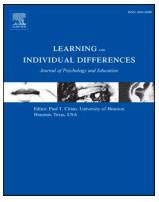
Learning and Individual Differences 121 (2025) 102724

Contents lists available at ScienceDirect

Learning and Individual Differences

journal homepage: www.elsevier.com/locate/lindif



Using collaborative interactivity metrics to analyze students' problem-solving behaviors during STEM+C computational modeling tasks☆

Caitlin Snyder [a,*], Clayton Cohn [b], Joyce Horn Fonteles [b], Gautam Biswas [b]

[a] University of Detroit Mercy, Detroit, MI, United States
[b] Vanderbilt University, Nashville, TN, United States

ARTICLE INFO

Keywords:
Collaborative problem solving
Computational modeling
Synergistic STEM+C learning
Multimodal learning analytics

ABSTRACT

Recently, there has been a surge in developing curricula and tools that integrate computing (C) into Science, Technology, Engineering, and Math (STEM) programs. These environments foster authentic problem-solving while facilitating students' concurrent learning of STEM+C content. In our study, we analyzed students' behaviors as they worked in pairs to create computational kinematics models of object motion. We derived a domain-specific metric from students' collaborative dialogue that measured how they integrated science and computing concepts into their problem-solving tasks. Additionally, we computed social metrics such as equity and turn-taking based on the students' dialogue. We identified and characterized students' planning, enacting, monitoring, and reflecting behaviors as they worked together on their model construction tasks. This study investigates the impact of students' collaborative behaviors on their performance in STEM+C computational modeling tasks. By analyzing the relationships between group synergy, turn-taking, and equity measures with task performance, we provide insights into how these collaborative behaviors influence students' ability to construct accurate models. Our findings underscore the importance of synergistic discourse for overall task success, particularly during the enactment, monitoring, and reflection phases. Conversely, variations in equity and turn-taking have a minimal impact on segment-level task performance.

Educational relevance and implications statement: The complexities of collaborative problem-solving for computational modeling in science provide a unique opportunity to explore individual and group learning. Specifically, in this manuscript, we examined differences in collaborative problem-solving behaviors, characterized by social and domain specific metrics, and their impact on groups' ability to complete computational modeling tasks in kinematics. We identified the impact of interactivity metrics, such as equity and turntaking, and the interweaving of science and computing concepts during collaborative discourse, on groups' performance. Finally, we analyzed differences between groups' planning, enacting, monitoring, and reflecting behaviors through interactivity metrics and students' segment-level performance. Our findings highlight key differences in students' problem-solving behaviors that will have implications in future work targeting adaptive support for problem-solving tasks.

1. Introduction

In recent years, there has been a growing emphasis on implementing technology-enhanced learning environments in secondary school classrooms to support problem-based learning (PBL) in STEM fields (Asghar et al., 2012; Jeong et al., 2019). In our work, we have leveraged the connections between science and computing (C) (NRC, 2012) to develop STEM+C curricula. These curricula integrate authentic computational modeling of scientific processes with related problem-solving tasks. While these environments have shown promise in supporting learning across multiple domains (Sengupta et al., 2013; Weintrop et al., 2016), researchers have also noted that they can introduce additional complexity, exacerbating students' difficulties in constructing and integrating knowledge during model-building tasks (Basu et al., 2016; Chi, 2008). We address these challenges by encouraging students to collaborate, explore, and develop ideas while constructing and evaluating their solutions for complex computational modeling tasks. Collaborative problem-solving (CPS) approaches have significantly

☆ This article is part of a Special issue entitled: 'Learning in the Digital World' published in Learning and Individual Differences.

* Corresponding author.
E-mail address: snydercr@udmercy.edu (C. Snyder).

https://doi.org/10.1016/j.lindif.2025.102724
Received 30 September 2023; Received in revised form 28 April 2025; Accepted 12 May 2025
Available online 2 June 2025

enhanced student learning (Beers et al., 2005; Sears & Reagin, 2013). Previous research shows that collaboration fosters the development of shared knowledge and improves problem-solving behaviors (Dillenbourg, 1999; Roschelle & Teasley, 1995).

Previous studies conducted by our team have demonstrated that open-ended, technology-enhanced problem-solving environments create a shared situational context that promotes genuine collaborative problem-solving in STEM+C domains (Hutchins et al., 2020; Snyder et al., 2019; Snyder et al., 2024). This research examines students′ collaboration behaviors in the C2STEM environment (Hutchins et al., 2020) as they engage in model construction and debugging activities while building computational models kinematics. We analyze students′ learning and collaborative behaviors using the planning, enactment, and reflection framework from self-regulated learning (SRL; Schunk & Zimmerman, 1998). This framework is applied specifically to analyze students′ socially shared regulation of learning (SSRL) processes. We chose this framework over others (e.g., Hadwin et al., 2011) due to the complexity and open-ended nature of the STEM+C PBL computational modeling tasks.

Given our study′s small sample size, this paper does not seek to draw broad conclusions about the connections between groups′ learning and CPS behaviors. Instead, we present an exploratory analysis that characterizes the relationships among students′ collaborative interactions, their problem-solving behaviors, and model-building performance. This analysis, conducted through the lens of multimodal data (i.e., student discourse and log file data), enhances our understanding of students′ productive and unproductive CPS behaviors in authentic STEM+C PBL contexts.

More specifically, we study the social aspects of students′ collaborative dialogue using *turn-taking* and *equity measures* and the *synergistic* nature of their domain-specific conversations. Studying the synergistic nature of students′ conversations helps us understand how they develop and combine their science and computational knowledge to support their model-building and debugging activities (Hutchins et al., 2020; Snyder et al., 2019). Using this exploratory analysis framework, we leverage log data and discourse summaries to analyze segment-level correlations between social interaction measures—such as equity and turn-taking—and model-building performance. Next, we broaden our analysis to explore students′ planning, enacting, monitoring, and reflecting behaviors to better understand how these metacognitive behaviors relate to their interaction process measures and their success (or lack thereof) in model construction tasks.

The remainder of this paper is structured as follows: Section 2 reviews existing research in collaborative learning and delineates how our study extends this research within STEM+C domains. Section 3 details our research questions and analytical methods, including a description of our C2STEM environment, curriculum, and the research study conducted in a high school STEM classroom. Section 4 presents the analyses to address our research questions. Finally, Section 5 provides the conclusions and suggests directions for future research.

2. Background

Collaboration is an important learning process, promoting deeper thinking and developing advanced problem-solving skills (NRC, 2012). Roschelle and Teasley defined collaboration as "*coordinated, synchronous activity that arises from a continuous effort to construct and maintain a shared understanding of a problem*" (Roschelle & Teasley, 1995, p. 70). Successful collaboration depends on active contributions and coordination among group members, as well as effective social interactions to foster a *shared understanding*, which aids in *knowledge co-construction* and problem-solving (Larkin, 2006; OECD, 2015). Key interaction skills for effective collaboration include making and promoting contributions, translating ideas into problem-solving steps, monitoring progress, reflecting on results, and providing constructive feedback through argumentation and explanation (Garrison & Akyol, 2013; Grau & Whitebread, 2012).

Related to STEM learning practices, the Next Generation Science Standards emphasize collaboration-related processes like argumentation and information communication as essential for science education (NGSS, 2013). Simultaneously, the NGSS recognizes the growing connections between science and computing and acknowledges computational thinking as a key science practice (Grover & Pea, 2013; Wing, 2006). Computing in science learning (i.e., STEM+C) has been actualized through inquiry tasks and computational modeling (e.g., Hambrusch et al., 2009). There has been substantial research on the benefits of learning science through computational modeling (diSessa, 2001; Sengupta et al., 2013; Sherin et al., 1993). However, studies have also documented challenges that students encounter in such settings, such as the difficulty of translating science disciplinary knowledge and mathematical relationships into computational forms for model building (Basu et al., 2016).

Measuring students′ collaboration in integrated science and computing curricula necessitates tracking the concepts and practices they apply across both domains during their problem-solving tasks Sengupta et al. (2013). While researchers have explored the multidimensional nature of collaboration for university students engaging in complex tasks (e.g., Nasir et al., 2021; J ¨arvela ¨ et al., 2020), further research is needed to deepen our understanding of how collaborative processes influence learning for K-12 students. For this study, we utilize CPS, capitalizing on real-time problem-solving discussions among students to enhance our comprehension of STEM+C PBL.

In our K-12 STEM+C learning environment, students engage in complex computational modeling tasks in an open-ended setting. Regulatory processes like planning, enacting, and reflecting are crucial in these problem contexts (Azevedo et al., 2010). Research highlights the importance of planning in complex tasks, where students set goals, break them into manageable sub-goals, formulate plans, and identify execution strategies (Eichmann et al., 2019). While this framework is frequently used to study individuals′ self-regulated learning, there is an added layer in CPS contexts: students must cultivate a shared understanding of the goals, reach a consensus on strategies, and navigate their differing knowledge backgrounds (Zimmerman & Moylan, 2009). In STEM+C contexts, students must also manage complexities by decomposing tasks, sharing responsibilities, and elaborating on plans and strategies during problem-solving activities.

Reflection processes are vital in Problem-Based Learning (PBL) (Barrows et al., 1980; Hmelo-Silver, 2004). Previous research has emphasized learning cycles and students′ adaptation across these cycles as they reflect on their learning processes (Rakovi ´c et al., 2022). Consequently, researchers have focused on supporting and evaluating reflection behaviors (Carpenter et al., 2021). In this study, we consider students′ reflection behaviors after they complete parts of a complex task and after they complete the entire task (Schon & DeSanctis, 1986). Furthermore, students often track their progress while engaging in learning and problem-solving tasks (Schwartz et al., 2009). Debugging strategies are essential for building correct computational models. In collaborative settings, students may use their shared understanding of the problem to monitor and debug their models, pausing for reflection activities to evaluate their evolving solutions (Kalina & Powell, 2009; Stahl & Hesse, 2009).

