International Society of the Learning Sciences

Teachers' Learning to Support Students During Science Inquiry: Managing Student Uncertainty in a Debugging Context

Jessie Nixon , Utah State University, jessie.nixon@usu.edu
Colin Hennessy Elliott, Drexel University, ch3457@drexel.edu
Michael Schneider, University of Colorado, Boulder, michael.j.schneider@colorado.edu
Jeffrey B. Bush, University of Colorado, Boulder, jeffrey.bush@colorado.edu
Srinjita Bhaduri, University of Colorado, Boulder, srinjita.bhaduri@colorado.edu
Mimi Recker, Utah State University, mimi.recker@usu.edu

Abstract: This paper analyzes two teachers' participation in professional learning (PL) activities designed to help them learn to support students when they face uncertainty during a computationally-rich science inquiry unit and their subsequent enactments of the unit. In this unit, students assemble a physical computing system (PCS) coupled with sensors to program and display streams of environmental data. Students inevitably encountered moments when they were "stuck" and required teacher help. The PL activities consisted of teachers taking on "student-hat/teacher-hat" roles attempting to solve buggy PCSs followed by a discussion. Analyses using a lens on teaching as improvisation illustrated how PL activities helped teachers develop an understanding of the PCS, confidence in enacting the unit, and pedagogical strategies for making in-the-moment decisions to support students facing uncertainties. Analyses also revealed teachers' approaches for managing students' uncertainty level and the improvisation it required: one more constraining and one more expansive.

Introduction

Science education standards in the U.S. highlight several practices that are intended to reflect the work of modern scientists and engineers (NGSS Lead States, 2013). In particular, the Next Generation Science Standards (NGSS) foreground "uncertainty" as a key driver in students' developing scientific knowledge and making sense of the world. Manz and Suarez (2018) argue that "uncertainty is endemic to science" (p. 772) as "scientific practices are largely concerned with managing uncertainty and drawing conclusions in light of it" (Manz & Suárez, 2018, p. 772). For students, these moments of uncertainty arise as they engage in science inquiry when there are often "nonobvious decisions about what to do" (Manz & Suarez, 2018, p. 772). The science standards thus speak to the importance of having teachers help students grapple with uncertainty in inquiry-driven instruction as they orchestrate student learning. For many science teachers, enacting pedagogical practices that effectively support students in managing uncertainty for productive learning remains a challenge.

Moreover, just as modern science inquiry is increasingly driven by computation, so too is science classroom inquiry. Traditional classroom science instruments are being replaced with physical computing systems (PCS), like the BBC micro:bit. A PCS can be programmed and coupled with sensors to gather, process, and display streams of environmental data. By creating these systems, students engage in computational thinking practices that can have deep synergies with their disciplinary science learning.

In the context of these computing-rich inquiry activities, uncertainty often arises during moments when students' artifacts do not function as they intended, such as displaying the wrong unit of measurement (F° vs. C°). When students are unable to make their artifacts work as intended and unable to progress until they have received some kind of help, they are considered "stuck" (Huadong & Brennan, 2019). These moments of being "stuck" add additional layers of complexity for teachers as they must attend to the students' immediate concerns, help them fix their errors, while also support them in increasing their problem-solving strategies and persevering in the face of difficulties. This can be a challenging orchestration task as science teachers also often lack the necessary computing knowledge to troubleshoot and debug such systems (Tsan et al., 2022).

While prior work has examined student troubleshooting of physical computing (e.g., DesPortes & DiSalvo, 2019) and debugging programs in STEM and science contexts (e.g., DeLiema et al., 2022), much less has examined how teachers learn to support students engaged in these processes (Tsan et al., 2022). Even less has examined how teachers support students working with PCSs and engage with uncertainty as they troubleshoot bugs in the hardware (e.g., a faulty wire connection), debug the software (e.g., an incorrect conditional statement), or resolve issues across their interactions (DesPortes & DiSalvo, 2019).

