# Instructor facilitation of STEM+CT discourse: engaging, prompting, and guiding students' computational modeling in physics

Caitlin Snyder, Vanderbilt University, caitlin.r.snyder@vanderbilt.edu
Nicole M Hutchins, Vanderbilt University, nicole.m.hutchins@vanderbilt.edu
Gautam Biswas, Vanderbilt University, gautam.biswas@vanderbilt.edu
Gayathri Narasimham, Vanderbilt University, gayathri.narasimham@vanderbilt.edu
Mona Emara, Damanhour University, mona.emara@dmu.edu.eg
Bernard Yett, Vanderbilt University, bernard.h.yett@vanderbilt.edu

**Abstract:** The integration of computational modeling into instruction in science classrooms is complex in that it requires the synergistic application of students' developing science and computational thinking knowledge. This is not only difficult for students, but teachers often find it hard to parse the science content from the computational constructs to guide students when they have difficulties. Leveraging past literature that highlights the beneficial impact instructors can have when they immerse themselves in group problem-solving discussions, this paper examines the instructors' role in facilitating students' construction of and problem solving with computational models. We utilize a case study approach to analyze instructor-facilitated, synchronous group discussions during applications of synergistic learning processes to understand how instructors may elicit students' knowledge, misunderstanding, and difficulties to help guide, prompt, and engage groups in this complex task for more productive integration in K-12 science classrooms. We hope that this will lead to better scaffolding of students' learning, and better support for teachers when they use such curricula in classrooms.

## Introduction

Computational modeling has received increased attention as a vehicle for integrating computational thinking (CT) in existing K-12 science classrooms (Wilensky & Reisman, 2006; Hutchins et al., 2020). This constructivist approach has been shown to be effective in helping students learn STEM and CT concepts (e.g., Sengupta et al., 2013), and develop problem-solving skills (Hutchins et al., 2020). In addition, it provides environments for encouraging productive, socially-shared collaborative problem-solving skills development (Emara et al., 2021). However, integration of STEM and CT in K-12 science classrooms comes with its own set of challenges for both students (Basu et al., 2016; Hutchins et al., 2020) and instructors (Wilkerson-Jerde et al., 2015). In this paper, we target two key challenges: (1) limited understanding of how instructors can engage in students' integrated learning of science and CT during computational modeling (Wilkerson-Jerde et al., 2015), and (2) student difficulties in communicating about applying complex, synergistic learning processes, such as initializing and updating variables during computational model building, and leveraging data tools and debugging processes to generate correct models (Snyder et al., 2019). To do so, we examine instructor engagement in student thinking and knowledge construction during computational modeling through instructor-facilitated, small group discourse, a pedagogical tactic that has shown to support problem-solving skill development (Fung et al., 2016).

While student discussions during learning and problem solving have been shown to be beneficial to student learning (Chi & Wylie, 2014), group discussions are not always effective (Dillenbourg et al., 1996). These difficulties can be exacerbated during scientific modeling due to the open-ended nature of the problem-solving task, as students collectively manage the complex integration of STEM and CT concepts and practices (Snyder et al., 2019), limiting applicability of common collaborative interventions. In this context, interventions must consider students' STEM and CT prior knowledge and their ability to (1) apply complex synergistic processes, such as translating prior knowledge into computational forms, and (2) communicate these processes to group members. This may provide a unique opportunity for instructors to engage in students' STEM and CT knowledge construction and problem-solving processes to support and to facilitate STEM+CT classroom integration.

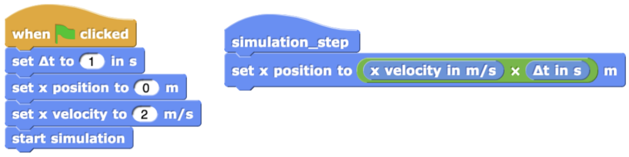
This paper examines the research question: *how can instructor-facilitated small group discussions support our understanding of students' learning and problem solving during scientific computational modeling?* To answer this question, we define a framework that captures the roles instructors play in classroom instruction and group discussion. We adapt this framework to evaluate the instructor-facilitated, small group discourse during scientific computational modeling tasks. Our analyses revolve around two case studies examining how instructors guided small-group discussions linked to their students' computational modeling tasks. We study how instructors prompted students to elicit their knowledge in CT and the target science domain, exposed misunderstandings, and helped students develop complex, synergistic learning processes to support their learning tasks. We hypothesize that targeted instructor-facilitated discourse may support a deeper understanding of students' STEM and CT knowledge that may be difficult to assess through traditional summative assessment approaches. We illustrate how instructors, through explanation prompting, encourage discussions on translating conceptual knowledge to the computational form and support the groups' knowledge co-construction processes.