Researchers have highlighted the benefits of leveraging multimodal analysis to better understand students′ cognitive and metacognitive behaviors (e.g., J ¨arvela ¨ et al., 2021). However, this requires aligning and interweaving multiple data modalities, such as aligning students′ conversations with their activity data collected in log files (Wise et al., 2021). Research on multimodal learning analytics (MMLA) has been focusing on these efforts (Blikstein & Worsley, 2016), and recent calls for an advanced understanding of how to leverage MMLA for collaboration analysis have highlighted the need for improved methods that support actionable analysis (Wise et al., 2021). For example, multimodal analysis may require segmenting the data into analyzable and actionable

chunks. Researchers have previously leveraged learning-adjacent data at set time intervals, such as every 30 s. Such arbitrary choices can significantly impact the analysis of students' learning behaviors, particularly because they do not leverage specific educational contexts (Knight et al., 2017). To harness the benefits of collaborative learning and promote productive collaboration to support STEM+C learning, we developed and adopted a *contextualized time segmentation* approach (Snyder et al., 2024) to analyze students' model-building and debugging activities.

In summary, we expand previous research on collaborative learning in STEM+C computational modeling tasks using an exploratory multimodal approach to (1) employ context-targeted segmentation methods to analyze students' collaborative problem-solving behaviors and (2) integrate interactivity and domain-specific metrics to gain deeper insights into K-12 students' collaborative learning behaviors.

3. Research Questions

In this paper, we address the following research questions:

RQ1. How do interaction measures, such as equity and turn-taking, and the synergistic dialogue measure relate to students' model-building performance? and

RQ2. How do students' collaborative problem-solving behaviors, such as planning, enacting, monitoring, and reflecting, relate to their interaction process measures and ability to construct computational models in kinematics?

To address RQ1, we conduct a fine-grained, segment-level correlation analysis of group interactivity and synergistic dialogue using segment-level model scores. Our segmentation method, derived from collected log data, enables the analysis of students' model-building and debugging activities within specific task contexts. We examine segment-level performance across groups to determine how it relates to their synergistic and collaborative dialogue.

To address RQ2, we broaden our analysis of the interaction metrics to uncover links between the metrics and groups' collaborative problem-solving behaviors. Our study includes manual coding of large language model (LLM)−generated summaries of group discussions and students' actions during each segment. Based on the interaction metrics and segment-level success, we examine students' planning, enacting, monitoring, and reflecting behaviors. These findings enhance our understanding of collaborative problem-solving behaviors and their relationship to students' abilities in building computational models.

4. Methods

This section describes our STEM+C learning environment, measures for studying collaborative interactivity, our research study in a high school classroom, and our analysis methods.

4.1. C2STEM learning environment

Our block-based programming environment, C2STEM, illustrated in Fig. 1, helps students learn their science and computing concepts and practices (Hutchins et al., 2020). The environment provides students with domain-specific modeling (physics, in this case) blocks and additional computational blocks. Students can drag and drop blocks from the list provided on the left onto the script area to build their computational model (Hutchins et al., 2020). The model can be simulated to observe the behavior of the object(s) on the stage. Students using the C2STEM system can develop both partial and complete models, then simulate these models to see how objects move and how the related variables change over time. Students using the C2STEM system can develop partial or complete models and then simulate these models to see how objects move and how the related variables change over time.

Students typically analyze and debug their evolving models by assessing the motion of the object(s) on stage. The environment provides resources that help students evaluate and debug their models. In addition to animation and variable inspection functions displayed on stage, students can access graphing and table tools where selected variable values are plotted at each simulation step during a simulation run. For example, when variables such as x-position and x-velocity are selected for display, the simulation run generates position-time and velocity-time graphs, as shown in Fig. 1. Students can also use the table tool, which is updated with the current x-position and x-velocity of the object at each time step. The graphs and tables assist in interpreting the motion variables in relation to relevant physics concepts and laws, such as the

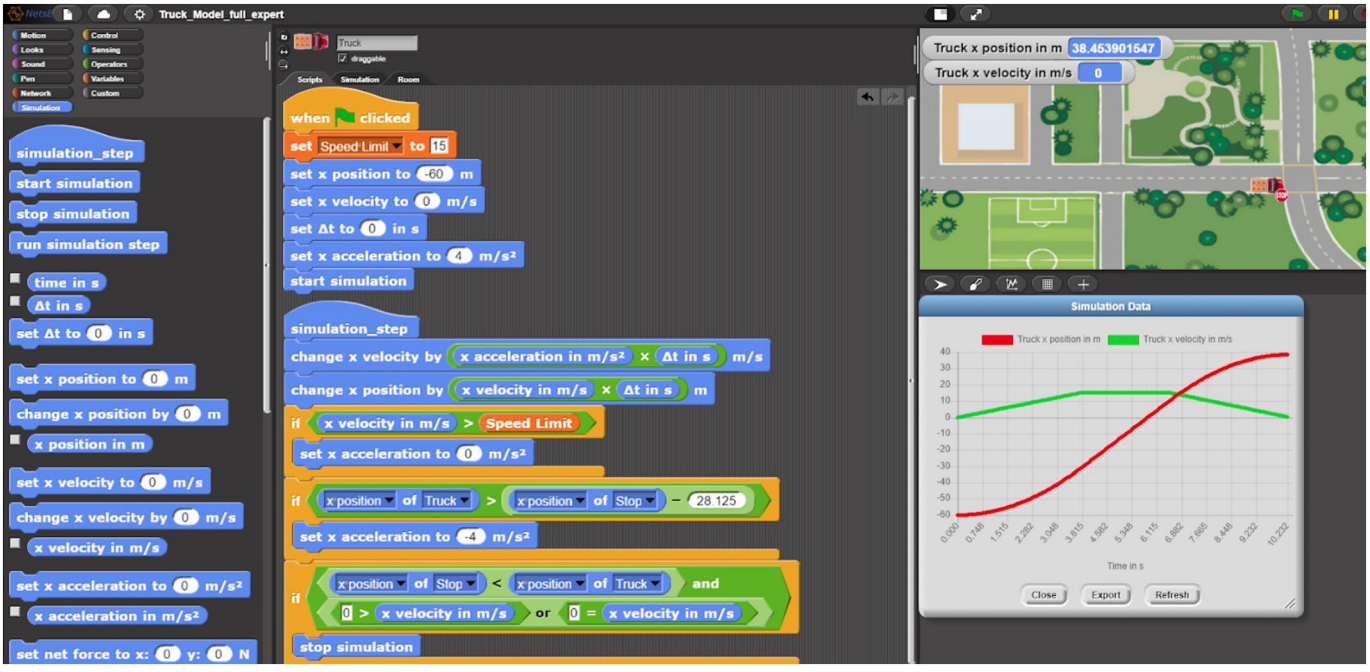


Fig. 1. CSTEM environment.

relationship between velocity and acceleration.

Students can also use these features to assess and reflect on the correctness of their models. From a computing perspective, these tools support data analysis and debugging that are known to be key computing and model-building practices (Grover et al., 2018; Weintrop et al., 2016). C2STEM also facilitates collaborative problem-solving as students leverage these tools to co-construct knowledge (Hutchins et al., 2021; Snyder et al., 2019).

Our curriculum adopts a modular and systematic evidence-centered design (ECD; Mislevy & Haertel, 2006), where each module consists of a sequence of inquiry and computational modeling tasks with accompanying formative assessments. As the students progress through the curriculum, we decrease the amount of instructional scaffolding. For example, we initially provide step-by-step instructions to illustrate problem-solving steps, but these scaffolds are withdrawn in later tasks. The open-ended model-building tasks provide very little scaffolding, requiring students to plan, develop, and implement problem-solving approaches, such as decomposing the problem to build the model in parts.

In this study, we focus our analysis on the first open-ended challenge task in this curriculum assigned to students in week 3 of the study. Students were instructed to use the domain-specific modeling blocks to construct a simulation model of the truck's motion, starting from rest, accelerating to the speed limit, cruising at that speed, and then decelerating to stop at a *STOP* sign. Unlike earlier instructional tasks that students worked on in weeks 1 and 2, they did not receive step-by-step instructions for solving this task. Instead, they were provided with a few hints; for example, it was suggested that they use conditional statements to model the behavior changes of the truck as it sped up, cruised, slowed down, and eventually stopped at a *STOP* sign. They were given a hint at the appropriate point about using kinematic equations to calculate the look-ahead distance, i.e., the distance from the stop sign at which the truck needed to begin slowing down. In this paper, we specifically analyze students' activities and behaviors in the open-ended challenge task to gain better insights into their collaborative interactions and problem-solving behaviors during open-ended STEM+C learning.

4.3. Research study and participants

Our research team conducted a two-month-long study, working two hours per week with 14−15-year-old 10*th* grade high school students in a STEM program hosted by a university in the Southeastern United States. The students had varied backgrounds in computing. Some had completed a high school programming class, whereas others had no formal programming experience. None of the students had taken a high school physics course, but some had been introduced to basic kinematics in introductory science classes.

For the study, students were divided into 13 groups (one triad and 12 dyads). The triad consisted of one student who did not consent to data collection procedures, so we did not analyze data from this group. The consenting students in the dyads were paired based on prior research purporting the benefits of heterogeneous prior knowledge pairs (e.g., Zhang et al., 2015). The student with the highest total pretest score (i.e., the sum of their pretest scores in kinematics and computing) was paired with the student who had the lowest pretest score, and so on. Each student dyad worked together on a single laptop with a shared mouse and keyboard. Before students started working in the C2STEM environment, there was a class discussion on good collaboration practices. However, the students were given no specific instructions on how to work together. They worked on the kinematics curriculum for two hours each week for eight weeks. The data reported in this paper is from their work on one of three kinematic challenge problems − a one-dimensional accelerated motion challenge task. Our data collection procedure was approved by our university Institutional Review Board. This included collecting summative pre- and post-test assessment data, logged actions in the C2STEM environment, the final computational models constructed by each group, and video and audio data using laptop webcams and OBS software. Student actions were recorded in log files with timestamps. Student conversations were automatically transcribed using Otter.ai™, which produced diarized transcripts. Two research team members then edited these transcripts for clarity and accuracy. Three dyads were excluded from our analyses because of problems with audio data collection during the study.