To address this gap, we previously analyzed the in-situ strategies as teachers enacted a science inquiry instructional unit using a sensor-equipped PCS. We examined how teachers supported their students when they assembled their PCS. This analysis led to the design of professional learning (PL) activities focused on helping teachers learn to support students during troubleshooting and debugging (Hennessy Elliot et al., 2022). We implemented these activities as part of a four-day summer professional learning (PL) workshop with middle and high school science teachers. In this PL activity, teachers took turns in the role of a student who is stuck with a buggy system (produced with bugs previously encountered in classroom implementations of the unit) as well as the role of a supporting teacher. Participants in the PL then reflected on their experiences in this activity. The goal of the activities was to help teachers better understand their students' challenges during debugging and develop strategies to help students navigate moments of uncertainty during classroom enactments.

In this paper, we present an analysis of two teachers' participation in this PL. We examine their problem solving during the PL debugging activity, their reflections on this activity, and how their experiences shaped subsequent classroom enactments of the instructional unit. Our research was guided by the question: In what ways did the PL activities help teachers learn about managing their students' uncertainty as they engaged in debugging during a computationally-rich science inquiry unit?

Literature review

As teachers increasingly integrate programming into science classrooms, research has only begun to examine how teachers, often new to programming themselves, support their students during moments of debugging (e.g., DeLiema et al., 2022; Tsan et al., 2022). This perspective is particularly important as teachers' interactions with students are core to how students develop debugging skills and address gaps in understanding (Blikstein & Moghadam, 2019; McCauley et al., 2008). To address this gap in the research, we previously analyzed the in-situ strategies teachers used during a computationally-rich science inquiry unit to support their students when they became stuck while wiring and programming their PCS (Hennessy Elliott et al., 2023). We found that teachers do not necessarily have to be programming experts to effectively support students in debugging their physical computing systems. This work also highlights that debugging moments are highly unique and contextual and therefore require varied approaches on the part of the teacher.

To best prepare teachers for these moments of uncertainty, recent research has explored how professional learning workshops might be designed to help teachers support their students' debugging. For example, DeLiema et al., 2022 demonstrate the effectiveness of having teachers practice debugging instruction in mock classrooms and reflect on video of classroom debugging practices. Similarly, Tsan et al. (2022) describe a debugging model where teachers used a mnemonic device to assist in debugging and reflected on how they would help their students. However, they found that this strategy led to teachers mostly focused on helping students fixing the bug rather than understanding the cause of the bug.

Because learning the skills of debugging is essential to computer science (McCauley et al., 2008), PL opportunities should provide scaffolding for teachers to support students in developing debugging strategies. Strategies can help students when they run into other similar bugs in the future and can help students "handle unforeseen, novel impasses" (DeLiema et al., 2022, p. 7). Debugging can serve as rich learning opportunities as they naturally spark discussions about the learning process that help students develop critical thinking strategies and important thinking processes (Fields et al., 2021). Such thinking strategies and processes are important for teachers to support students as they encounter uncertainty in science and STEM (Manz & Suarez, 2018).

Several researchers have argued that disciplinary learning in science must go beyond a focus on scientific knowledge and practices to include epistemic affect, or learning how to feel like a scientist (e.g., Jaber & Hammer, 2016). Feeling like a scientist involves encountering and learning to grapple with the kinds of uncertainties central to the discipline (Jordan & McDaniel, 2014; Manz & Suarez, 2018). Scientists describe feeling uncertainty not only when modeling and explaining phenomena, but also when working with various tools, measurements, and determining how to analyze their data (Manz & Suarez, 2018). Further, scientists increasingly use computational tools and design processes in their inquiry and must therefore be able to debug the tools they use. Therefore, students engaging in authentic science practices that use computation to produce data should be equipped to debug and redesign tools by drawing upon technical and scientific knowledge (Hardy et al., 2020) when they encounter moments of uncertainty.

In the context of this paper, the scientific tool students used for scientific inquiry was a PCS, which offers an alternative set of introductory activities to engage students in computing (Kafai et al., 2014). PCSs are relatively user-friendly and affordable (Anastopoulou et al., 2012; Blikstein & Moghadam, 2019), but they are a challenge for debugging as bugs can occur in any part of the system: the software (or program), hardware, or the interactions between the two (DesPortes & DiSalvo, 2019).