## Integrating the C2STEM learning environment

C2STEM is designed to support the synergistic learning of physics and CT through the building, testing, debugging, and use of computational models to solve physics problems (Hutchins et al., 2020). The learning environment includes a block-based programming environment, an extension of Snap! (snap.berkeley.edu), equipped with a physics *domain-specific modeling language* to allow students to focus on constructing computational models of physics processes. The system includes *data tools* that provide students a means for checking and evaluating their physics models as they build them. In the curriculum students build increasingly complex models and complete a series of formative and summative assessments to help students and instructors monitor their evolving STEM and CT knowledge. The curriculum for this study was divided into three modules: (1) 1-D motion, where students build a model of a truck speeding up from rest to a specified speed limit, cruising at that speed, and then slowing down (decelerating) to come to rest at a stop sign; (2) a drone task where students build a computational model of a flying drone dropping a package onto a target; and (3) a second drone task where students model a flying drone dropping two packages in succession on two different spots on the ground.

For C2STEM implementations, we have developed instructor material to support modeling tasks based on prior classroom experience. Teachers also have access to the correct computational models. We have collected frequently-observed difficulties students face in building computational models. Difficulties include errors in domain-specific concepts and practices (e.g., setting an object's y-velocity to the gravitational constant, and errors in initializing and updating variables in CT). For example, Figure 1 shows a student's incorrect code for updating the *x*-position of an object. In this illustrated code, the *x*-position is set to '*x velocity in m/s * Δt in s*' but the student should have used the change *x*-position block. This demonstrates a potential misunderstanding about the difference between setting and changing variable values. We recommend instructors should prompt students to explain the behavior of the object by running the simulation and walking through the execution to let the student gain an understanding of how the simulation model executes and identify potential errors in their model. Another instructor move may involve prompting students to use the data tools to evaluate the position values and discuss how this information compares against their intuitions and knowledge of the physics phenomena.

**Figure 1**
*Example student CT difficulty in a student's computational model code*




## Analyzing instructor facilitation during an C2STEM implementation

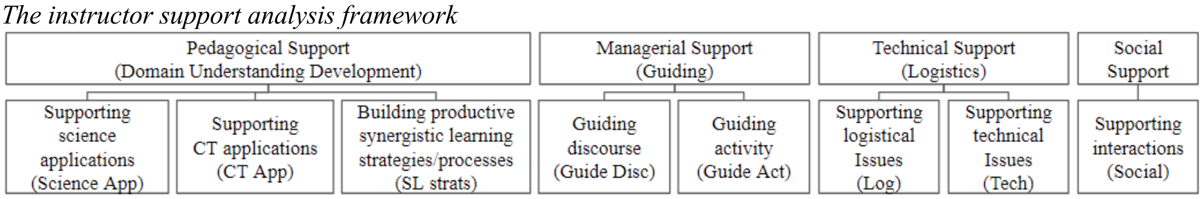
The impact of instructors during group discussions has been shown to help students' problem-solving processes as they think about the salient features of the problem and develop their problem-solving strategies (Webb et al., 2009). Research has identified strategies for facilitating online discussion that includes developing a social presence and emphasizing student-to-student interactions (Rovai, 2007). Our research is primarily focused on asynchronous online discussions and their effects on students' learning. In this research, we analyze facilitation strategies for synchronous discussions that are supported by our instructor material to improve our understanding of instructor engagement and to facilitate computational modeling of scientific processes.