4.4. Data analysis methods

Our data analysis procedures included three key components: (1) Measures to quantify students' collaborative interactions as they worked on their learning tasks; (2) context-targeted segmentation of time-aligned multimodal data; and (3) LLM-generated summaries from the conversations extracted from each time-aligned segment. We discuss our analysis methods in greater detail below and describe how these methods are combined to understand students' collaborative problem-solving behaviors. Overall, this work analyzes **276 problem-solving segments** consisting of **2786 utterances** and **2275 actions** in the C2STEM environment that occurred over 9 h of problem-solving (approximately one hour per group).

4.4.1. Measuring collaborative interactions during computational modeling in science

We measure collaborative interactions by characterizing students' conversations along three dimensions: (1) *social*, where students interact with their partners to generate a common understanding of the problem-solving task (Jeong & Chi, 2007); (2) *domain-specific*, where students work together to acquire and combine their science and computing knowledge (Snyder et al., 2019); and (3) *knowledge application performance*, where we evaluate students' STEM+C learning and progress in their model building.

Social Measures of Collaboration Communication among group members is important for successful collaborative learning and effective communication requires contributions from all group members (Rummel et al., 2009). We adopt two interactivity measures to study the *social* aspects of students' collaborative problem-solving:

1. *equity* (EQU) in students' dialogue, i.e., the balance in the amount that each student contributes to the conversation; and
2. *turn-taking* (TT), i.e., how much do students respond to each other's statements and questions as they work together?

Equity measures *symmetry* in students' conversations, and helps them to negotiate differing perspectives and achieve common understanding (Meier et al., 2007). For students working in pairs, the equity measure was calculated by evaluating the number of utterances made by each student using the following formula:

$$1 - abs\left(\frac{\#utterances_{S1} - \#utterances_{S2}}{max(\#utterances_{S1}, \#utterances_{S2})}\right), \quad (1)$$

where $S1$ and $S2$ represent the two students who worked together. The computed value is in the range $[0, 1]$, where a value closer to 0 indicates more inequity (i.e. one student spoke more utterances than the other during the segment) and a value closer to 1 indicates greater equity (i.e., the students had relatively equal number of utterances during the segment) in the conversations between the students.

Turn-taking promotes back-and-forth conversations among students, fostering shared understanding through question posing, explanation, and argumentation (Jeong & Chi, 2007; Soller, 2001). The turn-taking measure quantifies the number of times the students switch speakers during conversation segments. For example, if the speaking pattern is S1, S1, S2, S1, S2, then there are three switches, while a speaking pattern of S1, S1, S1, S2, S2 has only one switch. The turn-taking measure for each segment is computed as:

$$\frac{\#utterance\ switches}{\#utterances - 1} \quad (2)$$

A value of 1 indicates that the students alternated by switching speakers during a time segment. Conversely, a value of 0 indicates that one student spoke for the entire duration without the other contributing to the conversation.

4.4.2. Domain-Specific Measure

We extend our analyses by measuring the *synergistic content*, i.e., the interleaving of science and computing concepts in the students' dialogue (Hutchins et al., 2020; Snyder et al., 2019). Two researchers hand-coded approximately 20 % of the students' conversation segments based on the physics and computation concepts they discussed. They achieved a Cohen's kappa value of 0.83 through back-and-forth discussions. Using this coding scheme, we calculated a synergistic score for each segment using the following formula:

$$1 - abs\left(\frac{\#utterances_{Computing} - \#utterances_{Physics}}{max\left(\#utterances_{Computing}, \#utterances_{Physics}\right)}\right), \quad (3)$$

This computed value is in the [0, 1] range, where a value closer to 0 indicates low synergistic discourse (i.e., most utterances in this segment focused on one domain). A value closer to 1 indicated high synergistic discourse (i.e., conversations in this segment included concepts in both domains).

Knowledge Application Measures Students' overall STEM+C learning was computed by scoring their final models with a pre-defined rubric generated using a systematic evidence-centered design approach (ECD; Mislevy & Haertel, 2006). The rubrics are discussed in (Hutchins et al., 2020). The groups' overall model-building performances (PERF), normalized to a [0,1] score, are listed in column 2 of Table 4 in Section 4.1.

We also evaluated their model-building progress by scoring the students' current computational models at the segment level (the segmentation method is discussed in Section 3.4.2 below). We hand-coded their model representations on a qualitative scale into the following categories: (1) *Consistent progress*, if at the end of the segment, there were no errors in the model and the students had added to their previous model; (2) *Some progress*, if at the end of the segment, there was at least one less error in the computational model as compared to the model in the previous segment and they may or may not have added to their computational model; (3) *No progress*, if at the end of the segment, the students had added to their computational model but they did not fix errors from the previous segment; and (4) *Backward progress*, if at the end of the segment, the students had added new errors to their model. These qualitative scores were mapped on a quantitative scale of 0−3 for analysis, with 0 corresponding to backward progress, 1 corresponding to no progress, 2 corresponding to some progress, and 3 corresponding to consistent progress.

4.4.3. Context-targeted segmentation

In contrast to conventional multimodal data segmentation methods that rely on learning-adjacent techniques such as predefined time segments, we devised a novel segmentation approach tailored to our specific problem context. We first encoded students' computational model-building actions into abstract syntax trees (ASTs). ASTs are conventional tree-based representations used in compilers to delineate the syntactic structure of computer programs (Grosch & Emmelmann, 1990). Leveraging sub-trees from these ASTs, we categorized students' model-building actions, which included adding, removing, adjusting, moving, and populating blocks into the following high-level categories: (1) *initialization* of relevant variables; (2) *variable updating* in the simulation loop to capture the dynamic behavior of the system; (3) *conditional statements* that primarily captured changes in the dynamic behavior; and (4) *variable updating governed by specific conditions* that were linked to the conditional statements. We illustrate the four high-level categories with example code shown in Fig. 2.

A segment ends when students' model-building actions switch from one of these four categories to another. For example, if a group first added blocks to create a conditional statement that would check if the truck should be in cruising mode (i.e., *if x velocity > 15 m/s*; see Fig. 2), this would indicate a problem-solving context focused on conditions to model the cruising mode. Next, if the students added the block, *set x acceleration to 0 m/s*^{2}, the earlier segment classified as a *conditional statement* segment would conclude, and a new segment, *initialization*, would begin.

Segment categorization only uses the log data. However, the log data provides no information on why the students switched context and what they planned to do after introducing the context switch in their computational model. On the other hand, the conversations the students had just before and during this segment provided us with much more information about students' problem-solving behaviors. For example, assume a group made a plan to model the motion of the truck cruising. To implement this, they added the conditional statement, and the subsequent conversation implied that they realized they had forgotten to initialize a variable, i.e., set the *x acceleration to 0 m/s*^{2}. More generally, this segmentation and categorization helps us align the multimodal data (logs and discourse) into segments in time and then use LLMs to summarize their conversations in each segment. This presents a systematic approach to understanding and interpreting students' behaviors in each segment of their model-building task. In our study, the average length of a segment was one minute and 49 s.

We used additional information from the logged actions to classify each extracted model building segment into *construction* or *debugging* episodes. For construction episodes, students added blocks and completed fields associated with the blocks in their current model. For debugging episodes, students reviewed code that they had already created. Debugging episodes often included actions like 'run simulation' and/or 'use the data tools'. In addition, any adjustments, removal, additions, and moving of blocks after the model was assessed were also classified as part of debugging episodes. As an example, if a group started working on a conditional statement that modeled the truck motion changing from cruise motion (constant velocity) to a slow-down motion to make it stop at the stop sign, all the model-building actions connected to creating the new conditional statement were classified as part of a construction episode because the students were adding new constructs to their model. On the other hand, if the students ran a simulation after creating the slow down to stop motion and made changes to the blocks in that part of their code, then the associated actions were classified as part of a debugging episode since students were making adjustments or changes to an already existing part of their computational model.

4.4.4. LLM-generated discourse summaries

To analyze students' segment-level problem-solving, we processed group discourse for each segment generated (see Section 3.4.2) using GPT-3.5, a transformer-based large language model (LLM) (Vaswani et al., 2017) linked to ChatGPT.[1] The LLM summarized student conversations, aiding in characterizing specific segments into planning, enacting, monitoring, reflection, off-task, and other categories. These summaries also enhanced our understanding of how students navigated through debugging processes when they encountered errors. However, this analysis approach is inherently limited for three primary reasons: (1) the LLM model was pre-trained and it was not possible to use conventional approaches to fine-tune the deep learning model using re-training methods; therefore, summary generation was a "black box" process; (2) we ran into token limitation problems when processing

[1] https://openai.com/blog/chatgpt

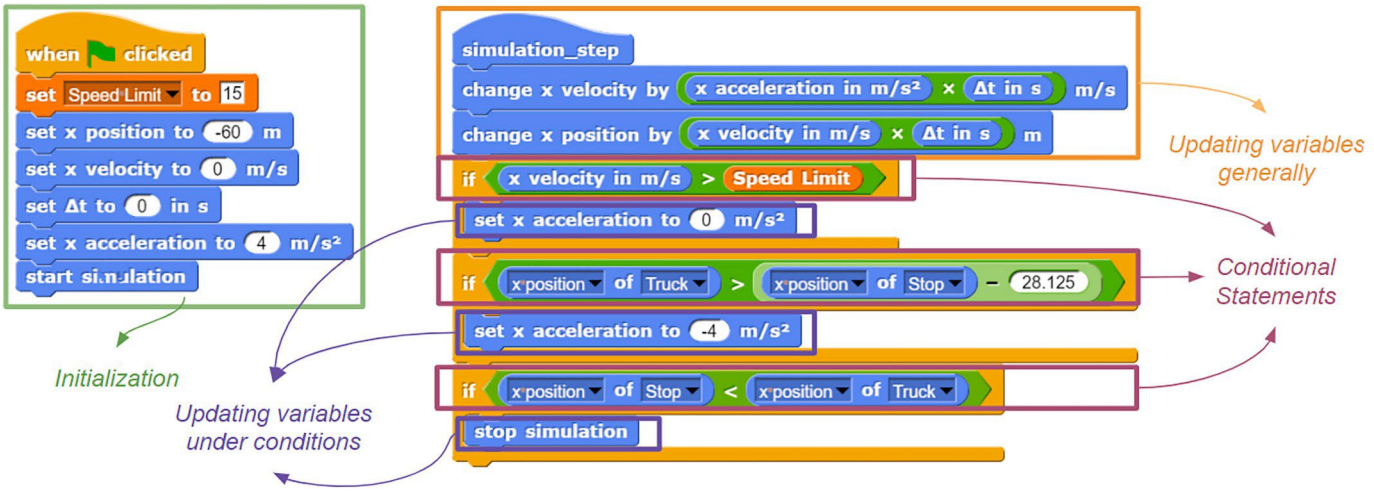


Fig. 2. C2STEM task-specific context used for segementation.

conversation segments; and (3) the lack of knowledge persistence (across segments) limited our ability to capture the temporal evolution in student thinking across multiple segments.