The process of debugging PCSs used in science classrooms, similar to data collection and analysis, therefore involves encountering moments of uncertainty as students come to decision points where they do not know exactly what to do with the system. Debugging physical computing systems to use in science classrooms is a step in the process of becoming data producers (Hardy et al., 2020) that offers students – and their teachers –

pedagogical experiences with wrestling with uncertainty that they all can build on when making sense of real-world data.

Conceptual framework: Teaching debugging as improvisational

In this paper, we conceptualize teaching as "disciplined improvisation" (Sawyer, 2004), involving a strategic balance of structure and open-endedness. In exploring the integration of physical computing in science classrooms, conceptualizing the improvisation that teachers undertake is particularly important in developing teachers' ability to support student sensemaking exploring the uncertainty of questions without a single answer (Jurow & McFadden, 2011) and position science – and in turn computing – as a practice rather than a set of facts. Through experience and professional learning, teachers develop a shared repertoire of methods and strategies they draw on supporting students wrestling with uncertainty; yet their work requires being flexible enough to react to the changes, shifts, or needs of each moment.

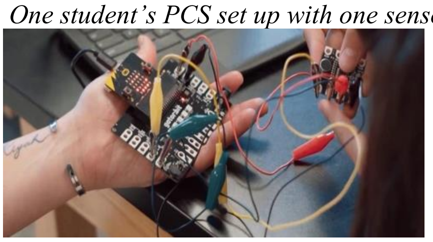
This framing affords a conceptual lens on what decisions teachers make in-the-moment to support students as they debug their PCSs and what strategies and/or structures they draw on – possibly developed in professional learning activities – to make those decisions. Improvisation, whether in art or in teaching, does not mean that "anything goes" (Sawyer, 2004); rather, improvisation has its own structures and methods (DeZutter, 2011; Halverson, 2021). Considering teaching as improvisational offers a window into how effective teaching involves authentically listening to students and facilitating students to build on each other's ideas (Phillip, 2019). Phillip (2019) describes further that when done well, teaching as improvisation involves, as a classroom community, crafting critical questions that examine issues of power in relation to historical, social, political and economic processes. Science teachers, as Jurow and McFadden (2011) argue, must make moves to set the stage by providing "slots" for students to participate in disciplinary practices using student contributions to extend the classroom community's scientific thinking.

Instructional unit and professional learning activities

This study is part of a larger multi-year, multi-site design-based implementation research project that is co-designing NGSS-aligned instructional units and PL activities that use the PCS in service of science inquiry (Biddy et al., 2021). The instructional unit in this paper introduces secondary school students to physical computing as a tool to support sensor-driven inquiry in science and STEM classes. The instructional unit uses a storyline approach (Reiser et al., 2021), where the lesson is grounded in students' generating questions about particular phenomena, planning and conducting investigations to address their questions, and creating models to explain findings.

In the storyline unit, students explore the phenomenon of the PCS that collects data streams such as classroom sound levels, local environmental conditions, or soil moisture levels in classroom plants. During this unit, students investigate the capabilities of the technology by wiring and programming physical data displays that respond differently based on data stream values (see Figure 1). The data displays incorporate a micro:bit which is wired to various sensors and programmed with MakeCode, a block-based programming language.

Figure 1
One student's PCS set up with one sensor (sound sensor)



Before implementing the storyline, participating teachers attended PL sessions which provided training on wiring and programming the PCS to explore scientific phenomena. The focus of this study is on the PL activities designed to help teachers support their students during moments of uncertainty while debugging. The PL activities were designed to provide teachers experience debugging both from the student and teacher perspectives (called student-hat/teacher-hat) (Biddy et al., 2021). During the activity, teachers were organized into groups of three, with two acting as a pair of students debugging a PCS and the third playing the role of the teacher. Each group was given a buggy PCS, planted with commonly occurring bugs that students encountered in previous implementations of the unit. The teacher pair, playing the role of students, collaboratively debugged the provided PCS, trying to find and fix the various hardware and software bugs that had been planted. The person in the teacher role gave the "students" a few minutes to work on the bug prior to entering into the conversation and facilitating the debugging process. After the activity, the PL facilitators engaged teachers in a reflective discussion about the activity, focusing on strategies teachers found helpful for supporting students during moments of uncertainty.