Figure 1 shows the analysis framework used in our research. This approach is inspired by Berge's (1995) four instructor roles for online learning: (1) pedagogical, (2) managerial, (3) technical, and (4) social. Students may face multiple difficulties in understanding, applying, and integrating science and CT concepts in practices when building computational models (Basu et al., 2016). Instructors can mitigate these difficulties, once identified, to help students overcome these difficulties. In other words, *pedagogical support* is essential to helping students progress in their computational modeling tasks. Previous work has demonstrated that students need help in developing and applying productive synergistic learning strategies to help them translate their knowledge and understanding into successful computational structures for model building (Hutchins et al., 2020). In the form of *managerial support*, the instructor may guide the students discourse or model construction towards applying a strategy by identifying relevant goals or guiding students to think about how they translated their knowledge in a previous task. The difficulties students face can be compounded by an unfamiliarity with the environment which can be mitigated with *technical support*. Instructors may also facilitate the discussion via *social support* by prompting and encouraging students to state viewpoints and discuss them with each other.

In C2STEM, *pedagogical support* manifests as (1) *supporting CT* applications, such as how to correctly implement the updating of a variable necessary in the physical phenomenon; (2) *developing physics knowledge*, such as the relationship between velocity and acceleration to ensure correct object motion; and (3) *building synergistic learning strategies,* such as translating conditional behavior changes to the right form of conditional statements, initializing and updating variables, debugging, and using data tools to construct or problem solve. *Managerial support* encourages students to reflect and plan, such as guiding students to remember how something was implemented in a previous task or suggesting a modeling-in-parts approach to develop more complex models. The complexity of the C2STEM modeling environment (e.g., the large variety and number of blocks) necessitates *technical support* to help students navigate an unfamiliar environment. Finally, the complex nature of tasks creates an opportunity for collaboration between students, which can benefit from *social support* from instructors.

Previous work has revealed unproductive collaboration in terms of difficulties students face in applying STEM and CT constructs, as evidenced by inability to co-construct knowledge, and inability to interpret and explain model behaviors (e.g., Emara et al., 2021). Facilitating discourse is considered to be an important task and responsibility of a synchronous instructor. These instructor facilitations are often delivered in the form of instructional statements, group participation encouragement and prompting questions. Instructor's question prompting may help students clarify confusions when explaining model behavior (Grover et al., 2019), identify and understand errors during debugging (Hutchins et al., 2020), justify their problem-solving strategies, and correct their misconceptions and suboptimal use of strategies (Kinnebrew et al., 2017; Webb et al., 2009). Moreover, instructors could use various guiding support as means to make the students' interaction more thought-provoking with the aim of translating STEM concepts and practices into computational form (Kinnebrew et al., 2017; van Vondel et al. 2017). For this analysis, we coded discourse based on the lowest level of Figure 2.

**Figure 2**
*The instructor support analysis framework*




## Study description and data analysis methods

This study was implemented in the Southeastern United States with 22 high school students and 5 researchers who played the role of instructors for the study. The intervention included synchronous, small group computational modeling tasks, which were completed individually and asynchronously. They targeted students' abilities to apply what they learned during their small group work. Students worked in groups of 3 or 4 and the groups did not change through the course of the study. The synchronous, small group work occurred once a week for five weeks over Zoom, necessitated by COVID19 protocols. During these sessions, students had 10-15 minutes of instruction and 15-20 minutes of group discussion in a Zoom breakout room facilitated by the same instructor every week, except for week 3 when a full 30 minutes was used for instruction. Prior to the study, the instructors met for one hour to discuss the curriculum and review the instructor material discussed above. Following each synchronous session, instructors met for 15-30 minutes to review student progress and discuss their experiences.

Data collected for this study included students' pre- and post-assessment results, final computational model project files (scored using a predefined rubric from Snyder et al. (2019)), and the recorded Zoom sessions. When student groups collaborated, the student responsible for constructing the model (e.g., connecting the blocks, running the simulation, etc.) shared their screen on the Zoom session. For this analysis, we focus on two groups that had the same instructor in order to systematically compare differences in (1) the instructor's evidence-based responses to different group combinations and (2) each group's responses to the instructor's feedback.