To overcome these limitations to some extent, we developed an initial process that involved two steps: (1) exploring the knowledge retrieval and generation scope of the LLM and (2) engineering and utilizing existing prompt patterns so that GPT 3.5 could generate reasonably accurate LLM summarizations to answer RQ2. Table 1 outlines this human-in-loop training framework. The exploration phase targeted two key processes: (1) *Exploration of the LLM Scope* and (2) *Learning Output Testing*. During *Exploration of the LLM Scope* we leveraged our collaboration metrics (discussed in Section 3.4.1) to develop prompts targeting key collaboration constructs. This included LLM-generated summaries of student discussions in physics and computing domain knowledge to support their computational modeling task. During the human-in-the-loop component, the research team reviewed results, memoed key issues in the summaries generated, and iteratively refined the prompts until identified issues were minimized and researcher consensus on the quality of the summaries was achieved. This approach involved reviewing every summary and compared it to the discourse it was summarizing to ensure that the summaries accurately reflected the discourse during that segment. This process is illustrated in Fig. 3. The different prompts and notes for each iteration can be found in the Appendix. These summaries were also used in work focused on the human-in-the-loop component (Cohn, Snyder, et al., 2024).

As an example, during the *Exploration of LLM Scope*, the initial LLM summarization process did not recognize key conceptual knowledge components, such as the *lookahead distance* (the distance from the stop sign when the truck needs to slow down to come to a stop) and how it was calculated. We addressed this more generally, by asking ourselves the question: *Could the LLM generate the solution for the problem task assigned to the students*? This resulted in engineering a prompt pattern we called Code Generation during the *Learning Output Testing Process*. During testing, we identified that the LLM needed the problem description, domain-specific knowledge, and context for generating acceptable solutions to the computational modeling task.

In the engineering phase of our human-in-the-loop framework, we employed a testing approach that built upon existing prompt engineering methodologies (Cohn, Hutchins, et al., 2024; Marvin et al., 2023; Schmidt et al., 2024; White et al., 2023). This phase was characterized by two principal processes: (1) *Input Semantics Generation*, where we developed a Meta Language to enhance the Large Language Model's (LLM) comprehension of tasks and discourse, and (2) *Output Customization*, which aimed to tailor the LLM's output to address specific research questions, such as RQ2. The Input Semantics Generation involved refining the metalanguage to improve the LLM's performance in computational modeling tasks. Subsequently, in the Output Customization phase, we integrated a *Code Generation prompt* pattern to facilitate the generation of precise summaries of problem-solving behaviors within groups. An illustration of this is the "lookahead distance" concept, where the inclusion of detailed problem task information enabled the LLM to accurately detect and interpret discussions about the distance calculation for a truck's deceleration in student conversations.

Overall, this two-step process resulted in a prompt with the following components (see *Appendix* for the full prompt):

- a *context manager pattern* (allowing for control of the context of the LLM's output (Cohn, Hutchins, et al., 2024; Snyder et al., 2024; White et al., 2023) that described the model-building task and incorporated the components such as the Code Generation pattern described in our example;
- a *persona pattern*, described by White et al. as an approach that "gives the LLM a persona or role to play when generating output" (p.4, 2024; 2024; 2023), in which we indicated to the LLM that it would play the role of a teacher trying to interpret the students' conversations; and
- a *task-specific pattern* that indicated which of the four task-specific segment types a particular group was working on and what that meant in the model building context (e.g., if the segment was labeled as initialization, the prompt stated: "In this segment, the students are working on assigning initial values to variables, such as position and velocity of the truck.").

The prompt concluded with an input semantics statement outlining the transcript's format.

4.4.5. Analyzing planning, enacting and reflecting behaviors using interaction measures and segment-level performance

The LLM-generated summaries provided an overview of group problem-solving behaviors during each segment. To extract and analyze these behaviors, we hand-coded every summary based on prior SRL research (White et al., 2009; Winne, 2010; Zimmerman & Moylan, 2009) outlined in Section 2, utilizing the coding scheme in Table 2. Additionally, we categorized segments into further categories: (1) students received help from the researcher, (2) students engaged primarily in off-topic discussions, and (3) students performed actions without any discussion at all. Although we considered leveraging the LLM to simultaneously summarize and code the segments, we opted for hand-coding the summaries in conjunction with our human-in-the-loop iterative prompt design approach (described above in Section 3.4.3) to validate

Table 1
Framework for prompt generation to support context-based segment summarizations.

Process	Description	Example
Exploration of LLM Scope	Testing preliminary prompt patterns to identify deficiencies in LLM knowledge base and retrieval	An initial prompt was generated for ChatGPT to summarize conversations between students that were working collaboratively in a computer-based learning environment. The input context included information about the learning environment and the domain-specific problem the students had to solve. This process was supported by the *Fact Check List* and *Reflection* (Cohn, Hutchins, et al., 2024; White et al., 2023) patterns that prompt the LLM to provide the rationale behind its output.
Learning Output Testing	Generating the input knowledge needed for LLM to produce a correct learning task solution (i.e., what knowledge is needed to solve the same problem as students)	Utilizing an iterative process, we developed a combination of inputs that resulted in the LLM generating a correct solution for the problem task. We developed a *Code Generation* pattern that helped the LLM to solve the computational model problem assigned to students. As a result, the LLM learned to summarize the processes that students were utilizing for building that segment model.
Input Semantics Generation	Meta language creation based on the previous two tasks to maximize LLM′s understanding of input and ability to process input content	Based on the prompt pattern for Meta Language Creation (Cohn, Hutchins, et al., 2024; White et al., 2023), we provided ChatGPT with information to better understand the semantics of the input so that it could generate adequate output. As an example, we identified individual student utterances using the label *SPEAKER: DISCOURSE* and prompted ChatGPT to summarize each student′s contribution to the conversation.
Output Customization	Leveraging a combination of prompt generation patterns from literature (e.g., Cohn, Hutchins, et al., 2024; White et al., 2023) to adapt to the requirements of the current research study (i.e., what is the research goal supported by the LLM-generated summary?)	We used an iterative prompt generation process leveraging previously identified prompt patterns supporting output generation to best target our research goal of relating group problem-solving behaviors with their ability to complete the computational modeling task (RQ2). Our experiments produced a prompt that combined the *Meta Language Creation* pattern for input semantics, the *Persona* pattern, and the *Context Manager* patterns based on a manual review of summary outputs by two authors for each prompt iteration.

the consistency of the LLM-generated summaries. In the future, we plan to use the hand-coded labels to fine-tune our LLMs to automatically recognize learning behaviors. To evaluate these students′ problem-solving behaviors, we analyzed them using the synergistic, turn-taking, equity, and segment-level performance measures described in Section 3.4.1. In this way, we could better understand these problem-solving behaviors in the context of students′ synergistic and collaborative interactions and their segment-level performance.

5. Results

This section outlines our method for addressing the two research questions and reviews the findings of our analyses.

5.1. RQ1: How do interaction process measures, such as equity and turn-taking, and the synergistic dialogue measure relate to students' model-building performance?

To answer RQ1, we applied the segmentation method described in Section 3.4.2 and calculated the equity (EQU), turn-taking (TT), and synergistic (SYN) scores for the 276 model construction segments across all nine groups. To provide an overview of the groups, we computed the scores for each group by averaging across all of their segments (see Table 4).

We analyzed the segment-level behaviors of groups by correlating the values of interactivity metrics with performance scores and effort (i.e., total number of actions) at the segment level. Given the small number of groups, we assumed a non-normal distribution and calculated the Spearman rank correlation to assess the relationships between segment performance measures and collaborative interaction metrics (SYN, TT, and EQU). Note that we verified the independence of observations assumption was not violated by calculating the intraclass correlation using a two-way mixed model for each measure: *PERF* ICC = 0.005 (95 % CI: −0.07, 0.20), *ACT* ICC = −0.086 (95 % CI: −0.11, −0.002), *EQU* ICC = 0.007 (95 % CI: −0.06, 0.23), *TT* ICC = 0.013 (95 % CI: −0.60-0.24), *SYN* ICC = −0.061 (95 % CI: −0.01, 0.07). The fact that these correlations range between −0.1 and 0.1 implies that the independence of observations assumption holds for this dataset.

Table 3 shows the correlation between the three collaborative interaction measures and two performance measures: (1) the final model score (PERF), and (2) the total number of actions performed to build the model (ACT, as a measure of their effort). We found a low correlation between performance and effort ($\rho = 0.1$), indicating a weak link between the number of actions students took and their progress in model construction at the segment level. Conversely, Table 3 reveals that synergistic discourse was moderately correlated with performance ($\rho = 0.42$). The data supports previous findings regarding the crucial role of synergistic (combined physics and computing) dialogue in computational modeling performance (Snyder et al., 2019). Both turn-taking (TT) and equity (EQU) showed a weak positive correlation with performance, with $\rho = 0.05$ and $\rho = 0.08$, respectively. Notably, groups typically executed more actions during segments with greater turn-taking, evidenced by a weak positive correlation ($\rho = 0.31$). This relationship is also reflected in the weak correlation between actions and equity ($\rho = 0.27$). These findings imply that equitable engagement and turn-taking may promote exploratory behavior but do not lead to model-building progress. We will investigate these relations in more depth in future work.