Methods and study context

This study took place in a rural-serving district in the western United States. Teachers participated in a four-day PL workshop designed to help them learn about computationally-rich storylines using the PCS. Two teachers were selected for further analysis because we had complete records of their PL activities and classroom enactments. These two teachers are representative of the other teachers in the project in that they were middle and high school science teachers who took part in the four day workshop and enacted the instructional unit in their classroom during the fall of 2022. The teachers, Eric and Trevor, were in their first year participating with the project and neither had prior programming experience. Eric was a third-year high school science and physics teacher and Trevor was a fourth-year middle school science teacher.

The data sources included video recordings and transcripts from the professional learning workshop's debugging activity (41 minutes) and reflective discussion (12 minutes); video recordings of debugging moments, transcripts, and field notes from Eric and Trevor's enactments of the storyline; and email correspondences. We focus our analysis on two lessons (lesson 2 and 4) where students collaboratively program and wire the PCS. We consider a debugging moment to involve the moment the teacher approaches the student(s) working on their PCS until the teacher walks away. Reviewing Eric and Trevor's classroom video, we located five debugging moments for Eric and four for Trevor. Debugging moments lasted between 49 seconds to six and a half minutes.

To answer our research question, we conducted a thematic analysis (Braun & Clarke, 2006) of teachers' discussions during the PL activities as well as their strategies during classroom enactments. We developed inductive process codes (Miles et al., 2014) to surface the strategies teachers used for debugging during the PL activity, discussion, and classroom enactments. We conducted second level descriptive coding of these strategies tracing the situations where teachers did not have a consensus or obvious strategy. To compare the strategies teachers took up, which ones they adapted, and new strategies that arose in the moment, we created a checklist matrix (Miles et al., 2014). Finally, we distilled themes based on our analysis.

Findings

In this section, we describe the themes we distilled through analyzing the debugging PL activity, the reflective discussion afterwards, and Trevor and Eric's later classroom enactments of the storyline unit. These themes drive our findings section: 1) developing strategies and understandings that promoted confidence and 2) setting the stage for managing how their students engaged with uncertainty.

Developing strategies and understandings that promoted confidence

The debugging PL activity helped teachers gain knowledge about the hardware and software and develop debugging strategies that assisted them in making decisions to support their students. While in "student-hat", teachers practiced different debugging strategies – e.g., reading the code aloud, checking the wiring, and using the available instructional resources like a wiring diagram – examining their effectiveness from a student perspective. By working in the role of a student, teachers developed knowledge about the hardware and software interactions, interrogating the relationship between the environment, the sensors, and their program. They collaboratively made sense of how the hardware and software worked together, a task that is often difficult for novices in working across the physical system and programming simultaneously (DesPortes & DiSalvo, 2019).

Teachers did more observing than talking when in the role of the teacher (teacher-hat) during the PL, often positioning themselves behind the "students" so they could look over their shoulder and examine the code. During the reflective discussion, Trevor described that his trepidation to step in stemmed from: "when I saw those two, … [I felt that] I was already watching the professionals and they were already figuring out stuff that I wouldn't have thought."

When in teacher-hat mode, we only coded two strategies to support "students'" debugging during the activity: asking questions and giving directions. Both Trevor and Eric used initial questions to get a sense of what the "students" were working on when they first walked up (e.g., "so what are you guys working on here?") and questions about hardware (e.g., "What else controls how much if you get power or not?"). While these strategies are helpful, they only touch the surface of possible moves that teachers can do to support students in engaging with uncertainty. Despite observing only surface level engagement in these practices. Participating in this activity led to Eric and Trevor in discussing a multitude of deeper strategies during the reflective discussion.