We use a descriptive case study approach to analyze how instructor-facilitated small group discussion can support our understanding of students' learning and problem solving during scientific computational modeling. Specifically, we identify how instructors *identify* and *address* student misunderstandings and knowledge gaps through instructor-facilitated discussions of students' model construction steps. We examine the instructor-facilitated group discussion dialogue segments by coding the instructor utterances using the scheme in

Table 1. Note that due to the overlap between support, we combine the two types of guiding into one code label. Two coders coded the utterances with very good agreement (κ = 0.89). We analyze students' individual computational model scores and prior knowledge, measured through pretest scores. The two groups selected for analysis had the same instructor and a mixture of prior knowledge, measured through a pretest, in both STEM and CT. The physics portion of the test had a total score of 17 points, with a student median of 12 points (σ = 2.35). The CT portion of the test had a total score of 16 points, with a median of 9.25 points (σ = 2.81). We categorized students as high or low prior knowledge based on their scores relative to the median. In the two groups we analyze in this work, the CT scores ranged from 13 to 6 points and physics scores ranged from 14 to 9 points.

**Table 1**
*Coding scheme for implementing the instructor support analysis framework*

| Code | Description | Example(s) |
|---|---|---|
| Log & Tech | Helping the students log on, open projects, other technical or logistical issues, etc. | "S3 could you be the navigator and open the form today?"; "The block is in that section" |
| Guide Disc & Guide Act | Stating goals, guiding students to look at something | "Your goal for the next six minutes is to build and answer this question as a group." |
| Science App | Utterances focused on physics concepts | "What causes velocity to change?" |
| CT App | Utterances focused on CT concepts | "So, the simulation step repeats every time" |
| SL Strat | Utterances referring to synergistic learning processes or strategies | "That's an interesting way to debug"; "Definitely use the data tools to help you here"' |
| Social | Encouragement of social interactions | "Hi everyone", "S4, are you there?" |

## Results

The case studies presented target instructor-facilitated, small group discourse, of two groups, focused on key synergistic learning processes - initializing and updating variables, debugging, and using data to debug models. The first group (G1) has four students: two students, S1 and S2, with high prior knowledge in physics and CT, and two students, S3 and S4, with low prior knowledge in both domains. S1 and S2 contributed the most during the three discussion sessions (30% and 25% respectively) while S3 and S4 contributed 3% and 13% of the total utterances. The instructor contributed 28% of total utterances. The second group (G2) has three students: S5 who has low prior knowledge in both domains, S6 who has high physics prior knowledge and low CT prior knowledge, and S7 who has low prior knowledge in physics but high prior knowledge in CT. Unlike G1, the instructor contributed most to the group discussions with 39% of the total utterances over the three days. S5 and S6 contributed 31% and 21% respectively while S7 contributed the least with 9%. As seen in previous work (Snyder et al., 2019), students with low prior knowledge, S3, S4, S5 and S7, particularly improved over the course of the study. We hypothesize this may be due to the effectiveness of the group discussions facilitated by the instructors. In the following analysis, we explore the different ways the instructor facilitated these discussions in each group.

### Instructor facilitation

We analyze the instructor-facilitated discussions utilizing the framework described above. Although facilitated by the same instructor, the role the instructor played is very different across the two groups. The role of the instructor in G1 discussions was primarily in the form of *technical support*: 38% of the utterances dealt with logistical and technical issues, 32% of utterances were *managerial support*, 26% *pedagogical support* and 4% *social support*. The role of the instructor in the G2 discussions was primarily *pedagogical support*: 46% of the utterances were classified as applications of physics, CT or synergistic learning process, 33% *technical support*, 11% *managerial suppor*t and 5% *social suppor*t. We hypothesize this difference in the support focus is due to the distribution of prior knowledge in each group. While the instructor was not aware of the prior knowledge of the students at the time of the discussions, the instructor's focus was impacted through the students' misconceptions and knowledge gaps that became apparent through their explanation prompting efforts, illustrated below.