Table 4 presents the aggregated summary statistics, including metrics for synergistic dialogue (SYN), turn-taking (TT), and equity (EQU) across all nine groups. The table organizes the groups from the highest performer (*PERF* = 0.97) to the lowest (*PERF* = 0.42) to facilitate comparisons between performance and the synergistic and collaborative measures. The higher-performing groups generally had more synergistic dialogue (SYN), but group G12 was an exception. In contrast, no clear patterns emerged for the social interaction measures, turn-taking (TT) and equity (EQU), in comparison to group performance and effort. Given the small number of groups in this study, we will conduct studies with a larger number of groups in the future to understand interaction dynamics and its consequences in group problem-solving.

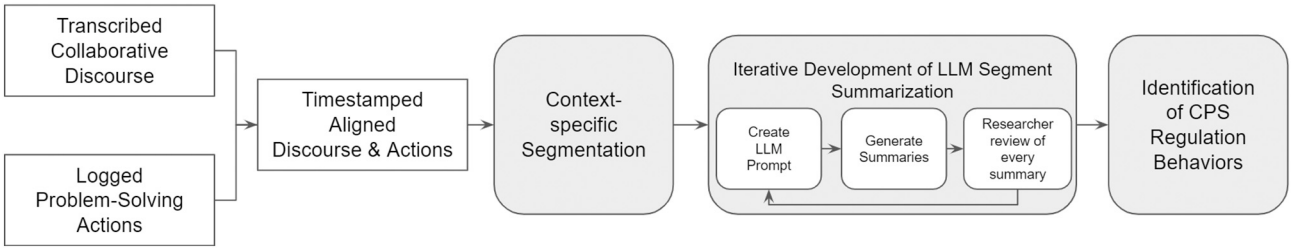


Fig. 3. Analysis Pipeline.

5.2. *RQ2: How do students' collaborative problem-solving behaviors, such as planning, enacting, monitoring, and reflecting, relate to their interaction process measures and ability to construct computational models in kinematics?*

To explore students′ problem-solving behaviors and address *RQ2*, we performed a detailed analysis of groups′ planning, enacting, and reflecting behaviors at the segment level. First, we present an overview of the distribution of these CPS behaviors across all 276 segments, along with the average interaction measures (i.e., turn-taking, equity, and synergistic discourse) for each type of behavior (Section efsec:cps-inter-rq2). Next, we analyze the relationship between CPS behaviors, their segment-level success, and these interaction measures (Section efsec: cps-inter-perf-rq2).

5.2.1. *CPS behaviors and interaction process measures*

Table 5 shows the average interaction measures (SYN, TT, EQU), average time span, and number of segments (n) for each of the CPS behaviors described in Section (3.4). Note that there are no SYN, TT, or EQU measures for segments where students were enacting without discussion, having off-topic conversations, or receiving assistance from the researchers. Segments in which students exhibited enacting behaviors were the most frequent, with 57 occurrences, while enacting and monitoring behaviors appeared in 34 segments. Planning behaviors often occurred alongside enacting in 51 segments. In contrast, 17 segments exhibited planning alone, indicating that students more frequently discussed their plans during enactments rather than keeping the two separate. Reflection behaviors were present in 20 segments, while segments that combined planning and reflecting were uncommon, with only 3 segments. The considerable variability in the average length of these segments, as shown by high standard deviation values in Table 5, makes it difficult to draw comparative conclusions regarding the time spent on each segment type.

The synergistic dialogue during planning segments was lower (SYN score = 0.72) than during enacting (SYN score = 0.88) and monitoring (SYN score = 0.88). This suggests that students may have focused more on one domain at a time during planning, while their dialogue became more synergistic when constructing their models. This suggests that model-building tasks and monitoring progress during model-building required discussions centered around both domains. For segments that included planning and enacting, the average SYN score was 0.82 as students had some segments of discourse that focused on one domain when planning and then transitioned to more synergistic dialogue when enacting. When considering reflection behaviors, students′ dialogue had the lowest average SYN score (0.56), but the standard deviation was high (0.5). When analyzing these segments individually, a number of segments had student dialogue that was highly synergistic with SYN scores above 0.90, implying the discussions included both domains. In contrast, there was also a number of reflection segments focused on one domain with SYN scores of 0.

The collaboration measures highlight that students displayed more collaboration during segments that had a combination of behaviors, in particular during planning and enacting (EQU = 0.68 and TT = 0.54) and enacting and monitoring (with EQU = 0.68 and TT = 0.55). When only enacting, students were less collaborative in their dialogue (EQU = 0.53 and TT = 0.40). We hypothesize that segments where students exhibited multiple CPS behaviors allowed for more opportunities for contribution from both students. For example, while enacting, one student might take the lead (in both discourse and action), but while planning and enacting, both students had opportunities to be involved. During reflection segments, students′ discourse had a higher EQU score (0.55) but a lower TT score (0.35), suggesting that both students had ideas that they verbalized, but there was a lack of back and forth within the group. Planning segments had low average EQU (0.48) and TT (0.46) scores, which match with prior research (Snyder et al., 2024) that showed students often had to do continuous planning during model building because they could not achieve consensus during the initial planning phase. In other words, the group would have to go back to planning after trying to enact one group member′s ideas because they had not achieved consensus in their approach earlier.

5.2.2. *CPS behaviors, interaction process measures and segment-level success*

To further explore groups′ collaborative behaviors, we analyzed their interactions concerning segment-level success as defined by the knowledge application measures in Section 3.4.1. Specifically, we identified segments where students succeeded in their model-building tasks (i.e., made some or consistent progress) and those where they did not succeed (i.e., made no progress or introduced additional errors). We focused on segments where students exhibited model-building and debugging behaviors, rather than planning and reflection behaviors, where students discussed their models without making modifications. Table 6 presents the number of segments, average interaction measures, and average time for each successful and unsuccessful problem-solving segment. Students exhibited more unsuccessful problem-solving segments than successful ones, emphasizing the complexity of the computational modeling task and the frequent challenges students face in building and debugging their models.

When considering segments where students exhibited only enacting behaviors, their discourse was more synergistic and collaborative when they were successful in their model building. However, the differences were not significant (SYN = 0.90 vs. 0.87, TT = 0.41 vs. 0.39, and EQU = 0.56 vs. 0.52). The average time spent per segment was similar, but there was a wider variability in the enacting segments that were unsuccessful. Students sometimes spent more time performing unsuccessful actions or quickly switched to working on a new part of their model when they had difficulties. During successful enacting and monitoring segments, the student exhibited more synergistic dialogue (SYN = 0.95) in comparison to the unsuccessful segments (SYN = 0.86). This points to an avenue for future research to explore if the synergistic nature of their monitoring discourse is the main reason for success.

Interestingly, the turn-taking scores were identical in both successful and unsuccessful enacting and monitoring segments (TT = 0.55), but the students′ equity scores were higher on average during unsuccessful segments (EQU = 0.70 vs. EQU = 0.62). These results imply that synergistic dialogue, where students discuss concepts across both domains, signifies successful monitoring; however, the connection to equity and turn-taking is less clear and requires further examination in future

Table 2
Coding scheme for LLM-generated summaries.

Code	Description	Example: Portion of LLM summary
PLANNING	Students decompose the problem and/or discuss steps to be taken to construct the simulation	The students discuss how to create a conditional statement for the truck′s motion. They consider using an "if-else" statement to check if the velocity equals 15 m/s, and if it does, the truck should cruise. If not, they discuss the possibility of using nested "if" statements to calculate the distance or time remaining until the stop sign and decelerate accordingly.
ENACTING	Students discuss the actions they are taking to construct the model	One student asks what value to use for delta t, and another student suggests using 0.1. The first student thanks the second student for the suggestion and confirms the value of 0.1
REFLECTING	Students review the simulation behavior and/or discuss the implications of their constructed code	The students discuss an issue they had in the previous sub-task with the truck′s motion model. They mention a block that did not work and discuss how they used a [delta t] variable with a value of 0.1, which they believe caused the truck to move slower and smoother.
PLANNING & ENACTING	Students plan and enact (construct) their solution in the same segment	They try to figure out how to make the truck slow down, and one student suggests that they need to add a new part to the model to make it slow down. They use trial-and-error approaches to refine their model and are open to suggestions from each other.
PLANNING & REFLECTING	Students plan solution steps while reflecting on past work in the same segment	S7 suggests they should check if the truck starts slowing down when it reaches the speed limit now that they have finished modeling the speeding up a portion of the task, but S21 disagrees and suggests looking at the graph. S7 then confirms that the truck does maintain constant velocity and suggests including velocity in the conditional statement. S21 agrees and apologizes for the confusion. S7 begins to suggest how the conditional statement should be structured.
ENACTING & MONITORING	Students are building parts of the model and checking it in the same segment	One student introduces blocks to increase the truck velocity and another student observes that it is not increasing. The first student asks why and the second student responds that it is just not working and taking a long time. The second student then suggests that maybe the "if" statement may not be working.
ENACTING WITH NO DISCUSSION	Students are performing actions to construct their model but there is no discussion about the approach.	There is no conversation to summarize.
ASSISTANCE	Students are receiving help from one of the researchers	They are trying to figure out how to make the truck accelerate and then cruise once it reaches the speed limit of 15 m/s. The researcher suggests using an if statement and checking the simulation data to see if the model is working as they expect.
OFF-TOPIC	Students are having an off-topic discussion unrelated to the task	The conversation is not related to the task of creating a computational model of truck motion.

Table 3
Correlations between Interactivity metrics, Performance, and Effort - Segment Level numbers in parentheses are the *p*-values.

