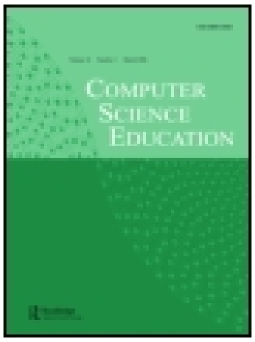

Computer Science Education

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ncse20

Two-Computer Pair Programming: Exploring a Feedback Intervention to improve Collaborative Talk in Elementary Students.

Zarifa Zakaria , Jessica Vandenberg , Jennifer Tsan , Danielle Cadieux Boulden , Collin F. Lynch , Kristy Elizabeth Boyer & Eric N. Wiebe

To cite this article: Zarifa Zakaria , Jessica Vandenberg , Jennifer Tsan , Danielle Cadieux Boulden , Collin F. Lynch , Kristy Elizabeth Boyer & Eric N. Wiebe (2021): Two-Computer Pair Programming: Exploring a Feedback Intervention to improve Collaborative Talk in Elementary Students., Computer Science Education, DOI: 10.1080/08993408.2021.1877987

To link to this article: https://doi.org/10.1080/08993408.2021.1877987

Published online: 03 Feb 2021.

COMPUTER SCIENCE EDUCATION
https://doi.org/10.1080/08993408.2021.1877987

ARTICLE

Two-Computer Pair Programming: Exploring a Feedback Intervention to improve Collaborative Talk in Elementary Students.

Zarifa Zakaria [a], Jessica Vandenberg [a], Jennifer Tsan [b], Danielle Cadieux Boulden [b], Collin F. Lynch [b], Kristy Elizabeth Boyer [c] and Eric N. Wiebe [d]

[a]Department of Teacher Education and Learning Sciences, North Carolina State University, Raleigh, NC, USA; [b]Department of Computer Science, North Carolina State University, Raleigh, NC, USA; [c]Department of Computer & Information Science & Engineering, University of Florida, Gainesville, FL, USA; [d]Department of STEM Education, North Carolina State University, Raleigh, NC, USA

ABSTRACT

Background and Context: Researchers and practitioners have begun to incorporate collaboration in programming because of its reported instructional and professional benefits. However, younger students need guidance on how to collaborate in environments that require substantial interpersonal interaction and negotiation. Previous research indicates that feedback fosters students' productive collaboration.
Objective: This study employs an intervention to explore the role instructor-directed feedback plays on elementary students' dyadic collaboration during 2-computer pair programming.
Method: We used a multi-study design, collecting video data on students' dyadic collaboration. Study 1 qualitatively explored dyadic collaboration by coding video transcripts of four dyads which guided the design of Study 2 that examined conversation of six dyads using MANOVA and non-parametric tests.
Findings: Result from Study 2 showed that students receiving feedback used productive conversation categories significantly higher than the control condition in the sample group considered. Results are discussed in terms of group differences in specific conversation categories.
Implications: Our study highlights ways to support students in pair programming contexts so that they can maximize the benefits afforded through these experiences.

ARTICLE HISTORY
Received 20 December 2019
Accepted 15 January 2021

KEYWORDS
Pair programming; collaboration; elementary school; feedback; intervention

Introduction

Researchers and practitioners have begun to incorporate collaboration in programming activities because of its reported instructional and professional benefits. Studies suggest that collaboration is an effective pedagogical approach for programming instruction for undergraduates (Hanks et al., 2011) and to prepare students for further education and workforce needs (National Research Council, 2013). It fosters higher-order thinking skills

CONTACT Zarifa Zakaria zzakari@ncsu.edu Department of Teacher Education and Learning Sciences, North Carolina State University, Raleigh, NC, USA

(Williams et al., 2002) and facilitates effective knowledge sharing through productive dialogue (Kavitha & Ahmed, 2015). Pair programming is where two individuals solve a programming challenge together. Though originating in industry, pair programming is often incorporated in college-level CS courses where the bulk of research on the pedagogical approach has occurred (Umapathy & Ritzhaupt, 2017; Williams et al., 2000). Fewer studies can be found at the high school and middle grades, and only a few studies can be found on elementary students' pair-programming practices. The research done with elementary students indicates that pair-programming with block-based programming environments can foster problem-solving skills, develop their understanding of programming concepts, and work as a motivating environment to explore other domain-related concepts (Calder, 2010; Lai & Yang, 2011). Given the general acceptance of group work in the elementary grades as a teaching and learning strategy (i.e. collaborative inquiry, project-based problem-solving etc.) (Adams & Hamm, 1998; De Lisi & Golbeck, 1999) and the growing popularity of block-based programming for this age-range (Franklin et al., 2015), it seems worthwhile to explore pedagogical practices related to pair programming with elementary-aged students as well.