The PL reflective discussion proved to be a rich activity to help teachers in building a repertoire to draw upon during the enactment of the unit. Our analysis of classroom debugging moments revealed that the teachers used seven of the eight strategies that were mentioned during the reflective discussion and two of the strategies used during the PL activity (see Table 1). Eric left the PL experience with a clear checklist he imagined he would use in his classroom, saying: "It's literally wire. It's power supply, wires, switches, code…That's the checklist and that's the order I want it." In contrast, Trevor only mentioned strategies for troubleshooting hardware. He mentioned wanting to know the wiring so well that he could walk up to students and quickly assess if there was an issue with the wiring. This strategy was evident during his later classroom enactments as he focused on asking students questions to make sure their hardware was working correctly.

Table 1
Teacher strategies derived from PL and classroom enactments

Strategy	PL Activity	PL Reflection	Classroom Enactments
Asking questions	√	√	√
Articulating the problem		√	
Articulating the goal		√	√
Reading code aloud			√
Talking to other students		√	
Pointing students to resources		√	√
Switching hardware		√	√
Allowing students to figure it out		√	
Giving directions	√		√
Checking switches or lights		√	√

Through participating in the PL activities, teachers also devised strategies to support students in learning the process of debugging, rather than focusing on simply fixing the bug. They imagined asking students questions to get them to explain or talk through their code, pointing students to tutorials or other instructional resources, and asking students to check with other students or compare their code.

During the classroom enactments, both Trevor and Eric regularly asked questions of their students as their main support strategy, building on the foundational practice they developed during the PL. However, the questions Eric asked of his students were more disciplinary-specific than the questions he asked during the PL. Most of the questions he asked during the PL were questions that could be asked in most instructional contexts to assess what students were working on (e.g., "what are you trying to do?" or "what is your goal?"). In contrast, Eric's most frequent question during enactments was "how are you going to display your data?" This question became a regular strategy for Eric in determining students' goals while also getting them to explain how they intended their system to work.

The PL reflective discussion revealed a tension among teachers about how to best support students when assembling and using the PCS. In the discussion, the teachers considered when it is appropriate to step in and when it is better to let students struggle, moving back and forth between thinking from the student's perspective (e.g., "I'm the kind of student that doesn't want someone to just tell me the answer. I want to work on it long enough that I get the chance to find it myself before") to the teacher's perspective (e.g., "Like you're a bad teacher if you don't walk around and hover so you can check and say, How's it going…if they're not ready… you move on"). Discussing this tension, Trevor reflected that students are often appreciative when teachers step in, yet not all teachers participating in the discussion agreed with this assertion.

In the reflective discussion, Trevor and Eric recognized that many debugging moments would require them to enter a situation where they would not necessarily know where students were in the process of debugging. They noted that they would want to quickly determine how to support students while also quickly assessing where students were getting stuck. They concluded that asking questions such as "what are you trying to do?" would support both their students and themselves in deciding how best to proceed.

Setting the stage for students' engagement with uncertainty

In this section, we describe debugging moments excerpted from the two teachers' enactments, illuminating how they both drew upon the repertoire developed during PL and moved beyond these strategies, improvising to best support their students. In addition, we share the ways both teachers structured lessons 2 and 4 as part of their grounding for engaging with students working with the PCS. By setting the stage for the kind and amount of improvisation necessary to support their students, Eric and Trevor expanded or contracted the amount of uncertainty available for students. Eric's use of strategies and structure of lessons 2 and 4, compared to Trevor's, resulted in a more expansive approach and required a broader improvisational repertoire that he had to call on in-the-moment when supporting students.

For example, there were several times when Eric approached students who were stuck where he followed a similar checklist he had devised during the PL activity. In an email written after implementing the storyline, Eric wrote “that debugging activity we did [in PL] was a life saver because now I know all the mistakes they make and how to fix them” (email correspondence, September 2022). While using his checklist and asking questions were helpful in the initial stages of approaching students who were stuck, Eric improvised when these initial questions did not get to the root of the problem and when neither he nor his students could locate the bug. In one instance, after reading the code aloud (the last step in his mental debugging checklist), he checked which tutorial students were using, suggested students pull up a new tutorial to compare their code to the code in the tutorial, read the tutorial code aloud, moved back and forth between the students’ code and the example code, and changed a variable in the code all the while narrating to his students what he was doing and his hypothesis about what might be causing the bug.

