Research Statement

Leah Perlmutter

Often, it is possible to make technology that addresses problems in the world related to access, inclusion, and support. A lot of my work does this; however making technology for good is not enough. Too often, the sphere of technology instigates problems related to access, inclusion, and support. For instance, the toxic culture of mainstream CS academic environments fails to support and include all participants equitably, which limits who becomes tech makers, and in turn, the identities of tech makers influence what kind of technology they choose to create. I want to both make technology that does good in the world and improve CS education so that a more diverse group of people can become tech makers, and can themselves create technology that more people benefit from. Through a broad corpus of work in HCI, Robotics, Accessibility, and CS Education, I explore the design and use of computer science technology and instruction for access, inclusion, and support.

Outlook on undergraduate research

My outlook on undergraduate research stems from my experiences as an undergraduate researcher at Colby College, and from my experiences as a graduate researcher at UW coordinating a research team with several undergraduates, and mentoring an undergraduate researcher. In alignment with [1], I see early research experiences as a way to engage and retain a diverse group of students. Accordingly, I prioritize exposure to research methods and practices over output produced.

My future work on college level CS Education is well suited to a diverse group of undergraduate researchers. It is approachable to undergraduates because it has low barriers to entry. Like quantitative methods in CS, my qualitative methods, such as interviewing and thematic analysis, require training and practice to acquire the skill needed for quality work. However, there are no prerequisite courses, and new researchers can get started with their role in conducting research in tandem with learning the methods from scratch. CS Education is likely to engage undergraduate students and set them up for success because they're direct stakeholders. Many undergraduates attend to educational systems, since they're in one, and they may wish to help improve the system for students who come after them. Since the population being studied is their own peer group, they are likely to have good intuition about undergraduate CS culture, leading to astute research questions and insightful analyses of data. I wish to produce research findings to transform CS education systems to benefit groups that are currently marginalized in CS. The ability to do that depends directly on problem posing, research design, and analysis by researchers who identify as members of those groups, hence a diverse group of undergraduate research mentees is vital to this work. I plan to select research assistants based on potential, not experience in CS or in research, because experience level upon entering college is strongly connected with privilege and is likely to bias toward White and Asian males while excluding women and non-binary people, as well as Black, Indigenous, and People of Color and disabled prospective researchers.

Increasing access to mental health skills

Dialectical Behavior Therapy (DBT) is a skills-based therapy proven effective as treatment for teens experiencing anxiety, depression, and suicidality, but has access barriers because it requires a multi-month, time-intensive commitment and is expensive. My work on DBT-based micro-interventions aims to offer teens a way to learn some DBT skills without engaging in a full DBT therapy program. In this interdisciplinary work connecting Psychology with Computer Science, we designed and implemented an app as a micro-intervention to help users acquire *emotional clarity*, which is part of Emotion Regulation, one of the four main skills taught in DBT. We built a web app[1] using javascript that helps users to gain emotional clarity, that is, to label their emotions and identify the source of each emotion, based on information from the DBT Skills Training Manual [5]. We evaluated the app in a pilot study where 16/20 participants agreed or strongly agreed with the statement, "The app helped me better understand my emotion(s)."

While working on the emotional clarity app, I mentored an undergraduate student, Rachel Sitt. Rachel was initially interested in frontend coding, and open to new experiences and other research tasks. I made sure that Rachel

[1] https://github.com/lrperlmu/emotional_clarity

was paid for her work and helped her to stay engaged by assigning concrete weekly tasks including coding and reading research papers, based on her capacity given responsibilities outside of research. During our weekly meeting time, we focused on building Rachel's skills. I performed code reviews to help with her programming skills and taught her about version control with git and project management with github issues. We also discussed paper reading strategies as well as take-aways from the papers she read. At the end of our time working together, Rachel expressed gratitude with a note saying, "You've been there for me through every line of code, every implementation detail, and every single time I felt lost in the grand scheme of the CS life." Rachel has gone on to become a Software Engineer at Microsoft.

Supporting novice users of personal robots

In the rising field of end-user robotics, supporting novice users is important so that robots can be usable to a broader range of people. When a human is commanding or collaborating with a robot, it's important that the human and robot be able to communicate about objects in physical space. *Mental model discrepancy* describes the situation where the human and the robot have a different understanding of the physical environment. This may especially be the case with novice users. In [8] I led a research team to address mental model discrepancy with *transparency*, that is, by having the robot communicate what it knows about its environment so that the human can understand its mental model.

I built a system involving a personal robot situated in a model living room environment that can understand which target object a person is verbally describing and/or pointing at. The robot uses *transparency* mechanisms to reflect its understanding back to the user: it speaks a description of the object, gazes toward the object, and uses an arm to point to the object (human-like transparency). It can also display a model of the environment with the target object highlighted, either on a computer monitor or a VR headset (visual transparency). In a study adding visual transparency significantly improved the speed and accuracy with which participants commanded the robot, compared to a baseline with only human-like transparency. With visual transparency, participants changed their behavior to adapt to the robot's mental model by using the numeric object labels that the robot displayed in the visualization.