	SYN	TT	EQU
performance (PERF) (p-val)	0.42 (0.004)	0.05 (0.75)	0.08 (0.59)
Total actions (ACT) (p-val)	0.21 (0.16)	0.31 (0.04)	0.27 (0.07)

Table 4
Overall Summary Statistics for the nine groups. The performance measure (PERF), the interaction measures, synergistic dialogue (SYN), Turn-Taking (TT), and Equity (EQU), and behavior measures, the ratio of Construction (CONSTR) to debugging (DEBUG) actions is normalized to [0,1] scores. The total number of actions (ACT) and the number of segments (Num Segs) in students′ computational model-building work also appear in the Table.

group	PERF	ACT	Num Segs	SYN	TT	EQU	CONSTR/DEBUG
g3	0.97	228	24	0.81	0.59	0.89	0.30
g2	0.95	374	45	0.72	0.39	0.92	0.74
g11	0.92	145	10	0.79	0.52	0.82	0.84
g9	0.89	165	22	0.82	0.42	0.53	0.82
g12	0.84	262	21	0.58	0.24	0.54	0.82
g8	0.76	172	23	0.62	0.48	0.91	0.70
g4	0.71	293	47	0.50	0.32	0.86	0.34
g7	0.68	309	50	0.63	0.30	0.84	0.50
g5	0.42	327	34	0.69	0.42	0.87	0.30

Table 5
CPS Behaviors, Counts (n), Average Interaction Measures (SYN, TT, EQU), and Average Segment Length (Time, minutes:seconds).

Behaviors	Count	SYN	TT	EQU	Time
	n		Average(SD)		
Planning	17	0.72 (0.4)	0.46 (0.3)	0.48 (0.3)	1:23 (1:32)
Reflecting	20	0.56 (0.5)	0.35 (0.3)	0.55 (0.4)	1:09 (1:25)
Planning and Reflecting	3	0.94 (0.04)	0.46 (0.4)	0.46 (0.4)	0:58 (0:13)
Enacting	57	0.88 (0.3)	0.40 (0.3)	0.53 (0.4)	1:37 (3:15)
Planning and Enacting	51	0.82 (0.2)	0.54 (0.2)	0.68 (0.3)	2:33 (3:24)
Enacting and Monitoring	34	0.88 (0.2)	0.55 (0.2)	0.68 (0.3)	2:22 (2:27)
Enacting with No Disc.	49	NA	NA	NA	0:52 (2:43)
Off-Topic	15	NA	NA	NA	1:54 (2:19)
Assistance	30	NA	NA	NA	2:35 (2:50)

Table 6
Successful (SUC) and Unsuccessful (UNSUC) Problem-Solving Segments and Average Interaction Measures (SYN, TT, EQU), and Average Segment Length (Time, minutes:seconds).

Behaviors	Success	Count	SYN	TT	EQU	Time
		n	Average(SD)			
Enacting	SUC	17	0.90 (0.2)	0.41 (0.3)	0.56 (0.4)	1:23 (1:42)
	UNSUC	40	0.87 (0.3)	0.39 (0.3)	0.52 (0.4)	1:43 (3:42)
Planning and Enacting	SUC	14	0.78 (0.3)	0.52 (0.2)	0.66 (0.3)	3:44 (5:10)
	UNSUC	37	0.87 (0.2)	0.56 (0.2)	0.69 (0.3)	2:06 (2:24)
Enacting and Monitoring	SUC	6	0.95 (0)	0.55 (0.1)	0.62 (0.3)	1:25 (1:00)
	UNSUC	28	0.86 (0.2)	0.55 (0.2)	0.70 (0.3)	2:34 (2:38)
Enacting and No Discussion	SUC	11	NA	NA	NA	1:07 (2:38)
	UNSUC	38	NA	NA	NA	0:48 (2:46)

research. This contrasts with the planning and enacting segments, where the average synergistic score was lower in the successful segments (0.78) than the unsuccessful segments (0.87). Additionally, the average time spent on successful versus unsuccessful segments was longer (3:44 vs. 2:06), suggesting that during computational model building, concentrating on planning and enacting in one domain before addressing the other may be beneficial, despite the increased time commitment. Furthermore, during segments where students were enacting without discussion, there was significant variability in the time spent; however, the ratio of successful to unsuccessful segments remained similar to other behaviors.

6. Conclusions and future work

The complexities involved in analyzing students′ collaborative problem-solving performance and behaviors in computational modeling for science present a unique opportunity to explore individual and group learning, as well as how interactivity and synergistic dialogue influence the understanding of science and computing concepts during model-building tasks. In this paper, we investigated differences in collaborative problem-solving behaviors and their impact on groups′ ability to complete computational modeling tasks in kinematics. We assessed how interactivity metrics, such as equity and turn-taking, and the collaborative discourse′s synergistic nature affect groups′ overall capability to construct accurate computational models.

We analyzed the differences in the groups′ planning, enacting, monitoring, and reflecting behaviors. By assessing the interactivity and synergistic metrics, we were able to define these collaborative behaviors in relation to their model-building performance. All these analyses were bolstered by multimodal approaches, where we integrated students′ conversations and activity logs to segment their model-building activities and analyze their collaborative behaviors for each problem-solving segment.

Our analysis of collaborative interactivity and domain-specific processes during problem-solving yielded several initial findings with potential for systematic, in-depth future research. First, our results for RQ1 highlighted the importance of maintaining synergistic dialogue, extending past findings (e.g., Snyder et al., 2019, Grover et al., 2019) to demonstrate the necessity of consistently leveraging both domains across all model-building task components. Second, while previous research has emphasized the significance of equity and turn-taking in collaborative problem-solving, we found that fluctuations in equity during segmented model-building are not critical indicators of student collaboration and task completion efficiency. This underscores the need for future research to better understand collaborative interactions, particularly in designing formative feedback approaches to enhance collaboration.

Beyond understanding the components of collaborative interactivity, our analysis for RQ2 centered on planning, enacting, monitoring, and reflecting behaviors during problem-solving. Our findings showed that synergistic dialogue varied by behavior type. Successful model-building segments demonstrated more synergistic discussions during the enacting and monitoring phases, highlighting the importance of integrated dialogue while tracking progress. This contrasted with effective planning segments, where discussions were less synergistic as students focused on one domain at a time before integration. Collaboration measures revealed more balanced participation in segments that combined behaviors, such as planning and enacting or enacting and monitoring. In contrast, single behavior segments, like enacting alone, often saw one student taking the lead. Reflection segments exhibited higher equity but limited turn-taking dialogue, suggesting that students shared ideas without much back-and-forth discussion. Future work aimed at adaptive support for problem-solving will build on these findings and additional research to promote monitoring and reflection behaviors (e.g., Carpenter et al., 2021) as students engage in their problem-solving tasks.

We acknowledge several limitations in this study. The small sample size, resulting from challenges in collecting and analyzing extensive multimodal data in classroom settings, limits the generalizability of our findings. Consequently, our exploratory analysis highlights an in-depth examination of students′ problem-solving behaviors. Despite these constraints, our study makes valuable contributions by utilizing a widely used SRL framework to enhance our understanding of students′ collaborative problem-solving (CPS) during STEM+C PBL. It demonstrates the effectiveness of various analytical approaches for comprehending CPS and employs multimodal classroom data to explore the interplay among group dynamics, regulatory behaviors, and performance outcomes. Future research will expand this analysis to include more participants and diverse tasks to examine its applicability. While this study focuses on regulatory CPS processes, the relationship of these processes with traditional CPS practices warrants further investigation. The categorization of groups by performance could be enhanced with larger participant pools to include additional collaborative learning variables for a more thorough analysis. Our ongoing research investigates how individual factors, such as speaking time and domain knowledge, influence group regulatory processes. To ensure coding integrity in discourse analysis, we employed systematic coding and review of the large language models′ generation processes. However, it is important to recognize the inherent limitations of LLMs, including biases and the issue of non-persistent memory. While we aimed to verify the authenticity of LLM-generated summaries through human oversight and manual coding, further validation is required for automating the coding processes (e.g., Suraworachet et al., 2024). Future work will concentrate on improving the use of LLMs for the automated coding of CPS behaviors, particularly in STEM+C educational environments.

Funding acknowledgment

This work is supported by the National Science Foundation under award DRL-2112635. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

CRediT authorship contribution statement

Caitlin Snyder: Writing − review & editing, Writing − original draft, Visualization, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Clayton Cohn:** Writing − review & editing, Writing − original draft, Visualization, Validation, Methodology, Formal analysis, Conceptualization. **Joyce Horn Fonteles:** Writing − review & editing, Writing − original draft, Visualization, Validation, Methodology, Formal analysis, Conceptualization. **Gautam Biswas:** Writing − review &

editing, Writing − original draft, Visualization, Validation, Supervision, Project administration, Methodology, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Appendix A

Fig. 4 shows the final LLM prompt (V4) we used to generate the segment summaries that is described in more detail in (Snyder et al., 2024). The highlighted green component is the *context manager* pattern (Snyder et al., 2024; Cohn, Hutchins, et al., 2024; White et al., 2023), where we detail the task and the learning environment. Next, in blue, is the *persona* pattern (Snyder et al., 2024; Cohn, Hutchins, et al., 2024; White et al., 2023), where we informed the LLM of its task. While reviewing the summaries during the human-in-the-loop approach, we identified instances of hallucination[2]. The section in red was included in the prompt to address these issues. In the yellow section, we included details about what segment type the students were working on (i.e., initialization, conditional statements, updating variables under conditions or generally). Finally, our prompt concludes with the *meta language creation* pattern (purple) with the input template (i.e., SPEAKER: DISCOURSE) and asking the LLM to summarize the segment.

Students work to create computational models that simulate motion guided by kinematics. In this model, a truck, starting from rest, must accelerate to the speed limit. The truck must then cruise at the speed limit as long as it can before having to stop at a stop sign. The truck's maximum acceleration is 4 m/s, and its maximum deceleration is -4 m/s. The speed limit is 15 m/s. The truck's initial position is -60. The stop sign's position is 38.16. The students have access to the following variables: acceleration, delta t, velocity, position, stop_sign_position. You are trying to help a teacher understand the students' conversations in relation to the model segments they are constructing to create the full computational model of truck motion. You are not allowed to infer things that are not explicitly stated in the discourse segments when providing your summary but if you feel you cannot provide a sufficient summary based on the text provided, take your best guess anyway. In this segment, the students are working on initializing the variables needed to model the truck's motion. In the next section is the students' discourse. Each line in the discourse is an individual student utterance with the format SPEAKER : DISCOURSE. Based on these instructions and the context, summarize the following conversation.