While in this instance these strategies ultimately did not help Eric and his students resolve the bug, he approached the experience positively by saying to his students: “I don’t know if I’ve ever seen this before, this is fun!” At the end of the debugging experience, Eric proposed a hypothesis about what might be causing the issue, explaining the details to his students and expressing that while the sensor was still not returning a value they expected, the reason might be that they did not have enough variability in moisture level and that they could try measuring in a new environment. Eric’s approach here involved improvising new approaches while also modeling how not knowing the answer or direction can be an enjoyable act. At the end of the moment, Eric also allowed his students to remain in uncertainty as they came to no obvious way to debug their system. Instead, he proposed a new strategy of trying to try out the PCS in another environment. Overall, Eric’s approach, as indicated in his facilitation of lessons 2 and 4, expanded possible stuck points students could encounter thereby creating more room for them to encounter and deal with uncertainty.

Trevor also drew upon many resources from the PL, including asking questions of students; however, his questions were often in the form of more answerable “yes or no” questions, such as “did you hit download again?” and “you’ve wired it correctly?” Trevor started his interactions with students with these types of questions, geared towards troubleshooting the hardware and its connection to the program. When these did not help resolve the issue, Trevor would then examine the code. When there, he most often directed students about what to do by pointing to different blocks on the screen and directing students what to do. For example, leaning into the screen to decipher the code, Trevor explained to two students where to move a different block in their code: “Pull that one back out. [points at screen and moves finger to match verbal directions]. And then take this right here and put it, is there a circle in it? Or just put it right here and move this up here into the show...” Overall, this directive approach was more focused on the product of getting a working PCS rather than the process of developing debugging skills.

Trevor’s approach to setting the stage of the unit limited the possible stuck points students might confront. He structured lessons so that his students were always at the same place, providing all directions aloud to the whole class in a step-by-step fashion, and asking students to stop after they finished each step in order to wait until all students were ready to move on. In addition, Trevor chose to modify lesson four (where students designed PCSs using multiple sensors). Instead, he asked students to write in their science journals about what they could create with their PCS. Through these actions, Trevor constrained students’ opportunity for engaging in uncertainty, which also minimized the amount of improvisation needed to support students in debugging moments. In this way, Trevor’s range of improvisational moves were more constrained than Eric’s.

Conclusion and implications

This paper examines two teachers’ participation in professional learning (PL) activities designed to help them learn to support students when they face uncertainty during a computationally-rich science storyline. It also examines how this participation shaped their subsequent classroom enactments. Findings revealed that the PL activities helped the teachers understand the PCS from their students’ perspectives and develop a repertoire of strategies to support students. Eric and Trevor’s classroom enactments also revealed divergent approaches to facilitating the storyline unit that enabled students to engage in and wrestle with varied levels of uncertainty.

In terms of classroom enactments, the two teachers set the stage to allow for distinct levels of student uncertainty. This meant differences in the kind and amount of improvisation needed to support their students. One teacher, Eric, created opportunities for students to encounter uncertainty by how he taught the storyline and approached students when they were debugging their PCS. In this way, giving students more opportunity to encounter and persevere through uncertainty with the PCS, Eric faced more instructional decisions to be made in-the-moment and therefore presented a more complex setting for improvisation.

In contrast, Trevor appeared to constrain opportunities for his students to encounter uncertainty – thus limiting his need to improvise while teaching. During the storyline unit, Trevor had his students all work at the same pace and provided instruction that applied to many students at once. This reduced students’ chances to encounter uncertainty on their own. This more structured approach also meant that the bugs students encountered were often routine and easily "fixed" without them necessarily learning more about the PCS and related science. While this approach constrained students from experiencing uncertainty, it may have provided Trevor an amount of comfort as he learned to integrate a new science unit and physical computing systems into his classroom. Of course, a variety of external factors may have influenced these differences in enactments, for example, class size, students' age, time, or comfort with the science and technology. In addition, rather than simply asking why some teachers' practices changed more than others after PL, it is important to recognize that teacher learning coevolves with classroom practice over time (e.g., Kazemi & Hubbard, 2008).

