

Challenges of preparing secondary STEM pre-service teachers in computational thinking

Hannah C. Sabo and Tor Ole B. Odden

Center for Computing in Science Education, Department of Physics, University of Oslo, 0316 Oslo, Norway

Tone F. Gregers

KURT, Department of Education, University of Oslo, 0316 Oslo, Norway

Computational thinking (CT), the thought processes used in solving computational problems, has been added to the educational standards in many countries around the world. However, this new content presents challenges around preparing teachers to integrate computational thinking robustly into the STEM disciplines. In this proceedings, we conduct a preliminary investigation of the challenges of preparing secondary pre-service teachers to integrate computation into their subject teaching. We present data from interviews with three pre-service secondary subject teachers in the KURT at UiO to highlight two challenges of pre-service teacher (PST) preparation: first, (1) the computation learned in upper division subject courses does not directly translate to the computational thinking that they will be teaching their future students; and (2) in their pedagogy courses, they learn about computational thinking but are not able to translate that into their practice of teaching. We consider the implications of and solutions for these challenges on pre-service teacher preparation.

I. INTRODUCTION

Since the 1980s, supercomputers have transformed how science and mathematics research is conducted, adding computational branches of these sciences. It is important to teach science in a way that is aligned with and prepares students for the modern "doing" of the sciences [1], which means incorporating computational thinking (CT) into middle and high school science courses. Additionally, CT has been incorporated into primary and secondary school standards across the world, including the United States [2, 3], the United Kingdom [4], and Norway [5]. So, it is essential to prepare teachers to integrate CT into their teaching, so that their students can leverage CT to gain physical insights in their related disciplines. There have been efforts to equip both pre-service and in-service teachers to incorporate computational thinking into their teaching.

In 2020, Advancing Interdisciplinary Integration of Computational Thinking in Science, a conference report, articulated a set of research areas and recommendations to guide the international effort to incorporate CT into K-16 schooling [6]. One of the main focuses of the report was professional development and teacher learning. In our study, we focus on of three of the four recommendations relating to teacher learning, which are to: (T1) determine the scope of pre- and in-service science teacher and science classroom computational needs at all levels, and determine the quantity and duration of professional development necessary for teacher expertise to develop and to be maintained; (T3) articulate and leverage theoretical frameworks for integrated computing education to guide the development of classroom tools, software, content standards, assessments, and expectations for teacher preparation and ongoing teacher certification; and (T4) understand whether and how teachers come to value computing as a component in science and mathematics education [6].

In this paper, we present a preliminary study responding to these recommendations, a first step in pursuit of the following research question: *What challenges arise when preparing pre-service, secondary STEM teachers to integrate computational thinking into their disciplinary teaching?*

address this research question, we analyze interview data from a Pre-Service secondary, subject Teacher (or PST) training program at the University of Oslo, in Norway.

II. LITERATURE REVIEW

To meet these new standards, Norwegian teacher educators must prepare PSTs to integrate CT into their disciplinary teaching. The turnaround between the introduction and implementation of these new standards is short. Because of the lack of historical research and experience, teacher educators face challenges related to preparing teachers in CT. As researchers, we are looking to identify these challenges so that we can improve PST education. In this section, we look to the literature to understand the current practices for preparing PSTs in CT, especially previously documented challenges in computational education.

CT, defined as "the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer - human or machine - can effectively carry out" [7], extends beyond simply coding and programming. In this paper, we envision disciplinary CT as a set of computational problem-solving techniques and meta-knowledge that students can use to make inferences about mathematics and science subjects.

CT is not a single construct, but rather a cluster of concepts, skills, and practices used in computational science [6] and is domain-dependent [8]. So, CT lacks a clear framework or definition. This flexibility allows secondary, subject PSTs to incorporate CT into their teaching in a way that is consistent with how CT used in their field; however, it complicates teacher training. How can teacher educators prepare a variety of subject teachers in CT?