Fig. 4. Final LLM prompt used to generate segment summaries reprinted from Snyder et al. (2024).

The following subsections have each of the prompts we iterated through. The segment summaries generated by each prompt were reviewed by two members of the research team. After every segment summary was judged to accurately summarize the segment dialogue, we stopped refining the prompt (leaving us with the final prompt in Fig. 4). However, while this prompt was sufficient for the analysis presented in this paper, additional prompt iterations may be necessary if these summaries were used for a different analysis with a different focus. Fig. 5 shows the notes taken about each prompt after reviewing the generated summaries and comparing them against the student discourse.

[2] https://machinelearningmastery.com/a-gentle-introduction-to-hallucinations-in-large-language-models/

Prompt Iteration	Components	Limitations
0	• Abbreviated description of task • Discourse	• Confusion about task components (e.g., stop sign) • Incorrect understanding of computational components (e.g., if-else statements) • Inferring ideas/conceptual connections that students don't verbalize
1	• Named environment description • Domain description • Task description • Researcher role description • Discourse	• Incorrect understanding of the named environment (e.g., thinking C2STEM stands for Coding and Creativity through STEM) • Incorrectly identifying segments of discourse as not relevant to the task
2	• Abbreviated unnamed environment description • Abbreviated domain description • Task description • Abbreviated teacher role description • Discourse	• Difficulty parsing discourse where students are vague with verbalizing names of components, etc. (e.g., "move this there" or "try that")
3	• Abbreviated unnamed environment description • Abbreviated domain description • Task description • Abbreviated teacher role description • Context of actions and progress made during segment • Discourse	• Incorrect understanding of some components of the solution (e.g., setting velocity to speed limit under the conditional) • Additional contextual information causes token limit to be reached
4	• Task description • Abbreviated teacher helper role description • Explicit instructions for no inference • Abbreviated context during segment • Discourse	• Specific to one C2STEM task

Fig. 5. Prompt Iteration Notes.

A.1. Prompt V0

You are a helpful assistant who summarizes science texts. Importantly, you are not allowed to infer things that are not explicitly mentioned. Additionally, if you feel you cannot provide a sufficient summary based on the text provided, take your best guess anyway. You must always attempt to provide a summary.

A.2. Prompt V1

C2STEM is a computer-based learning environment based on a novel computational paradigm that combines visual programming with physics modeling using Netsblox (a visual, block-based programming environment) to promote computational modeling in the physics domain while promoting synergistic learning of physics and computational thinking (CT) concepts and practices. C2STEM′s website can be found at this URL: https://c2stem.org/.

In C2STEM, CT concepts include those related to programming, such as conditional statements ("if blocks", for example), initializing variables, updating variables, and operator expressions. Physics concepts include those related to the kinematic equations that define Newtonian physics, such as velocity, acceleration, time (delta t), position, and distance. In C2STEM, high school sophomores work collaboratively to apply and combine their knowledge of physics and CT to accomplish various tasks in the C2STEM environment by creating computational models with C2STEM to complete several scientific simulations. In each simulation, students must first initialize all variables, define the behavior of the model for each time step, and test the computational model by running the simulation. In C2STEM, students have various tools at their disposal, such as graphs and tables, to help

them debug and improve their computational models. In this particular simulation, students must work together in pairs (dyads) and use C2STEM to build a computational model inside C2STEM that causes a truck starting from rest (initial velocity = 0 m/s), to accelerate to a speed of 15 m/s (the speed limit) and then maintain that speed. The maximum acceleration the truck can have is 4 m/s/s. At a certain distance away from the stop sign, the students must make the truck start to slow down so that its velocity is 0 at the stop sign. The truck can have a maximum negative acceleration of −4 m/s/2.

You are a scientific researcher whose job is to summarize transcribed segments of student discourse while they try to accomplish this truck task in C2STEM. You are doing this task to help a Research Team that is interested in understanding how students construct knowledge and want to understand whether or not students have a correct understanding of the relationships between the various physics and CT concepts as students work to build and test their computational models in the simulation. The Research Team is also interested in identifying times when certain scientific and CT concepts are missing from the discourse that are integral in fully understanding the relationships between concepts and how they fit into the kinematic equations. Because of this, you are not allowed to infer things that are not explicitly stated in the discourse segments when providing your summary. Additionally, if the discourse is relevant, but you feel you cannot provide a sufficient summary based on the text provided, take your best guess anyway. You must always attempt to provide a summary if you feel the discourse is relevant to the task at hand. If the discourse segment is not related to the C2sTEM truck task or is otherwise off-topic, simply respond 'not relevant.'

Each line in the discourse is an individual student utterance with the format SPEAKER: DISCOURSE. Based on these instructions, summarize the following conversation:

A.3. Prompt V2

Students are working in pairs on a computer-based learning environment that combines block-based programming with scientific modeling through a domain-specific modeling language to promote simultaneous learning of physics and computational thinking (CT) concepts and practices. CT concepts include conditional statements ("if blocks", for example), initializing variables, updating variables, and operator expressions. Physics concepts include those related to the kinematic equations that define Newtonian physics, such as velocity, acceleration, time (delta t), position, and distance. Students work to create computational models that simulate Newtonian physics. In each model, students must initialize all variables, define the behavior of the model for each time step, and test the computational model by running the simulation. Students have various tools at their disposal, such as graphs and tables, to help them debug and improve their computational models.

In this task, students are creating a computational model that simulates a truck starting from rest (initial velocity = 0 m/s), to accelerate to a speed of 15 m/s (the speed limit) and then maintain that speed. The maximum acceleration the truck can have is 4 m/s/s. At a certain distance away from the stop sign, the students must make the truck start to slow down so that its velocity is 0 at the stop sign. The truck can have a maximum negative acceleration of −4 m/s/2.

You are a teacher whose job is to summarize transcribed segments of student discourse. You are doing this task to help a Research Team that is interested in understanding how students construct physics and CT knowledge and want to understand whether or not students have a correct understanding of the relationships between the various physics and CT concepts. You are not allowed to infer things that are not explicitly stated in the discourse segments when providing your summary but if you feel you cannot provide a sufficient summary based on the text provided, take your best guess anyway.

Each line in the discourse is an individual student utterance with the format SPEAKER: DISCOURSE. Based on these instructions, summarize the following conversation and create a set of facts that the answer depends on that should be fact-checked and list this set of facts at the end of your output that generated the summarization.

A.4. Prompt V3

Students are working in pairs on a computer-based learning environment that combines block-based programming with scientific modeling through a domain-specific modeling language to promote simultaneous learning of physics and computational thinking (CT) concepts and practices. CT concepts include conditional statements ("if blocks", for example), initializing variables, updating variables, and operator expressions. Physics concepts include those related to the kinematic equations that define physics behavior, such as velocity, acceleration, time (delta t), position, and distance. Students work to create computational models that simulate motion guided by kinematics. In each model, students must initialize all variables, define the behavior of the model for each time step, and test the computational model by running the simulation. Students have various tools at their disposal, such as graphs and tables, to help them debug and improve their computational models.

In this task, students are creating a computational model that simulates a truck starting from rest (initial velocity = 0 m/s), to accelerate to a speed of 15 m/s (the speed limit) and then maintain that speed. The maximum acceleration the truck can have is 4 m/s/s. At a certain distance away from the stop sign, the students must make the truck start to slow down so that its velocity is 0 at the stop sign. The truck can have a maximum negative acceleration of −4 m/s/2.

You are a teacher whose job is to summarize transcribed segments based on these learning objectives of student discourse in order to design actionable feedback to the students. You are doing this task to help a Research Team that is interested in understanding how students construct physics and CT knowledge and want to understand whether or not students have a correct understanding of the relationships between the various physics and CT concepts. You are not allowed to infer things that are not explicitly stated in the discourse segments when providing your summary but if you feel you cannot provide a sufficient summary based on the text provided, take your best guess anyway. Here is context for what the students in this segment are doing in the environment:

In this segment the students added these components to the model [addedblocks], edited or adjusted these components on the model [adjustedblocks], and added, edited or adjusted these components that are not connected to the model [draftedblocks]. They ran their model [or used the data tools]. In this segment the students worked on [context == initialization: "initializing variables"; context == updating-variables-under-conditions: "updating variables under specific conditions"; context == updating-variables-every-sim-step: "updating variables every step in the simulation"; context == conditional-clause: "conditional statements"] and at the end of the segment the students' model [progress == backward: "had a new error"; progress == no progress: "still had a previously existing error but was added to without any new errors being created"; progress == progress: "had a previously existing error fixed"; progress == consistent: "had no errors and was added to".]

In the next section is the students' discourse. Each line in the discourse is an individual student utterance with the format SPEAKER: DISCOURSE.

Based on these instructions and the context, summarize the following conversation.

References

Asghar, A., Ellington, R., Rice, E., Johnson, F., & Prime, G. M. (2012). Supporting stem education in secondary science contexts. *Interdisciplinary Journal of Problem-Based Learning, 6*, 4.

Azevedo, R., Moos, D. C., Johnson, A. M., & Chauncey, A. D. (2010). Measuring cognitive and metacognitive regulatory processes during hypermedia learning: Issues and challenges. *Educational Psychologist, 45*, 210−223.

Barrows, H. S., Tamblyn, R. M., et al. (1980). *Problem-based learning: An approach to medical education. 1*. Springer Publishing Company.

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*, 13.

Beers, P. J., Boshuizen, H. P. E., Kirschner, P. A., & Gijselaers, W. H. (2005). Computer support for knowledge construction in collaborative learning environments. *Computers in Human Behavior, 21*, 623−643.

Blikstein, P., & Worsley, M. (2016). Multimodal learning analytics and education data mining: Using computational technologies to measure complex learning tasks. *Journal of Learning Analytics, 3*, 220−238.