Trevor's approach also aligns with findings from recent studies of teachers enacting inquiry science instruction, where although teachers seek to empower student inquiry, they may only take up student answers or ideas they had previously anticipated (Miller et al., 2018). This approach can minimize students' agency (Miller et al., 2018) and lead students to strive toward a single, correct solution. Moreover, when students do not have a chance to wrestle with uncertainty, they miss out on important opportunities to develop their own strategies to work through this feeling, potentially even finding pleasure in uncertainty or inconsistencies, as many scientists describe their work (Jaber & Hammer, 2016).

Findings for this work help inform future designs of professional learning where teaching for uncertainty is foregrounded. First, our work contributes to research on the benefits of providing both "student-hat/teacher-hat" experiences in PL by giving teachers the chance to engage in the discipline, experience uncertainty from both perspectives, and empathize with their students (Lowell & McNeill, 2020). We found that the approach was an effective tool to introduce teachers to unfamiliar computing concepts (Goode et al., 2014) and help them develop strategies to support their own students in debugging. Second, we found that while in "teacher-hat," the teachers mostly observed what the "students" were doing rather than offering support. While teachers may have felt their own uncertainty about how to support students during debugging (Yadav et al., 2016), the experience sparked a rich discussion and helped teachers develop new strategies and confidence.

Finally, the design of the PL activities was intended to support teachers in developing their own flexible approaches that could be adapted to a variety of debugging and uncertainty contexts. We considered alternate PL designs that would provide teachers with pre-planned debugging scripts and checklists. However, we feared that providing such guidance would turn the focus of the activity on simply fixing bugs rather than supporting teachers in developing a more expansive set of pedagogical strategies. Whether teachers actively participated while in teacher-hat or merely observed, our PL activities appeared to engage teachers in theorizing about the nature of uncertainty in the science classroom.

References

Anastopoulou, S., Sharples, M., Ainsworth, S., Crook, C., O'Malley, C., & Wright, M. (2012). Creating personal meaning through technology-supported science inquiry: Learning across formal and informal settings. *International Journal of Science Education, 34*(2), 251–273.

Biddy, Q., Gendreau Chakarov, A., Bush, J.B., Hennessy Elliot, C., Jacobs, J., Recker, M., Sumner, T., & Penuel, W. (2021). A professional development model to integrate computational thinking into middle school science through co-designed storylines. *Contemporary Issues in Technology and Teacher Education, 21*(1).

Blikstein, P., & Moghadam, S. (2019). Computing education literature review and voices from the field. In S. Fincher & A. Robins (Eds.), *The Cambridge Handbook of Computing Education Research* (pp. 56–78). Cambridge University Press. https://doi.org/10.1017/9781108654555.004

Braun V. & Clarke V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77–101.

DeLiema, D., Kwon, Y., Chisholm, A., Williams, I., Dahn, M., Flood, V., Abrahamson, D., & Steen, F. (2022). A multi-dimensional framework for documenting students' heterogeneous experiences with programming bugs. *Cognition and Instruction*, *41*(2), 1–43. https://doi.org/10.1080/07370008.2022.2118279

DesPortes, K. & DiSalvo, B. (2019). Trials and tribulations of novices working with the Arduino. *Proceedings of the 2019 ACM conference on International Computing Education Research (ICER '19)*. Association for Computing Machinery, 219–227. https://doi.org/10.1145/3291279.3339427

DeZutter, S. (2011). Professional improvisation and teacher education: Opening the conversation. In R. Sawyer (Ed.), *Structure and Improvisation in Creative Teaching* (pp. 27–50). Cambridge University Press.

Erickson, F. (1982). Classroom discourse as improvisation: Relationships between academic task structure and social participation structure in lessons. In L. C. Wilkinson (Ed.), *Communicating in the Classroom* (pp. 153–181). Academic Press.

Fields, D., Kafai, Y., Morales-Navarro, L., & Walker, J. (2021). Debugging by design: A constructionist approach to high school students' crafting and coding of electronic textiles as failure artefacts. *British Journal of Educational Technology, 52*(3), 1078–1092. https://doi.org/10.1111/bjet.13079

Goode, J., Margolis, J., & Chapman, G. (2014). Curriculum is not enough: the educational theory and research foundation of the exploring computer science professional development model. *Proceedings of ACM Technical Symposium on Computer Science Education.* Association for Computing Machinery, 493–498. https://doi.org/ 10.1145/2538862.2538948

Haduong, P., & Brennan, K. (2019, February). Helping K-12 teachers get unstuck with Scratch: The design of an online professional learning experience. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, 1095–1101.