### Explanation prompting episodes

The three vignettes shown in Tables 2-3 illustrate how our instructor facilitated successful group discussions during the computational modeling of physics phenomena. To support successful model construction and integration of STEM and CT concepts, the instructor utilized explanation prompting to identify key misunderstandings and knowledge gaps; and group management prompting to encourage students to share knowledge learned to support group computational model construction. We illustrate a common CT misunderstanding students have between the use of the set and change blocks in Table 2 over the course of two days. On Day 1, the instructor prompts S1 to check if they understand which block is appropriate for initializing variables. On Day 2, the student has an error in their model (Figure 3(a)) that is causing the truck to go from 0 m/s to 15 m/s instantaneously, instead of starting from rest and speeding up to 15 m/s.

**Figure 3**
*Computational models prior to instructor engagement for G1 (a) and G2 (b).*

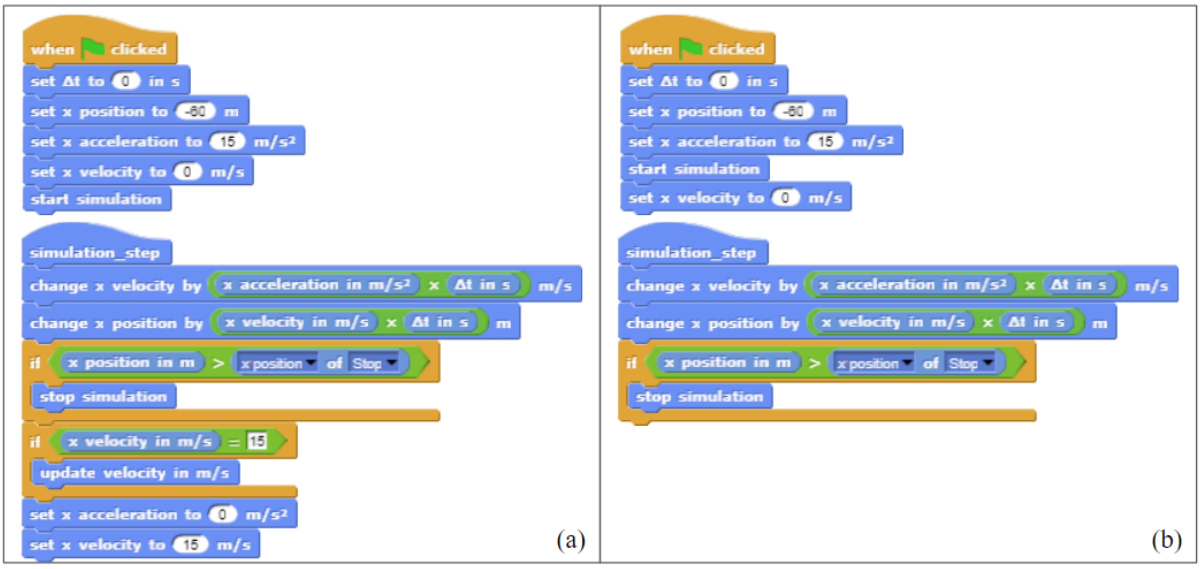



**Table 2**
*G1 truck discussion across two days*

| Utterance | Description |
|---|---|
| Day 1: *G1 is initializing needed variables for the truck stop task.* | |
| **Instructor:** What do you think...should you set or change the position in order to start at -15m*?* | **CT application**: The instructor is prompting the student to select the relevant block(s) for initializing the position of the truck. |
| **S1:** Set I think, I think it's set. I don't know, let me try. | *S1 correctly identifies and utilizes the appropriate block.* |
| Day 2: *S1 has already determined what behavior in the model does not follow physics principles.* | |
| **Instructor**: Why do you think the velocity stays at 15? | **Guiding**: The instructor guides the student, through questioning prompts, to identifying the CT error. |
| **S1**: I think because of this, this update velocity because I try to put if velocity equals meters per second goes 15. I tried to put these ones in there instead of the update last one. But it didn't do anything | *S1 correctly identifies that the update velocity block is incorrect and explains their debugging process implemented in the conditional. S1 is showing a lack of understanding in which block is used to update velocity and what the set block does.* |
| **Instructor**: I can see.... So one thing to remember is the set and the change. So under the green flag, you have that you're setting that initial value. And then for the simulation step, that's where you update the behavior. So that repeats every time (step), so at the end of your simulation block you have a set | **CT application**: After identifying the lack of knowledge, the instructor is explaining how the *set x velocity position* block, the last block in the simulation step, is going to set velocity to 15 m/s every step of the program. |
| **S1**: So I should make it change velocity | *S1 tries to confirm the way to debug their model.* |
| **Instructor**: You have it correct at the top though. But you just have some extra blocks. If that makes sense. | **CT application**: The instructor helps the student understand that they are correctly updating velocity with the *change x velocity* block but then are incorrectly setting velocity to 15 m/s. This setting of the velocity effectively overwrites the updating process. The instructor refers back to S2's model which was shown and discussed previously. |
| **S1**: I think so… | |
| **Instructor**: So you set the velocity after you change it. That what's happening | |
| **S1**: So set it and then change it. | |