Carpenter, D., Cloude, E., Rowe, J., Azevedo, R., & Lester, J. (2021). Investigating student reflection during game-based learning in middle grades science. In *LAK21: 11th international learning analytics and knowledge conference* (pp. 280−291).

Chi, M. T. H. (2008). Three types of conceptual change: Belief revision, mental model transformation, and categorical shift. In S. Vosniadou (Ed.), *Handbook of research on conceptual change* (pp. 61−82). Hillsdale, NJ, USA: Erlbaum.

Cohn, C., Hutchins, N., Le, T., & Biswas, G. (2024). A chain-of-thought prompting approach with llms for evaluating students′ formative assessment responses in science. In *, 38. Proceedings of the AAAI conference on artificial intelligence* (pp. 23182−23190). https://ojs.aaai.org/index.php/AAAI/article/view/30364. https://doi.org/10.1609/aaai.v38i21.30364.

Cohn, C., Snyder, C., Montenegro, J., & Biswas, G. (2024). Towards a human-in-the-loop LLM approach to collaborative discourse analysis. In *International Conference on Artificial Intelligence in Education (pp. 11-19)*. Cham: Springer Nature Switzerland.

Dillenbourg, P. (1999). Collaborative learning: Cognitive and computational approaches. In *advances in learning and instruction series*. ERIC.

diSessa, A. (2001). *Changing minds: Computers, learning, and literacy*. Cambridge MA USA: MIT Press.

Eichmann, B., Goldhammer, F., Greiff, S., Pucite, L., & Naumann, J. (2019). The role of planning in complex problem solving. *Computers & Education, 128*, 1−12.

Garrison, D. R., & Akyol, Z. (2013). Th e community of inquiry th eoretical framework. In *Handbook of distance education* (pp. 122−138). Routledge.

Grau, V., & Whitebread, D. (2012). Self and social regulation of learning during collaborative activities in the classroom: The interplay of individual and group cognition. *Learning and Instruction, 22*, 401−412.

Grosch, J., & Emmelmann, H. (1990). A tool box for compiler construction. *International Workshop on Compiler Construction, Springer.*, 106−116.

Grover, S., & Pea, R. (2013). Computational thinking in k−12: A review of the state of the field. *Educational Researcher, 42*, 38−43. https://doi.org/10.3102/0013189X12463051. URL: doi:10.3102/0013189X12463051, doi:10.3102/0013189X12463051, arXiv:doi:10.3102/0013189X12463051.

Grover, S., Basu, S., & Schank, P. (2018). What we can learn about student learning from open-ended programming projects in middle school computer science. In *Proceedings of the 49th ACM technical symposium on computer science education* (pp. 999−1004). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3159450.3159522.

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. In *The American Educational Research Association Annual Meeting*.

Hadwin, A. F., Ja¨rvel¨a, S., & Miller, M. (2011). Self-regulated, co-regulated, and socially shared regulation of learning. Handbook of self-regulation of learning and performance. *30*, 65−84.

Hambrusch S. Hoffmann C. Korb J.T. Haugan M. Hosking A.L. 2009 A multidisciplinary approach towards computational thinking for science majors SIGCSE Bull 41 183 187 doi:https://doi.org/10.1145/1539024.1508931 10.1145/1539024.1508931.

Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? *Educational Psychology Review, 16*, 235−266.

Hutchins, N., Biswas, G., Maro´ti, M., L´edeczi, A´., Grover, S., Wolf, R., Blair, K. P., Chin, D., Conlin, L., Basu, S., & McElhaney, K. (2020). C2stem: a system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology, 29*, 83−100.

Hutchins, N. M., Snyder, Caitlin, E.M., Grover, S., & Biswas, G. (2021). Analyzing debugging processes during collaborative, computational modeling in science. In *The International Society of the Learning Sciences Annual Meeting 2021*. International Society of the Learning Sciences (ISLS).

J¨arvela¨, S., Ga˘sevi´c, D., Sepp¨anen, T., Pechenizkiy, M., & Kirschner, P. A. (2020). Bridging learning sciences, machine learning and affective computing for understanding cognition and affect in collaborative learning. *British Journal of Educational Technology, 51*, 2391−2406.

J¨arvel¨a, S., Malmberg, J., Haataja, E., Sobocinski, M., & Kirschner, P. A. (2021). What multimodal data can tell us about the students′ regulation of their learning process? *Learning and Instruction, 72*, Article 101203.

Jeong, H., & Chi, M. T. (2007). Knowledge convergence and collaborative learning. *Instructional Science, 35*, 287−315.

Jeong, H., Hmelo-Silver, C. E., & Jo, K. (2019). Ten years of computer-supported collaborative learning: A meta-analysis of cscl in stem education during 2005−2014. *Educational Research Review, 28*, Article 100284.

Kalina, C., & Powell, K. (2009). Cognitive and social constructivism: Developing tools for an effective classroom. *Education, 130*, 241−250.

Knight, S., Wise, A. F., & Chen, B. (2017). Time for change: Why learning analytics needs temporal analysis. *Journal of Learning Analytics, 4*, 7−17.

Larkin, S. (2006). Collaborative group work and individual development of metacognition in the early years. *Research in Science Education, 36*, 7−27.

Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2023). Prompt engineering in large language models. *International conference on data intelligence and cognitive informatics, Springer.*, 387−402.

Meier, A., Spada, H., & Rummel, N. (2007). A rating scheme for assessing the quality of computer-supported collaboration processes. *International Journal of Computer-Supported Collaborative Learning, 2*, 63−86.

Mislevy, R.J., Haertel, G.D., 2006. Implications of evidence-centered design for educational testing. Educational Measurement: Issues and Practice 25, 6−20. URL: doi:https://doi.org/10.1111/j.1745-3992.2006.00075.x, arXiv:doi:https://doi.org/10.1111/j.1745-3992.2006.00075.x.

Nasir, J., Kothiyal, A., Bruno, B., & Dillenbourg, P. (2021). Many are the ways to learn identifying multi-modal behavioral profiles of collaborative learning in constructivist activities. *International Journal of Computer-Supported Collaborative Learning, 16*, 485−523.

NGSS. (2013). *Next generation science standards: For states, by states*. The National Academies Press.

NRC. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.

OECD. (2015). PISA 2015 collaborative problem solving framework. www.oecd.org/pisa/pisaproducts/Draft%20PISA%202015%20.

Rakovi´c, M., Bernacki, M. L., Greene, J. A., Plumley, R. D., Hogan, K. A., Gates, K. M., & Panter, A. T. (2022). Examining the critical role of evaluation and adaptation in self-regulated learning. *Contemporary Educational Psychology, 68*, Article 102027.

Roschelle, J., & Teasley, S. D. (1995). The construction of shared knowledge in collaborative problem solving. In *Computer supported collaborative learning* (pp. 69−97). Springer.

Rummel, N., Spada, H., & Hauser, S. (2009). Learning to collaborate while being scripted or by observing a model. *International Journal of Computer-Supported Collaborative Learning, 4*, 69−92.

Schmidt, D. C., Spencer-Smith, J., Fu, Q., & White, J. (2024). Towards a catalog of prompt patterns to enhance the discipline of prompt engineering. *ACM SIGAda Ada Letters, 43*, 43−51.

Schon, D. A., & DeSanctis, V. (1986). *The reflective practitioner: How professionals think in action*.

Schunk, D. H., & Zimmerman, B. J. (1998). *Self-regulated learning: From teaching to self-reflective practice*. Guilford Press.

Schwartz, D., Chase, C., Chin, D., Oppezzo, M., Kwong, H., Okita, S., Biswas, G., Roscoe, R., Jeong, H., & Wagster, J. (2009). Interactive metacognition: Monitoring and regulating a teachable agent. In *Handbook of metacognition in education* (pp. 340−359).

Sears, D. A., & Reagin, J. M. (2013). Individual versus collaborative problem solving: Divergent outcomes depending on task complexity. *Instructional Science, 41*, 1153−1172.

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*, 351−380.

Sherin B. diSessa A.A. Hammer D. 1993 Dynaturtle revisited: Learning physics through collaborative design of a computer model Interactive Learning Environments 3 91 118 arXiv:doi:https://doi.org/10.1080/1049482930030201.

Snyder, C., Hutchins, N. M., Cohn, C., Fonteles, J. H., & Biswas, G. (2024). Analyzing students collaborative problem-solving behaviors in synergistic stem+c learning. In *Proceedings of the 14th learning analytics and knowledge conference* (pp. 540−550). New York, NY, USA.: Association for Computing Machinery. https://doi.org/10.1145/3636555.3636912.

Soller, A. (2001). Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education, 12*, 40−62.

Stahl, G., & Hesse, F. (2009). Paradigms of shared knowledge. *International Journal of Computer-Supported Collaborative Learning, 4*, 365−369.

Suraworachet, W., Seon, J., Cukurova, M. 2024. Predicting challenge moments from students′ discourse: A comparison of GPT-4 to two traditional natural language processing approaches. arXiv preprint arXiv:2401.01692.

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 *Computer-supported collaborative learning*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*, 127−147.

White, B., Frederiksen, J., Collins, A. 2009. 10 the interplay of scientific inquiry and metacognition. Handbook of metacognition in education, 175.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv*, Article 2302.11382.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*, 33−35.

Winne, P. H. (2010). Bootstrapping learner′s self-regulated learning. *Psychological Test and Assessment Modeling, 52*, 472.

Wise, A. F., Knight, S., & Shum, S. B. (2021). Collaborative learning analytics. *International handbook of computer-supported collaborative learning*, 425−443.

Zhang, L., Kalyuga, S., Lee, C. H., Lei, C., & Jiao, J. (2015). Effectiveness of collaborative learning with complex tasks under different learning group formations: A cognitive load perspective. In *Hybrid learning: Innovation in educational practices: 8th international conference* (pp. 149−159). Wuhan, China: ICHL 2015. , July 27-29, 2015, proceedings 8, Springer.

Zimmerman, B. J., & Moylan, A. R. (2009). Self-regulation: Where metacognition and motivation intersect. In *Handbook of metacognition in education* (pp. 311−328). Routledge.