As a continuation of this work, I supported a collaborator in exploring how robots can optimize the use of projected arrows to point out physical objects to a person [11]. When robots can be transparent about their mental models, through human-like communication such as speech and gestures, and uniquely robotic communication such as visualizations and projections, it supports end users and makes robot technology more usable to non-experts.

Increasing access to information

When people learn more languages, they gain access to more information. Traditional classroom learning has barriers such as cost and rigid time commitments that may be bypassed through electronic learning. Learning with robots is a promising option because there are demonstrated cognitive learning benefits when material is taught by embodied, autonomous robots compared to computer-based lessons. One major challenge is creating lesson plans to be taught by a robot.

In [7], I coordinated a team of 11 researchers, including 4 undergraduate students, to develop a robot language tutoring system. We introduced *content mapping*, a novel strategy for generating lesson material, in which we drew upon an existing corpus of electronic lesson material and developed a mapping from the original material to the robot lesson, greatly reducing the time and effort required to create robot lessons. We built a system, KubiLingo, to implement content mapping for language lessons. With permission, we used Duolingo, a popular online language learning platform, as the source of our content. In a mixed methods study, user performance improved by a statistically similar amount with a robot lesson as with Duolingo lesson. Participants reported that KubiLingo was more distracting and less likable than Duolingo, indicating the need for improvements to the robot's design.

In this study, I hired undergraduate research assistants for a limited but important and specific task, running the user study. I made sure that they had fair wages, access to concrete instructions in the form of a study protocol, plenty of training in the form of study pilots, and a consistent and responsive point person to check in with.

Making accessible mobile technology

To be accessible to blind and low vision users, mobile technology relies on screen readers, which are highly versatile and effective tools that transform the features of a visual interface into speech, but they are limited by the

limitations of the visual interface. Small targets such as the buttons in a calculator app, though they can be accessed using a screen reader, are designed with sighted users in mind. Targeted key presses are time-consuming and error-prone for many screen reader users.

Another way to design accessible apps for blind and low vision users is to forego targets altogether. I helped to design GestureCalc, a digital calculator that uses target-free gestures for arithmetic tasks [3]. It allows eyes-free target-less input of digits and operations through taps and directional swipes with one to three fingers, guided by minimal audio feedback. I designed and conducted a mixed methods longitudinal study with eight screen reader users. Over three sessions, participants learned the GestureCalc gestures well enough to efficiently perform short calculations. With GestureCalc, they typed faster and made fewer erroneous calculations than with a screen reader. In interviews, participants recognized the effort in learning a new gesture set, yet reported confidence in their ability to become fluent with practice.

Understanding belonging in CS to inform inclusive practices

Recently, I have conducted qualitative research in CS Education, pertaining to student experiences of resubmission and identity formation. I co-led a research team that interviewed students about their experiences with resubmissions and their formation of a CS identity, after they took CS1 and CS2 courses with a new resubmission system for programming homeworks [6]. Interviewees indicated that resubmissions take the pressure off and enable them to achieve high grades incrementally, but our findings also suggest that resubmissions alone do not create space for CS identity and belonging. This finding motivated me to dig deeper into investigating what *does* promote belonging for students.

My current research is on student sense of belonging in college CS spaces, and the influence that TAs have on that. In a large university CS program TAs outnumber instructors and have the majority of contact hours with students. Being in a position of power makes TAs well placed to shape the culture of CS and to influence students' belonging and identity formation, for better or for worse. I conducted semi-structured 1:1 interviews with 14 participants, consisting of 7 pairs of students and TAs who interacted in CS office hours at UW and performed a preliminary thematic analysis of the transcripts. TAs and students both reported that TAs helped students feel a sense of belonging by treating them with respect, fostering understanding of the material, and helping them meet peers.

In ongoing analysis of the interview transcripts, I am exploring nuances of each participant's identity toward their experience in CS. For example, whether they are a TA or student, their ethnicity, gender, first generation status, and prior CS experience, and how all these identities influence their perspective on belonging.

Future work

Through my future research, I want to develop and implement ways to support CS students and build a CS culture more hospitable to students from historically marginalized groups, including by identifying small college practices that are transferable to large universities. In collaboration with undergraduates, I will (1) extend my prior work about supportive emotional reflection apps to a college CS setting, (2) extend my work about TA influence on student belonging in CS and (3) conduct new work about sociotechnical topics in the curriculum and their effects on CS culture.

Supporting CS Students through Emotional Reflection. I want to extend my work on apps for emotional reflection, and I see a college CS department as a good place for this because computing can be inaccurately portrayed as logical and therefore unemotional, but in fact, emotions are an important part of learning in the sciences. For instance, Radoff et al. describe how a student learned to become excited about the scary parts of learning physics, improving her ability to learn, a phenomenon known as developing *productive meta-epistemic affect* [9]. An app for reflecting on emotions could help CS students to develop productive meta-epistemic affect as a practical response to the frustrations of programming. Another purpose for an emotional reflection app could be helping students to feel less isolated in CS. In my belonging research, students reported that knowing others shared their struggles helped them feel a greater sense of belonging. In the face of a competitive CS culture that can be isolating, an app to anonymously share emotions could support students in feeling validated and not feeling alone by seeing that others share their situation. It is important to distinguish between student emotions that arise as a natural part of learning versus those emotions that

arise from systemic injustice in the education system. Therefore, the apps could also offer students guidance to distinguish between the two, and resources for reporting systemic issues.