| | |
|---|---|
| **Instructor**: If you remember with S2's code, and you'll want to set acceleration to zero but not the velocity because you're already doing everything right You're changing the velocity. Yeah. | |
| **S1**: So I set this back. | *S1 confirms that they are understanding correctly.* |

In the first episode (Table 2), the instructor first prompts the student in G1 about how they should implement the initialization of variables on Day 1. While S1 correctly identifies the block to use, they express uncertainty, which underscores a lack of knowledge about the differences in initializing variables versus updating variables as the simulation advances in time. This is elicited, through explanation prompting, by the instructor during the group discussion on Day 2. Due to the S1's further explanation of what is causing the issue with their model, the instructor can identify that the student does not have a physics misunderstanding but has a CT knowledge gap about the correct blocks to implement the updating of velocity.. It is interesting to note that while the group demonstrated success on the CT pretest, the instructor's prompting provided additional insight into students' understanding of the difficult CT concept (i.e., student difficulties in translating conceptual understanding of variables from the pretest to the construction of their computational model).

The second episode (Table 3) shows an excerpt from one of G2's discussions. This instructor-facilitated discussion begins with the introduction of a synergistic learning strategy, the use of data tools, and is followed by a discussion that switches between the application of physics and CT concepts. S5 has incorrectly modeled the speeding up of the truck to 15 m/s as seen in Figure 3(b).

**Table 3**
*G2 truck discussion*

| Utterance | Description |
|---|---|
| **S1:** And then, for the speed limit, I just changed set x acceleration. Instead of four, I changed it to 15. And that caused it to go 15 miles per second squared, and stop at the stop sign. And that's it. | S1 is describing how they tried to model the truck speeding up to 15 m/s. They have incorrectly initialized the acceleration value to 15 $m/s^2$. |
| **Instructor**: So click on the x position, there's a box right next to it, there's a checkbox. And then the x velocity below... Now try running your model...what do you guys see? What happened with the velocity? | **SL strategies:** By encouraging S1 to check the box next to x velocity, the instructor is helping the students monitor how the variable is updated in their computational model by using an inspection window that outputs the value. The instructor is encouraging the synergistic learning use of data tools to identify the error. |
| **S5**: Did it change? | *S5 runs the model and points out the changing velocity* |
| **Instructor**: So yeah, so when you have acceleration at 15, that's changing the velocity by 15 every time you repeat it. | **Physics application:** The instructor is explaining to the student that the velocity value is changing by 15 m/s at each time step, it is not being set to 15 m/s. |
| **Instructor**: If you...one thing I might switch here is… I'd just switch the x acceleration to... What was it...4?<br>**S1:** Yeah. So change it back to 4? | **Guiding:** The instructor is encouraging the student to initialize the acceleration value back to the given value of 4 $m/s^2$ after explaining that setting acceleration to 15 m/s does not make the truck speed up to 15 m/s. |
| **Instructor:** Yep. Okay. And then one other quick thing that we just never taught you guys, as if you can move the set x velocity up above the start simulation. | **Guiding:** The instructor is addressing another error in the model (initializing velocity to 0 m/s after the simulation step block). |
| **Instructor**: A computer is like a baby, it's just gonna go line by line and you did it right in that it is under the green flag. But what the computer does is it's calling the start simulation and then it's doing the velocity, and you just want it to do that one before the start. | **CT application**: The instructor is pointing out that the student is right to initialize velocity under the green flag however, through an explanation of how the model is going to be executed by the computer, is emphasizing it should happen before the *start simulation step* block. The student may use this to perform line by line debugging. |
| **Instructor**: Perfect. That's perfect. So now try, so you'll see and then stop. | **Guiding**: The instructor is encouraging S5 to see the behavior of the model by running it now that the two errors have been identified and fixed. |