Some PST training programs have incorporated CT into their science and math teaching methods courses; however, few researchers have studied this development. Within the limited body of research that exists, PSTs have learned about CT and its pedagogy through week-long modules in their math and science teaching methods courses [9–11]. In Yadav's course [11], the first module covered a CT framework. In the second module, the PSTs were engaged in CT activities. The authors found that after the modules, PSTs' understanding of and attitudes towards CT improved. While these efforts show that PSTs' have an increased understanding of CT, they do not examine teachers' preparedness to incorporate CT into their disciplinary teaching.

Some researchers and reports have documented some of the emergent challenges of training K-16 students and teachers in CT [1, 12]. At a professional development seminar, Hestness et al. [12] found the following challenges for training in-service elementary teachers: the need for professional support; a lack of time; and their lack of familiarity with CT. The teachers had no previous experience with computational thinking, so extra preparation and support is necessary.

The American Association of Physics Teachers, or AAPT, [1] published a report on incorporating computation into college-level physics major programs. They reported the following challenges: lack of explicit curricular goals; a range of instructor backgrounds; a range of student backgrounds; shortage of educational research; and lack of community and support. College physics instructors have limited access to resources based on their own backgrounds and the schools where they teach.

While neither of the articles in the previous paragraph focus on secondary, subject PSTs, we anticipate similar challenges occurring in our research context as found in Hestness [12] and the AAPT report [1]. However, there is a need for more research on the specific challenges that secondary subject PSTs face. In this paper, we present an exploratory study of two of those challenges. We will use these findings to further investigate PST preparation.

III. CONTEXT AND METHODS

In this paper, we present a pilot study which is the first step in a larger project investigating a pre-service teacher education program at the University of Oslo (UiO). We plan to use these findings to find strong themes, and refine the next iteration of our interview protocol. The data collection in this paper is limited to three interviews with PSTs.

A. Context

The study takes place in a Norwegian university secondary, PST preparation program. At this university, computation has been integrated into physics courses for many years. Recently, Norway has added CT standards into the K-12 curriculum.

The KURT (Kompetansesenter for undervisning i realfag og teknologi) program at UiO offers a 5-year integrated master's degree where the PSTs learn to teach in lower (age 13-16) and upper (age 16-19) secondary schools. Here the PSTs follow an educational "dual track" program containing courses from both subjects, like physics and mathematics, and pedagogical courses containing pedagogy, teaching methods and student teaching. The PSTs take the same science and mathematics courses as "pure" science and mathematics students. Since 2017, computational science has been integrated from the first semester in all science and mathematics subjects at the Faculty of Mathematics and Natural Sciences at UiO, including a first-year mandatory programming course. However, at the moment there is no emphasis on CT teaching methods in these subject courses. Thus, the teacher education program at UiO ensures that newly qualified teachers have a strong understanding of disciplinary computational science in their chosen subjects. However, the integration of CT-specific pedagogy is currently limited.

This context provides a unique case study where computation has already been robustly integrated into the subject courses since at least 2017. But, computational thinking was only recently added to their pedagogy courses. We study the early challenge that arise in preparing teachers to not only use computation but teach it effectively.

B. Data collection and analysis

We conducted interviews with a few PST students (N=3) at the end of the academic year. These students were recruited by email and financially compensated for their time. Our participants were in the last two years of their program and had completed their student teaching.

The interview protocol was developed by the first author, who is familiar with physics education research, and reviewed by the second author, who is familiar with both the teacher education program and Norwegian culture. She ensured that the protocol made sense within the Norwegian teacher training context.

We designed the interview protocol to investigate three aspects of their experience in the education program: (1) the

PSTs' experience and confidence with computational science in their subject courses and, (2) their experiences with CT in their teaching methods courses, and (3) their perceived readiness to incorporate CT into their own teaching. The questions were designed to elicit the PSTs' day-to-day experiences and to provide context into what aspects of their program were helpful or not helpful.

The interviews were conducted in English by the first author, which both the first and second author attended. When necessary, the second author translated between the interviewer and interviewee. Follow-up questions were asked when more clarity was needed. After the set of interviews was complete, the audio was transcribed by the first author.

