RESEARCH STATEMENT

My research goal is to reimagine educational programming environments as adaptive, data-driven systems that support students automatically as they pursue learning goals that are meaningful to them. To do so, I have designed algorithms to automatically generate programming hints and feedback using student data [C6,C9], created the first block-based programming environment to offer data-driven hints [C8] and investigated how students seek and use help when programming [C10,C11]. My research spans research communities and top-tier academic conferences, including Artificial Intelligence in Education (AIED), Educational Data Mining (EDM), and the International Computing Education Research Conference (ICER).

Motivation. Growing national interest in Computer Science (CS) education has fueled efforts to recruit a larger and more diverse group of computing students. These efforts frame programming as a way to create meaningful, creative and impactful artifacts. This approach has resonated with learners, and the number of undergraduate CS majors has tripled in the past decade [1]. However, CS can be a notoriously unforgiving subject, with the highest 4-year college attrition rate of any STEM field [2] and CS1 failure rates of 30-40% [3]. Students are struggling and need more support, and instructors alone cannot provide it. This is especially true in the current educational landscape, with overcrowded college classrooms [4], thousands of newly-trained K12 CS teachers [5], and the growing popularity of MOOCS and informal learning settings. My research addresses these challenges by developing novel, data-driven programming environments that support students automatically, reducing the burden on instructors and leading students to better learning outcomes.

Foundational Work: iSnap

The first milestone in my research has been the design and evaluation of iSnap [C8], a block-based programming environment that provides students with on-demand hints and feedback. iSnap is an extension of Snap!, a popular novice programming environment, which allows users to easily create meaningful and creative computational artifacts, such as games and scientific simulations. iSnap combines the approachability of Snap! with the AI- and data-driven support features found in Intelligent Tutoring Systems (ITSs), state-of-the-art learning environments that offer students feedback and facilitate individualized learning. My research with iSnap has produced three primary contributions: generating data-driven hints and feedback, supporting block-based, exploratory programming, and understanding how novices seek and use help when programming.

Generating data-driven hints and feedback. A hallmark feature of ITSs is their ability to provide on-demand hints to struggling students, with the goal of resolving their confusion, leading to learning rather than frustration. Traditionally, these hints are generated using expert-engineered domain models, which are time-intensive to produce. To avoid this cost, iSnap uses a novel, data-driven algorithm to generate hints using previous students' solutions. It matches the struggling student to a previous, successful student and offers recommendations using that student’s correct solution [C6]. Effectively matching students is particularly difficult in the domain of programming, due to the virtually infinite space of possible solutions to most problems. To address this, iSnap uses a novel distance metric designed to identify structural similarities in students’ code [C9]. Because iSnap has been deployed in classrooms since its creation, it is critical that it generate high-quality hints. I have evaluated the quality of iSnap’s hints using human ratings [C10] and by comparing them against human-authored hints [C9]. I found that iSnap produced the same feedback as a human tutor almost as frequently as a different human did, though it also produced additional suggestions that humans did not. I am currently applying this method to benchmark and compare the growing number of

1 Bold references refer to CV publications list.
programming hint generation algorithms on a standard dataset to encourage the research community to adopt higher standards for data-driven hint quality.

Figure 1. iSnap annotates a student’s code with suggestions. A student can request a next-step hint by clicking on a blue-outlined input slot or plus button, and iSnap will display a suggested edit. iSnap also highlights possible errors, such as the ”pen down” block, which should be moved.

Supporting block-based, exploratory programming. iSnap is the first block-based programming environment to offer data-driven support. Block-based environments allow users to construct a program by dragging together blocks of code, which can only be arranged in syntactically valid ways. The choice to support block-based programming grew out of my previous work evaluating its efficacy. I compared block and text interfaces in otherwise identical programming environments and found that middle school students completed significantly more of a short programming activity with the block interface, working more efficiently and spending less time idle [C2]. iSnap also supports the Beauty and Joy of Computing (BJC), an AP CS Principles curriculum built around Snap!. BJC is used by thousands of high school and undergraduate students, and my analysis of student feedback shows that it appeals well to diverse students [C3]. This gives iSnap real potential for classroom impact, but the exploratory nature of BJC’s programming assignments has also required iSnap to overcome significant algorithmic challenges for hint generation, since hints must support many design choices and opportunities for personalization [C6].