| | |
|---|---|
| **Instructor**: Um, so you can see that it keeps going up and…. right about there is where you want to stop accelerating, right? | **Physics application**: Moving onto the next component of the task, slowing down the truck, the instructor is helping the students understand what the behavior of the model should be, according to physics principles. |
| **Instructor**: How would you guys do that? | **Guiding:** The instructor is encouraging the other two students in the group to contribute to the discussion. |
| **Instructor**: S7 or S6, you there? | **Social:** The instructor checks in with the other two students in the group who have been quiet. |
| **S6**: I'm a little bit lost because I thought that list is...like the initial one was right so I'm a little bit confused about what the changes we made are doing. I'm just trying to keep up the best as possible. Could you try and re-explain it for me, please? | *S6 expresses confusion about the identification of an error in the model and asks for clarification.* |
| **Instructor**: So actually S5, do you want to describe how you found what was wrong? Because you were, you're right with what was wrong with that first model? | **Guiding**: Instead of answering the question, the instructor encourages S5 to explain to S6. The discussion continues with S6 explaining the error to S5. |

In the second episode (Table 3), the instructor is helping the students debug their model using the data tools. After walking G2 through how to view the values of the variables as the simulation runs, the instructor then prompts the students to explain what they are seeing. There are two errors in the model: a physics-focused misunderstanding of the relationship between velocity and acceleration and a CT-focused misunderstanding of how the ordering of blocks affects the updating of variables. The instructor emphasizes how acceleration and velocity relate, as well as explaining how the computational model runs to promote CT understanding. In this discussion, the instructor integrates physics and CT, using a strategy of switching between CT and physics concepts. It has been shown that this switching between domains leads to more successful modeling processes (Snyder et al., 2019). We can also see the instructor use prompting to encourage better collaboration.

## Discussion

In the two episodes above, we illustrate how instructors support students' synergistic learning of STEM and CT concepts and practices. We look at how instructors *identify* and *address* students' misunderstandings and knowledge gaps. The instructor often utilizes the strategy of explanation prompting to identify student difficulties and gain additional insight into the students' knowledge that a pretest may not provide. This is illustrated in G1, where S1 is stronger in CT than physics according to their pretest scores but they require more CT support in the group discussions as seen in the first episode. We may assume that more of S1's difficulties will be physics-focused when looking at his/her pretest scores, however, our analysis shows that even students with high prior knowledge may face difficulties in translating that knowledge to the computational form (Basu et al., 2016).

In addition to identifying student difficulties, the above episodes illustrate the strategies used by the instructor to address student issues. In the second episode, we can see the instructor uses explanation prompting to point the students towards the error in their model, followed by guiding instructional statements that both explain the physics and CT errors in more detail and guide the students towards correcting the issues with the model. When another student expresses confusion over the instruction, instead of repeating the explanation, the instructor guides S5 to explain what was discussed. This prompting strategy allows the instructor to assess the impact of previous instruction on S5's knowledge gaps, and it encourages discussion between group members. In the continuing discussion, S5 correctly explains the CT issue but doesn't address the physics misunderstanding which leads the instructor to emphasize how velocity is incorporated into the model. We also see the instructor consistently assessing a possible student difficulty in the first episode. The instructor's revisiting of the difference between set and change blocks during the truck task leads to successful knowledge understanding by S1, as seen in their following drone tasks where they correctly implement initializing and updating variables.

## Conclusions and future directions

Students often have issues translating science concepts and relations into executable computational structures, and this could be because they have difficulty in mapping the science concepts to the scenario they are modeling or a problem in understanding the basic computational structures. While we can identify the domain where students may have difficulties by analyzing their pretest results, these assessments often have difficulty predicting students' difficulties in building their computational models. In this work, we see how *explanation prompting* can identify knowledge gaps and misunderstandings that traditional assessments, such as pretests, may not. Instructor-facilitated group discussions, utilizing strategies such as explanation prompting, create an opportunity where misunderstandings and knowledge gaps can be identified and addressed on an individual level.