A thematic analysis [13] was used to investigate which aspects of their dual track program prepared or did not prepare the PSTs to integrate computation into their teaching. The themes centered on disconnections between the PSTs' preparation in CT and their perceived readiness to teach CT in their future teaching placements. These disconnections highlight current challenges in the KURT program.

From this analysis, two themes, or challenges of teacher preparation, emerged. Due to the preliminary nature of the interviews and the limited data set, we report on only two themes which were representative across the data set. With such a small data set, this is appropriate. We will use these themes to inform the next round of interview protocol revision. We considered themes to be challenges where all three students reported feeling unprepared.

IV. FINDINGS

We report two challenges across PST preparation at UiO. We found: (1) there is a disconnect between computation integrated into disciplinary courses and the necessary preparation for the PSTs to use that in their teaching practice; and (2) there is a disconnect in pedagogy/teaching methods courses where the knowledge and standards of CT is disconnected from their integration of CT into their teaching practice. We demonstrate these themes with quotes from the PSTs. All names are pseudonyms.

A. Challenge 1

Although the PSTs learned disciplinary computation in their subject courses, the PSTs saw those programming experiences as less relevant to their future practice.

During her interview, Camilla articulates this disconnection:

I would have liked to have seen more programming with the level of maths that we're going to teach because most of my programming has been done at a higher level in maths or physics. So to have done it at a lower level would have been nice. And to have done it practically because programming is something that you have to do to learn it. That's what I feel like I've missed

most... I can't really use the fact that I can find the eigenvalues or matrices in the uh high school level maths. (Camilla)

Similarly, Kristoffer discusses how he sees his earlier programming experiences as more relevant to what he will teach his future students than most of what he learned in his subject courses.

I believe that like the first year, the first semester, and the physics mechanical introduction course is the most relevant for programming that we are teaching in upper secondary school, the next years... that meets the curriculum for the upper secondary school most, and there were several times that I felt that the programming we did in the later physics courses was nice to work with it, but that was way above what we're supposed to teach. (Kristoffer)

Camilla and Kristoffer do not see their upper division programming as continuous with what they will be teaching. They see the CT in their introductory courses as more relevant than the upper division courses. Camilla and Kristoffer propose a way forward: to learn CT in the context that they will be teaching it rather than in upper division courses.

Camilla and Kristoffer did not see the same disconnection with physics concepts in their upper division subject courses, which also included material beyond the level of what they will be teaching.

Diana highlights a difference between what she learned in her subject courses and what her students will be expected to learn, but offers a different perspective.

It's not the purpose that they're [the students] going to be programming the whole thing, but they need to understand how the program works and how to use it. So, at university, we learn how to program everything from scratch. (Diana)

While Diana highlights the disconnection, she distinguishes between what she, the teacher, is expected to know and what her future students will learn, which might involve understanding and using programs, but probably not writing them from scratch, similar to the use-modify-create [14] and minimally working program [15] frameworks. Diana does not elaborate on why teachers learn beyond what they teach their students. She accepts that it is that way, so it makes sense for CT to be the same way.

With just their upper division programming experience, the PSTs do not feel prepared to authentically incorporate CT into their teaching of disciplinary subjects because they lack experience with CT at the level they will be teaching. They are not sure how to translate what they learn in their upper division courses to the content that they will be expected to teach in their future teaching placements. This disconnection highlights a challenge of PST preparation: it is not enough for them to learn computation in their upper division courses.

That knowledge is not connected to their practice of teaching yet.

B. Challenge 2

Additionally, the PSTs learn about CT and its standards in their pedagogical courses. However, the PSTs did not learn how to integrate those ideas into their teaching practice.

