to hear more from students
    - Complex, they're trying to be as inclusive as possible
- Transition efforts of the imposter syndrome for freshmen is made. But transfers are “behind” on catching up on peers. What can be done so **transfer students** can have this transition that freshmen students have?
    - Special orientation for transfer students on 2 levels: after recruiting admissions and also for orientations
    - Gaps that need to be filled: hard for transfers to know they're sufficiently prepared
    - Real problem for transfer students who came last year and are graduating this year. COVID is part of the problem, but also we haven't done a great job handling it
    - Orientation days are set by the campus - not sure about why new student transition program is 1 day for transfers and 3 days for freshmen

Academic Curriculum

- Are there any talks about changing the **ratio of required courses to electives**? Many students have expressed interest in taking more elective courses to cater to their interests.
    - Probably due for a major revamp of the curriculum
    - Trade-off between required courses and electives
    - Can sympathize that students want more electives, but there is value in having some required courses that faculty say every student should know
- Many other universities offer **student-led courses for credit**, such as UC Berkeley's DeCal program, which offers 1-2 unit classes that cover more modern topics, allowing for students to explore these interests. Is the CS Department open to running a similar program?
    - USIE program: one year course, teach for 1 unit
        - Note: this program requires that the classes offered have no prereqs. This prevents USIE courses from delving into more complex CS offerings that require prereqs such as CS 31, 32
        - → May be a way around this: listening out what we expect without listing official prereqs
    - 88S courses: students teaching and students taking get credit
    - LAs can deal with more practical parts of classes
- Some students (esp. non-traditional) find **recorded lectures** to be very helpful to their studies. Does the department plan to continue after COVID?
    - Possibly, will take resources
    - Issue: not all classes are equipped to record
    - Profs using slides might be easier: record audio over the slides
    - University already offers recording for the largest classes every quarter, but need human operators
- The intent of the **sci-tech and tech breadth** is to take courses outside CS. Could we have **more non-stem electives** or if we petition, could we make that petition process more transparent because it takes a long time and you don't know what will be accepted.
    - Sci-tech is for science and tech courses, tech breadth is outside CS
    - Open to receiving petitions for more novel combinations
    - Sci-tech petitions: Prof. Korf reads with OASA counselor, makes decision
    - Have had classes approved in Sociology, Communications, Digital Humanities, Political Sciences, etc.

- What **classes do you imagine UCLA offering in the near future** (for applicability). For example, Berkeley made their intro courses in Python and have students submit through github to get them familiar with that.
    - We're probably due for a major overhaul of curriculum, will take time
    - New courses: try to offer on a regular basis, they start as 188 first
        - Fundamentals of Data Science, Computer Vision
        - Looking to offer a new course in NLP
        - Did HCI before, but profs no longer available to teach it
    - Python in intro courses in a different order (35L)
        - 35L revamp includes JS, Node, React
- In my experience, many of the CS classes I've taken, especially 111 and 131 have pretty **unreasonable workloads**. I've had to spend 20-40 hours on assignments on top of lectures and other stuff. It potentially gates certain groups from the major and can affect student mental health? What is being done to make workload more manageable courses like these?
    - Prof. Korf needs to talk to instructors of those courses to see what can be done
    - We also have data from student evaluations
        - Classes in question: 111, 131, 35L
    - CS field has grown but CS curriculum hours has not
    - Trade-off: teach less or have greater workload. Needs curriculum revamp.
- Are there plans to have smaller class sizes / a way to have a **personal connection with professors**?
    - Can't do smaller class sizes, we don't have the resources.
    - Model has always had a lecture and discussion split, discussion provide smaller classrooms
    - Connecting with faculty: all faculty expected to have 2 officers hours a week
        - Maybe we can broaden office hours to not just get help but chat with professors
        - Can also email professors, use faculty advising
- Is there any cooperation between the **department and representatives from the private software industry** in curriculum? If so, what is the extent?
    - CS 130: taught by Google software engineers
    - Alumni Advisory Board: get feedback on our curriculum
    - Hiring outside lecturers
    - Curriculum sometimes influenced based on informal discussions with folks in industry, under the radar

        - Input from alumni and previous students also helps
    - CS 133 created as a response to demand from industry
        - Evolved: GPUs, IPGAs, Amazon Cloud

Academic Honesty

- **Why** do professors believe that students cheat?
    - Time pressure. Would love to know why students do it, hear suggestions
    - As faculty: make clear what collaboration is allowed and what isn't
    - Encourage faculty to not grade on a curve
        - Have gradic rubric based on foundational points, not against students
    - Have students offer more written explanations of understanding
    - Offer students opportunity to do custom projects
        - CS 97: lessens cheating
    - Impression: cheating more common in remote exams
- A common complaint is that a lot of homework and projects are graded too strictly and harshly, which leads students to rely on cheating. To what extent can we **incentivize learning over getting good grades** to potentially decrease the likelihood of cheating?
    - COVID has turned this upside down
    - Standard model: grade based on homework + exams
        - Would like to have grade based on homework alone, hard to tell if students are doing it on their own
        - [Prof Korf] Did no exams in Fall
    - [Prof. Cho] Gradescope didn't work well
    - [Prof. Cho] Release shell script that packages files students need, has small shell script that runs basic results, reduces testing
- Is there a possibility for there to be **skeleton code** for projects in CS35L/CS111/CS131? Project specs are hard to follow as a first-time learner. This could also reduce student's usage of github.
    - [Prof. Eggert] Will look into it, in the past has found it to be boring
    - [Prof. Korf] Try to make clear specs, but want students to struggle a bit so they understand the project
    - Large projects / new projects: hard to get specs the first time
    - Tradeoff: refining homework to make it polished + right difficulty vs. make news ones to reduce cheating
- Are professors willing to **incorporate feedback after projects**? This could provide professors with feedback about the true difficulty of the projects and also students other responsibilities in life?
    - [Prof. Korf]Interesting idea, will look into it. Perhaps submitting anonymously

    - [Prof. Cho] Put out Google Form after each, saw that there were too many projects and students taking longer than anticipated
        - After a while, can estimate how long it takes for similar projects
- Would the department be willing to incorporate feedback on some courses having **units that do not accurately reflect the workload** of some courses?
    - Yes. Department would be open to constructive feedback from students
    - Really value student's thoughts, need feedback
- Can professors **offer project redos** if students get lower than a certain score? I think this could decrease cheating.
    - [Prof. Korf] Difficulty: professors sometimes give solution to students after submitting
        - How to disincentivize students from not looking at the solutions
    - [Prof. Cho] They need to run the test script. If they don't run this and get a low score, then no regrading. Details are important.
    - [Prof. Gafni] Focus on principles, not details. Details will come with the job, and not important for preparing students for grad school.