While these supporting strategies help students in their model building tasks, future work will analyze the impact of different types of instructor support on students' knowledge construction processes. Other literature, e.g., Webb et al. (2009), showed instructor explanation prompting varied as they used it to identify and address student knowledge gaps while encouraging group participation. While the instructors did provide some of this social support, it was minimal compared to their domain focused explanation prompting. In future work, we will expand the social support students receive from instructors and study the impact. Limitations of our analysis include the small sample size. In the future, we aim to expand our analysis with multiple instructors of differing educational backgrounds and experience with CT and computational modeling and more diverse student groups, with an emphasis on better understanding the impact of the socio-cultural context of the research.

## References

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. Research and practice in technology enhanced learning, 11(1), 1-35.

Berge, Z. L. (1995). The role of the online instructor/facilitator. *Educational technology*, *35*(1), 22-30.

Chi, M. T., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. Educational psychologist, 49(4), 219-243.

Dillenbourg, P., Baker, M., Blaye, A. & O'Malley, C. (1996) The evolution of research on collaborative learning. In E. Spada & P. Reiman (Eds) Learning in Humans and Machine: Towards an interdisciplinary learning science. (Pp. 189-211) Oxford: Elsevier.

Emara, M., Hutchins, N. M., Grover, S., Snyder, C., & Biswas, G. (2021). Examining Student Regulation of Collaborative, Computational, Problem-Solving Processes in Open-Ended Learning Environments. Journal of Learning Analytics, 8(1), 49-74.

Fung, D. C. L., To, H., & Leung, K. (2016). The influence of collaborative group work on students' development of critical thinking: The teacher's role in facilitating group discussions. *Pedagogies: An International Journal*, *11*(2), 146-166.

Grover, S., Hutchins, N., Biswas, G., Snyder, C., & Emara, M. (2019). Examining synergistic learning of physics and computational thinking through collaborative problem solving in computational modeling. AERA, Toronto, CA.

Hutchins, N. M., Biswas, G., Maróti, M., Lédeczi, Á., Grover, S., Wolf, R., ... & McElhaney, K. (2020). C2STEM: A system for synergistic learning of physics and computational thinking. Journal of Science Education and Technology, 29(1), 83-100.

Kinnebrew, J., Segedy, J.R., & Biswas, G. (2017). Integrating model-driven and data-driven techniques for analyzing learning behaviors in open-ended learning environments. IEEE Transactions on Learning Technologies, 10(2), 140–153.https://doi.org/10.1109/TLT.2015.2513387

Rovai, A. P. (2007). Facilitating online discussions effectively. The Internet and Higher Education, 10(1), 77-88.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. Education and Information Technologies, 18(2), 351-380.

Snyder, C., Hutchins, N., Biswas, G., Emara, M., Grover, S., & Conlin, L. (2019). Analyzing students' synergistic learning processes in physics and CT by collaborative discourse analysis. In Proceedings of the international conference on computer supported collaborative learning (pp. 360–367). Lyon, France.

van Vondel, S., Steenbeek, H., Van Dijk, M., & Van Geert, P. (2017). Ask, don't tell; A complex dynamic systems approach to improving science education by focusing on the co-construction of scientific understanding. Teaching and Teacher Education, 63, 243–253.

Webb, N. M., Franke, M. L., De, T., Chan, A. G., Freund, D., Shein, P., & Melkonian, D. K. (2009). 'Explain to your partner': teachers' instructional practices and students' dialogue in small groups. Cambridge Journal of Education, 39(1), 49-70.

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—An embodied modeling approach. Cognition and Instruction, 24(2), 171–209.

Wilkerson-Jerde, M., Wagh, A. And Wilensky, U. (2015), Balancing Curricular and Pedagogical Needs in Computational Construction Kits: Lessons From the DeltaTick Project. Sci. Ed., 99: 465-499.