With the more common, popularized form of pair programming referred to here as 1-computer pair programming (1 C), dyads share one computer and have a defined role in the collaboration: the driver controls the programming environment, and the navigator monitors the progress and anticipates future steps (Hanks et al., 2011). These defined roles in 1 C pose several challenges in students' collaboration, such as imbalances in dialogue and work distribution, in addition to equity issues for elementary students (Lewis & Shah, 2015; Shah & Lewis, 2019; Shah et al., 2014; Tsan et al., 2018). Prior work has demonstrated that elementary students can show a clear disinclination towards 1 C because of the restrictions under navigator role (Bradbury et al., 2019). As 1 C involves a high degree of interpersonal negotiation and reconciliation during the turn-taking of driver and navigator roles, this is perhaps not surprising (Zakaria et al., 2019). Studies on children's social practices show that elementary-aged students are still developing necessary socio-emotional skills such as social problem-solving and showing empathy (Beauchamp & Anderson, 2010), which are needed when sharing a computer and negotiating turn-taking (Shah et al., 2014). These findings point to a need to explore alternative pair-programming strategies that better leverage the benefits of collaborative learning strategies. Some studies have also explored an alternative setting, referred to here as 2-computer pair programming (2 C) (Figure 1), in which pairs of students collaborate side-by-side with individual computers on the same programming project (Bradbury et al., 2019; Zakaria et al., 2019). Such a setting is also referred to in the literature as distributed programming, concurrent-coupled programming or side-by-side programming (e.g., Dewan et al., 2009; Nawrocki et al., 2005; Zakaria et al., 2019). Although such an environment seems promising given the independence it provides for pairs to learn and contribute to each other's work equally, it is important to examine the quality of collaboration in such a setting. In the 2 C setting, partners do not have roles. Thus, supporting students during their collaboration needs to be structured in a way that scaffolds appropriate discourse so students can productively collaborate. Designing effective support for students in upper elementary grades (9–11 years) will require in-depth exploration of the collaborative processes of these students as they engage in 2 C. By first determining the characteristics of collaboration

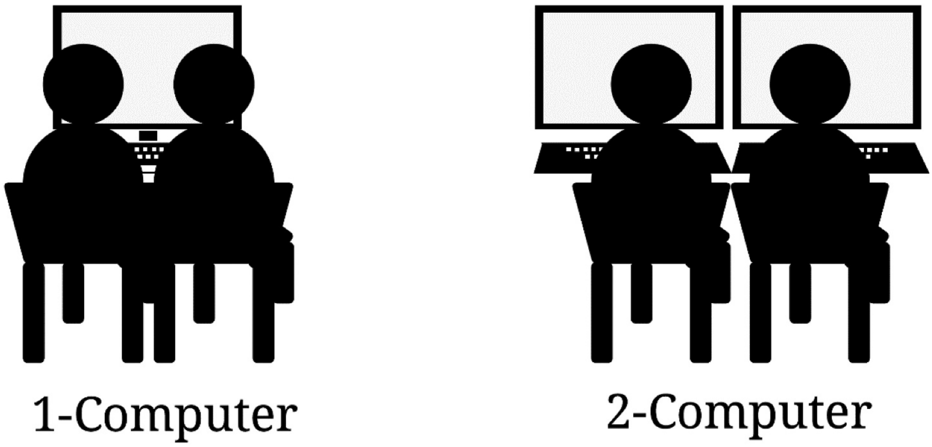

Figure 1. 1 C and 2 C Pair Programming (Zakaria et al., 2019).

during 2 C, we can then design and implement instructional interventions that expand the scope of quality collaboration in this setting.

Previous work

Collaborative talk

At the heart of the instructional impact of collaborative work is the constructive discourse between students that moves learning forward (Tudge, 1992). A framework by Mercer (2002) elaborates on students' dialogue, distinguishing more productive conversation from less productive ones. Drawing on Vygotsky's Zone of Proximal Development (ZPD) (Vygotsky, 1978) and Bruner's "Scaffolding" (Bruner, 1978), Mercer proposed the concept of Intermental Development Zone (IDZ) that focuses on the nature of interactive processes between teacher and student or peer-to-peer (Mercer, 2002). In these joint activities, three types of conversations can occur, which Mercer termed Cumulative, Disputational and Exploratory talk (Table 1). In Cumulative talk, speakers build positively but uncritically on what the other has said, whereas Disputational talk is characterized by disagreement and individualized decision making. In Exploratory talk, participants engage critically but constructively with each other's ideas which leads to improved reasoning and conceptual understanding (Bennett & Cass, 1989; Mercer, 2002). Exploratory talk has been shown to expand the joint ZPD by enabling partners to achieve a better mutual understanding of the problem (Fernández et al., 2015).