Halverson, E. (2021). *How the arts can save education: Transforming teaching, learning, and instruction.* Teachers College Press.

Hardy, L. Dixon, C., & Hsi, S. (2020). From data collectors to data producers: Shifting students' relationship to data. *Journal of the Learning Sciences, 29*(1), 104–126.

Hennessy Elliott, C., Gendreau Chakarov, A., Bush, J., Nixon, J., & Recker, M. (2023). Toward a debugging pedagogy: Helping students learn to get unstuck with physical computing systems. *Information and Learning Sciences, 124*(1/2), 1–24. https://doi.org/10.1108/ILS-03-2022-0051

Hennessy Elliott, C., Nixon, J., Bush, J., Gendreau Chakarov, A., & Recker, M. (2022, June). "Do I need to know what I am doing if I am the teacher?": Developing teachers' debugging pedagogies with physical computing. *Proceedings of the 16th International Conference of the Learning Sciences.* International Society of the Learning Sciences, 1565–1568.

Jabar, L., & Hammer, D. (2016). Learning to feel like a scientist. *Science Education, 100*(2), 189–220.

Jordan, M. E., & McDaniel, R. R., Jr. (2014). Managing uncertainty during collaborative problem solving in elementary school teams: The role of peer influence in robotics engineering activity. *Journal of the Learning Sciences, 23*(4), 490–536. https://doi.org/10.1080/10508406.2014.896254

Jurow, A., & McFadden, L. (2011). Disciplined improvisation to extend young children's scientific thinking. In R. Sawyer (Ed.), *Structure and Improvisation in Creative Teaching* (pp. 236–251). Cambridge University Press.

Lowell, B. R. & McNeill, K. L. (2020). Using the student hat to push on multiple goals in teacher professional learning. *Proceedings of the 14th International Conference of the Learning Sciences*. International Society of the Learning Sciences, 2241-2244. https://doi.dx.org/10.22318/icls2020.2241

Philip, T. (2019). Principled improvisation to support novice teacher learning. *Teachers College Record, 121*(6), 1–32. https://doi.org/10.1177/016146811912100607

Reiser, B. J., Novak, M., McGill, T. A., & Penuel, W. R. (2021). Storyline units: An instructional model to support coherence from the students' perspective. *Journal of Science Teacher Education, 32*(7), 805–829.

Sawyer, K. (2004). Creative teaching: Collaborative discussion as disciplined improvisation. *Educational Researcher, 33*(2), 12–20. https://doi.org/10.3102/0013189X033002012

Manz, E., & Suarez, E. (2018). Supporting teachers to negotiate uncertainty for science, students, and teaching. *Science Teacher Education, 102*(2), 771–795.

McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: a review of the literature from an educational perspective. *Computer Science Education, 18*(2), 67–92.

Miller, E., Manz, E., Russ, R., Stroupe, D., & Berland, L. (2018). Addressing the epistemic elephant in the room: Epistemic agency and the next generation science standards. *Journal of Research in Science Teaching 55*(7), 1053–1075.

NGSS Lead States. (2013). Next Generation Science Standards: For states, by states. Washington, DC: National Academies Press. www.nextgenscience.org/next-generation-science-standards.

Miles, M.B., Huberman, A.M. & Saldana, J. (2014). *Qualitative Data Analysis: A Methods Sourcebook*. Sage.

Tsan, J., Weintrop, D., & Franklin, D. (2022, July). An analysis of middle grade teachers' debugging pedagogical content knowledge. *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education.* Association for Computing Machinery, 533–539.

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education, 26*(4), 235–254.

Acknowledgements

This research was funded by the National Science Foundation (Award No. 1742053, No. 2019805 and No. 1742046) and the James S. McDonnell Foundation. The opinions expressed are those of the authors and do not represent views of the funding organizations. We thank our participating teachers.