When asked what she learned about CT in her pedagogy courses, Camilla summarizes this disconnect, saying

We discussed that [the CT standards were] coming in, right as it was being introduced. So, it was like, how will this look like? How much programming are they going to know at each stage? ... But not that much about directly about how to teach it. I think there was an underlying assumption that it would be similar to maths. (Camilla)

Kristoffer also discusses how there was very little focus on CT during the pedagogy courses that he took:

I don't remember having a lot of focus on programming during those [pedagogy] courses. Very little, I think. And that's a thing that I've been thinking about those last months when I thought about teaching programming... But, I definitely think that I could or should have had more knowledge on teaching programming. (Kristoffer)

In their interviews, Camilla, Kristoffer, and Diana recall learning about the CT standards in their teaching methods courses when it was first introduced to the standards. Camilla learned about what she will be expected to teach her future students, but not about pedagogy for CT. Kristoffer states that he feels like he should have learned more about CT pedagogy; and that it was missing in his pedagogy courses, especially given its presence in the standards.

Kristoffer and Camilla emphasize a disconnection between learning about CT and learning to implement it in their teaching practice. They feel unprepared to integrate CT in their lesson plans. From this disconnect, another challenge can be inferred: it is not enough to learn about CT and its standards. They must learn how to integrate CT into their teaching and meet the standards.

V. DISCUSSION

A. Challenge 1

The PSTs did not see a connection between their upper division subject CT experiences and the computational thinking that they will be teaching. However, we, the authors, still see value in their upper division coding experiences. Since disciplinary computation is domain-dependent [8], a disciplinary understanding of CT in the subjects can help them integrate authentic CT into their science and math teaching. One challenge that Hestness [12] identified was that teachers may need

to cultivate their understanding of what CT looks like in the disciplines. Therefore, upper division coding experiences are valuable for PSTs' understandings of how computation can be used in the service of gaining physical disciplinary insights. Although that computation may exceed what they will teach, the computational methods and techniques that they learn may be valuable in their practice of teaching CT.

College-level subject courses are frequently very busy with set curriculum [1]. Often times, science and math departments do not prioritize PSTs in their curricular decision making. So, it may be difficult to convince departments to provide CT experiences in the introductory courses, especially in a way that considers equipping PSTs to translate it to their teaching.

B. Challenge 2

The PSTs also learned about CT during their teaching methods courses. However, they only learned about the standards and what they are expected to teach their students, rather than CT-specific pedagogy. "Drop-in" programming and CT lessons have been well-documented in literature for the preparation of elementary PSTs [9–11]. These modules on CT have been studied and shown to be effective for helping those PSTs develop a more authentic understanding of CT. These modules respond to the challenge of limited time in the busy pre-service teacher curriculum [12]. While the modules are helpful, our findings suggest that a week-long module on CT may be insufficient for secondary subject PSTs, who may need more development in both their own computation and their ability to integrate into secondary-level science and math lessons.

C. Across both challenges

Each of the two challenges occur on a different side of the PSTs' dual-track program. The first challenge occurs in their subject courses, while the second manifested in their teaching preparation courses. Although the PSTs were given adequate experience programming themselves, they were not able to bridge the connection between the two sides of the program. This raises the question: how can we get these two components to integrate together?

Furthermore, the PSTs reported feeling unprepared to integrate CT into their teaching, we see the PSTs as close to prepared, just not *feeling* prepared. With their disciplinary computation experiences and pedagogical training, the PSTs are almost ready to design lessons that integrate CT into their subjects. We suspect that this gap can be bridged by having the PSTs explicitly reflect on their experiences with disciplinary computational science experiences and consider how to bring those ideas to a high school level. Additionally, we

can further prepare the PSTs by integrating pedagogical tools for teaching CT in their pedagogy courses.

D. Limitations

Due to the limited number of interviews, this study has significant limitations. We interviewed three PSTs and reported out some themes of challenges that were common among them. We are using this data to start to scope out the potential challenges of training PSTs and refine our understanding of the space for future studies.