Previous studies that employed this framework to analyze students' conversations during pair programming showed that elementary and middle school students organically use Cumulative talk more than the other two types (Campe et al., 2020; Zakaria et al., 2019). Despite the knowledge that students' use of Exploratory talk is improved with teacher guidance (Mercer et al., 2004, 1999), Rojas-Drummond and Mercer (2003) found that teachers rarely instruct students in effective collaborative talk. However, when taught how to engage in Exploratory talk, students produced improved arguments, and critically considered alternative views during collaborative work (Rojas-Drummond & Zapata,

Table 1. Conversation categories in the coding scheme and the inter-rater agreement (kappa) for Study 1.

Categories	Sub-categories	Description of sub-categories	Examples	*k*
Question	*Question*	Any question, concern, request for assistance, clarification, seeking or check confirmation.	"Where can I find the 'if' block?"	.85
	Respond to question	Any response to partner's question or concerns.	"You can search under categories"	.93
Discussion	*Agreement*	Agreement on any opinion, edits	Yeah, I think so too.	.83
	Disagreement	Disagreement on any edits or opinion.	No, this should be here. No, I think (pause)	.66
	Explanation	Explain what step they are taking or what edits they are doing or random iteration of what they are doing themselves. (have edited or editing in the present)	"if touching sprite 4, then stop" (while editing on the screen).	.70
	Suggestion	Any suggestions when directly talking to the partner and suggesting before taking an action.	"maybe we should add this block."	.88
	Reminder	Reminding, redirecting or warning about conducting any steps or to be on-task.	"You have to save it before exiting"	1.0
	Antagonistic action	Actions interactions that cause tension including hurtful comments instigating fights, prodding, putting down partner contributions, and showing annoyance with partner	"you are being ridiculous".	(no occurrence)
Manipulation	*Edits with consent*	Edits after discussion, agreement or getting a positive response from the partner.		(no occurrence)
	Edits without consent	Edits without discussing or getting any positive response.		.93

2004). Mercer et al. (1999) proposed a set of ground rules for Exploratory talk to use as guidance to facilitate collaborative discourse. With teachers' support, students could integrate partner's ideas and strategies, co-construct new ideas to contribute to the collaborative activity critically and productively (Warwick et al., 2013). Although 2 C offers balanced autonomy for students–a departure from traditional driver-navigator roles–it involves a higher level of technology coordination as both partners in 2 C have control of a computer. It envisions more verbal negotiation for decision-making while students simultaneously edit in the workspace. Because 2 C presents new dynamics in the collaborative affordances of pair programming, it is important to examine students' discourse in order to best leverage the affordances of this setting and enhance Exploratory talk.

Feedback on collaboration

Mercer and others believe that productive collaboration cannot simply emerge naturally but rather has to be taught and supported through modeling and feedback (Mercer, 1995; Mercer et al., 1999). Similar to cognitive skills, collaboration skills include procedural knowledge that inform students how to perform in a collaborative environment (Deiglmayr & Spada, 2010). Feedback is found to be effective when focused on different aspects of collaboration like group awareness (Pifarré et al., 2014), knowledge sharing breakdown (Soller, 2001; Zumbach et al., 2006), or unproductive communication (Gweon et al., 2006). It is also found that along with cognition, affective dimensions such as motivation can affect collaboration (Meier et al., 2007). Feedback on "social grounding" which Zumbach et al. (2006) summarize as interactions, collaboration, and motivational processes appear to be essential for learning. In terms of motivation towards collaboration, it is found that students with a growth mindset, which emphasizes the belief on effort exertion in utilizing different strategies to be successful in a task, value group work and its creative potential more (Alpay & Ireson, 2006; Dweck, 2013), and perform significantly better on programming with feedback focused on mindset (Cutts et al., 2010).

Preliminary work on elementary pair programming

A set of preliminary studies undergirds the studies reported here. Our initial classroom observations and other literature began to raise concerns with 1 C with elementary students. These findings led to an exploration of 1 C and 2 C settings where we conducted post-hoc focus groups to gather more information on student experience and preference. Students who participated in both settings expressed they had less autonomy and opportunity to equally contribute to the activity in 1 C (Bradbury et al., 2019). Prior research has demonstrated the advantages of using one's own computer rather than sharing a single computer during collaboration. Evidence shows that elementary students completed tasks with more time-efficiency and higher levels of enjoyment and engagement (Infante, 2009; Lewis, 2011; Scott et al., 2003). Moreover, 2 C pair programmers have been found to produce higher quality industry products (Bandukda & Nasir, 2010). Similar advantages have also been found in studies conducted with undergraduates (Cockburn, 2004; Nawrocki et al., 2005).