Understanding how novices seek and use programming help. Programming support, even if well designed and pedagogically useful, does little good if students do not use it. My work with iSnap shows that in programming, as in other domains, students often practice poor help-seeking behaviors, avoiding help when they are struggling, or abusing help to complete problems without trying [C8,C10]. I investigated the factors that influence help-seeking behavior with a qualitative study of students who completed two programming assignments, with human-tutor and iSnap help, respectively. The results suggest that while students’ desire for independence is a strong deterrent to help-seeking, this effect may not be as strong with computer-based help [C11]. It therefore may be possible to overcome some help-avoidance by presenting help as a ”tool” that savvy programmers use, rather than a tutor-like support system.
**Research Agenda**

My research goal is to build and evaluate programming environments that automatically support students in creating meaningful computational artifacts. I want to empower the growing body of educators who present CS in ways that connect with students’ interests (music, videogames), allow for personal expression (storytelling, digital art) and have clear real-world applications (data science, simulations). These **creative, open-ended projects** are challenging, and students require individualized support to be successful. My research lab will design the adaptive, data-driven systems to provide it. By enabling learning that is well-scaffolded and connected to students’ lives, I aim to **improve retention and learning** in CS classrooms and **make computing more accessible** to diverse students. Supporting these creative, open-ended programming projects is an ambitious undertaking that cannot be accomplished with current data-driven methods. My research agenda develops the foundational components that will make this possible:

**Exploring the Design Space of Programming Help.** Current data-driven support for programming focuses almost exclusively on next-step hints that tell a student *what is wrong* with their code or *what to do next* to complete a problem. However, as Suzuki and colleagues show, teaching assistants rarely offer this kind of "bottom-out" hint to students in CS classrooms, demonstrating a need to explore the design space of automatically generated help [6]. My lab will address this need by designing new, higher-level forms of data-driven help, including automatically generated worked examples and self-explanation prompts. These interventions are supported by strong empirical evidence in other domains, and they are especially appropriate for open-ended programming projects, since they can help students without forcing them to solve a problem in a predefined way.

**Negotiating Help with the Student.** Many ITSs can confidently *tell* a student what to do but lack the ability to *ask* a student what they *want* to do. In open-ended programming, this feature is imperative, since students can deeply personalize their solutions. Data-driven support may be able to identify numerous potential errors or next steps for a student, based on previous students’ solutions, but there will always be uncertainty as to which of these suggestions are applicable [C10]. My lab will address this challenge by designing novel, mixed-initiative help interfaces that work with students to identify problems and formulate appropriate solutions.

**Student Modeling for Open-Ended Programming.** Modeling a student’s knowledge allows a learning environment to tailor its resources and support to a student’s individual needs. These models typically require students to work on repetitive tasks that test a set of predefined skills, one at a time. Recently, Yudelson and colleagues have built student models for more complex programming tasks, automatically identifying skills from student code [7]. My lab will extend this work to model student knowledge in open-ended programming projects, where students practice many interdependent skills at once. I will use the resulting student model to adapt the support features described above to provide the right amount of scaffolding for each student.

**Application and Evaluation in the Classroom.** My research agenda will center on the design of usable systems for real-world classrooms and informal learning settings. My work will pair the design and implementation of new technologies with rigorous evaluations, both in the lab and in the classroom. Effective evaluation combines controlled studies to assess student outcomes with detailed qualitative investigations of how students use help features. These results then inform the iterative design of the features themselves and shape the theories that underpin them.

**Research Philosophy**

I believe that research must have the ultimate goal of positively impacting individuals’ lives, and I do so by addressing real-world problems in education. I tackle these problems with interventions that
are grounded in appropriate cognitive and educational theories, and I rigorously evaluate them with data mining and experimental studies. Tackling meaningful problems in education has required me to collaborate across institutions and disciplines [C4,C7,A5,A6]. Much of my work has focused on bridging the ITS/EDM and CER communities, and I have hosted conference discussions [A6], written position papers [W4] and participated in working groups, making the case that the most effective educational technologies will stem from both communities working together.

My vision for research centers around creating a cohesive research lab, recruiting promising PhD students and providing them with the support they need to develop their potential. While my research agenda will guide my students, I expect that their interests and expertise will in turn shape that agenda, including the funding we seek. My research has always focused on building usable educational systems, with clear target populations, which provide opportunities for newer lab members to quickly design and implement experiments. I have mentored two undergraduates as they developed contributions to iSnap, and I would similarly encourage undergraduates to work in my lab. They can gain useful experience developing research systems, while freeing up graduate students to focus more on research activities.

I have experience obtaining funding for innovative research, primarily from the National Science Foundation. I have designed and co-authored a successful grant application to the NSF’s Cyberlearning and Future Technologies initiative to develop and evaluate iSnap (#1623470; $550K). I plan to target similar funding opportunities at the intersection of computing education and advanced learning technologies, including Improving Undergraduate STEM Education (IUSE), Promoting Research and Innovation in Methodologies for Evaluation (PRIME), and STEM+Computing Partnerships. For each of these initiatives, I have helped to revise successful grant proposals. I will also seek early funding for myself and my students by applying for the NSF CAREER and CRII awards and by working with other faculty to apply for a Research Experience for Undergraduates (REU) site grant.

References