The research setting provides a limitation since 2017 (and earlier for physics courses), UiO has incorporated programming and computational science into all physics and mathematics courses. The PSTs enrolled in the KURT program likely receive more disciplinary computational instruction than at other institutions. For this reason, the PSTs would likely be more prepared to teaching CT than those from another institution. Yet, they still reported being under-prepared and unsure of how to integrate their learning into their teaching practice. At another institution, the pre-service teachers may be even less sure of how to connect upper division subject learning to computational thinking at the high school level.

VI. CONCLUSION

We responded to the recommendations of the Advancing Interdisciplinary Integration of Computational Thinking in Science report [6] for professional development and teacher learning by conducting a preliminary investigation in the KURT program at UiO. From our interviews, we identified two challenges that guide our future research and preparation of teachers. We have determined that secondary subject PSTs might need more preparation than they are currently receiving. We need to guide the PSTs in making explicit connections between the computational science that they learn in their upper division subject courses, and give them more pedagogical tools to integrate CT into their teaching practice. This finding also helps us start to identify a framework for PST preparation. Both sides of the dual track program are critical for our PSTs' CT teaching practice, however, the connections are not obvious and may require some explicit reflection on the PSTs part. Finally, the PSTs saw value in learning computation that would be relevant to their practice of teaching; they valued CT that was useful. We use these findings to guide our next iterations of both research on PSTs learning computation and future iterations of our teaching methods courses.

Acknowledgments

We are very grateful to Cathrine Wahlstrøm Tellefsen for her feedback on our work.

-
- [1] AAPT Undergraduate Curriculum Task Force, "AAPT Undergraduate Curriculum Task Force - 2016 - AAPT Recommendations for Computational Physics in the Undergraduate Physics Curriculum," The American Association of Physics Teachers, Sep. 2016.
- [2] National Research Council, A framework for K-12 science education: Practices, crosscutting concepts, and core ideas. National Academies Press, 2012.
- [3] NGSS Lead States, Next Generation Science Standards: For States, By States. National Academies Press, 2013.
- [4] Department for Education, "National curriculum in England: Computing programmes of study," Gov Uk, 2013.
- [5] Kunnskapsdepartementet, "Fornyere innholdet i skolen," Regjeringen.no, Jun. 26, 2018.
- [6] M. D. Caballero, K. Fisler, R. C. Hilborn, C. M. Romanowicz, and R. Vieyra, "Advancing Interdisciplinary Integration of Computational Thinking in Science," AAPT, College Park, MD, 978-1-931024-358, Jan. 2020.
- [7] J. M. Wing, "Computational Thinking: What and Why," in Presentation slides from Trippel Helix Conference on Computational Thinking and Digital Competencies in Primary and Secondary Education Stockholm, Sweden.
- [8] P. J. Denning and M. Tedre, Computational Thinking. MIT Press, 2019.
- [9] M. Walton and J. Walkoe, "Computational Thinking as a Context for Ambitious Math Instruction," 2021.
- [10] A. Yadav, N. Zhou, C. Mayfield, S. Hambrusch, and J. T. Korb, "Introducing computational thinking in education courses," in Proceedings of the 42nd ACM technical symposium on Computer science education, 2011, pp. 465-470.
- [11] A. Yadav, C. Mayfield, N. Zhou, S. Hambrusch, and J. T. Korb, "Computational thinking in elementary and secondary teacher education," ACM Trans. Comput. Educ. TOCE, vol. 14, no. 1, pp. 1-16, 2014.
- [12] E. Hestness et al., "Computational thinking professional development for elementary science educators: Examining the design process," in Society for Information Technology Teacher Education International Conference, 2018, pp. 1904-1912.
- [13] V. Braun and V. Clarke, "Using thematic analysis in psychology," Qual. Res. Psychol., vol. 3, no. 2, pp. 77-101, 2006.
- [14] I. Lee et al., "Computational thinking for youth in practice," Acm Inroads, vol. 2, no. 1, pp. 32-37, 2011.
- [15] P. W. Irving, M. J. Obsniuk, and M. D. Caballero, "P3: A practice focused learning environment," Eur. J. Phys., vol. 38, no. 5, p. 055701, 2017.