Understanding CS Identity and Belonging to Improve Inclusivity. With the goal of building a CS culture more hospitable to students from marginalized groups, I'm interested in understanding the CS identity formation among students who study CS for the first time in college. Since students judge their fit in CS based on value alignment [4], the current culture turns away students who care about social issues or who don't buy in to hierarchical communities; thus, constructing a CS learning environment where such students can thrive is both a means to and an end goal of collective culture change. One big question is how to empower these students to develop a CS identity authentically, without compromising their identity to assimilate into the existing CS culture or being repelled from CS because of a culture mismatch. I hope to extend my belonging work from UW across more institutions, especially small colleges. Small, undergraduate serving colleges are positioned to lead the way with inclusive culture, including in ways that large universities could adopt. Some research questions include, which inclusivity practices can be propagated from one institution to another? Which are specific to a small college or large university setting? Identifying existing, less-toxic microcultures of CS, for example at small colleges, could help develop ways to change mainstream CS culture.

Understanding the Impacts of Sociotechnical Lessons on CS Culture. One idea for culture change is teaching sociotechnical topics embedded in the curriculum, such as race, gender, and class in technology, and CS ethics. Both embedded ethics and standalone sociotechnical courses are gaining traction in college and university CS programs. But can sociotechnical content change the culture of CS? I am hopeful that it can. I'd like to understand the longitudinal impacts of a sociotechnical curriculum on students and beyond. Possible research questions include the quantitative, for example, how does a sociotechnical curriculum affect the demographics of enrollment, retention, and graduation in undergrad CS? But, students are more than their demographic groups -- CS identity is negotiated and constructed [10], a process unique for each student. Qualitative research questions can address this, such as, how does a sociotechnical curriculum influence development of CS identity and belonging for students from marginalized groups? How does a sociotechnical curriculum change what individual students think it means to be a computer scientist? Going farther, how does a sociotechnical curriculum influence CS culture in an academic department, or at a tech company populated by students who have graduated from CS programs, or in the tech industry as a whole?

References

[1] Alvarado, C., & Dodds, Z. (2010). Women in CS: An evaluation of three promising practices. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education - SIGCSE '10*, 57. https://doi.org/10.1145/1734263.1734281

[2] Björkman, C. (2005). Crossing Boundaries, Focusing Foundations, Trying Translations: Feminist Technoscience Strategies in Computer Science. http://urn.kb.se/resolve?urn=urn:nbn:se:bth-00296

[3] Chaudhuri, B., **Perlmutter, L**., Petelka, J., Garrison, P., Fogarty, J., Wobbrock, J. O., & Ladner, R. E. (2019). GestureCalc: An eyes-free calculator for touch screens. ASSETS 2019, 112–123. https://doi.org/10.1145/3308561.3353783

[4] Lewis, C., Bruno, P., Raygoza, J., & Wang, J. (2019). Alignment of Goals and Perceptions of Computing Predicts Students' Sense of Belonging in Computing. Proceedings of the 2019 ACM Conference on International Computing Education Research, 11–19. https://doi.org/10.1145/3291279.3339426

[5] Linehan, Marsha M. 2014. *DBT Skills Training Handouts and Worksheets, Second Edition.* Guilford Publications.

[6] Perlmutter, L., Everson, J., Yasuhara, K., Wortzman, B., & Lin, K. (2022). Reading Between the Lines: Student Experiences of Resubmission in an Introductory CS Course. SIGCSE 2022. https://doi.org/10.1145/3478432.3499112.

[7] Perlmutter, L., Fiannaca, A., Kernfeld, E., Anand, S., Arnold, L., & Cakmak, M. (2016, November). Automatic Adaptation of Online Language Lessons for Robot Tutoring. ICSR 2016. https://doi.org/10.1007/978-3-319-47437-3_65.

[8] Perlmutter, L., Kernfeld, E., & Cakmak, M. (2016, June). Situated Language Understanding with Human-like and Visualization-Based Transparency. Proceedings of Robotics: Science and Systems. https://doi.org/10.15607/RSS.2016.XII.040

[9] Radoff, Jennifer, Lama Ziad Jaber, and David Hammer. 2019. "'It's Scary but It's Also Exciting': Evidence of Meta-Affective Learning in Science." *Cognition and Instruction* 37 (1): 73–92. https://doi.org/10.1080/07370008.2018.1539737.

[10] Rodriguez, S. L., & Lehman, K. (2017). Developing the next generation of diverse computer scientists: The need for enhanced, intersectional computing identity theory. Computer Science Education, 27(3–4), 229–247. https://doi.org/10.1080/08993408.2018.1457899

[11] Weng, T., **Perlmutter, L.** Nikolaidis, S., Srinivasa, S., Cakmak, M. Robot Object Referencing through Legible Situated Projections. In ICRA 2019, 8004–10. https://doi.org/10.1109/ICRA.2019.8793638.