Current work

Exploring nuances of dyadic conversation may give us an in-depth understanding of elementary students' needs and help us design appropriate interventions to support effective collaboration during pair programming. Prior literature and our own work have pointed to the potential advantages of elementary students working in a 2 C setting. We define 2 C as a setting where student pairs each have their own computer screen, keyboard, and mouse (thus lowering points of conflict) but sit side-by-side where they can leverage discursive practices to create a single solution to a programming problem. Depending on the programming environment (i.e. Netsblox, Scratch etc.) used in 2 C, students could be working in a linked environment where their programming space was synchronized by the software, or in an unlinked environment where the nature of their physical proximity allowed them to visually and verbally compare and synchronize their programs manually. We conducted two consecutive studies with the goal of both more deeply understanding elementary students' collaborative discourse in 2 C settings, but also to look at how feedback shapes this discourse. More specifically, the goal of Study 1 was to refine the coding scheme used in the preliminary studies reported above to adapt it for use to discern students' collaborative process in a 2 C setting. The prior coding scheme was designed for use in a 1 C context by Ruvalcaba et al. (2016) to analyze videos of dyadic collaboration and thus needed to be modified for a 2 C setting. Utilizing those findings, a second study explored an intervention program to support students' collaboration in a 2 C setting. We have not compared or contrasted these two studies, rather Study 1 was considered informative for designing and conducting Study 2.

Results and coding experiences in Study 1 helped us to modify and refine discourse analysis methods and design a feedback intervention to improve students' collaborative talk during pair-programming activities. We consider talking to be productive when it includes characteristics of Exploratory talk, less productive with Cumulative talk and unproductive with Disputational talk. Feedback was provided to the dyads during their collaboration to enhance their use of Exploratory talk. Study 2 utilized this revised and refined coding scheme to explore a feedback intervention informed both by Mercer's (2002) Exploratory talk as well as Dweck's (2013) growth mindset theories.

Table 2. Demographics of the students in Study 1.

Student pairs	Gender	Race
Sandy	Female	White
Anthony	Male	White
Dorothy	Female	White
David	Male	White
Melony	Female	White
Rupert	Male	White
Clara	Female	Biracial
Luke	Male	Biracial

Study 1

For Study 1, we explore the following research questions: 1) *How do dyads utilize different categories of collaborative talk?* 2) *What are the discursive characteristics of dyadic collaboration in 2-computer pair programming?*

Method

The study took place at a suburban elementary school in the Southeastern United States. The participants were 15 fifth grade (10–11 years) students from which we used video data of 8 students, forming 4 dyads (formed by the teacher); 4 girls and 4 boys (Table 2). This subset of participants was chosen in terms of audio and video clarity. All the students were in an academically gifted student program and had been participating in programming lessons designed by the authors. The lessons covered topics such as conditionals, loops, debugging, and game design throughout three days of activities; however, they were in 2 C setting only one day during which they did several debugging activities. From these activities, we used one debugging activity that required students to use all the other concepts taught before. Students were tasked with fixing the code so that a mouse (Sprite 1) stopped walking back-and-forth when a snowflake (Sprite 2), falling from the sky, touched it. In this 2 C setting, students coded in NetsBlox (Broll et al., 2017). NetsBlox provided a linked, synchronous workspace (akin to Google docs) viewed on each of the dyad's computer. On average students took 6.3 mins to solve the problem and we analyzed a total of 25.53 mins of video data (Table 4). For this study, we used Open Broadcaster Software (OBS) (Bailey, 2017) to align webcam, screen capture, and audio data (captured through headsets) into a single file. In addition, we collected the students' programs and log data.

Using a mixed-methods approach (Creswell & Creswell, 2017), first, video-recorded dialogue was qualitatively analyzed as multiple cases of dyadic conversation (Toerien, 2013; Yin, 2017), then descriptives on coded data were explored. Grounded on the IDZ framework, we conducted a template analysis (King, 2004), a type of thematic analysis on the coded data.

For qualitative analysis, initially two researchers dual-coded 46% of the video data (along with verbatim transcripts) using the coding scheme (Ruvalcaba et al., 2016). Because the scheme was originally designed for 1 C, challenges immediately emerged during the trial coding sessions with 2 C data. For example, we lacked a category that

Table 3. Frequency of conversation categories in Study 1.

Categories	Sub-categories	Melony and Rupert (%)	Dorothy and David (%)	Sandy and Anthony (%)	Clara and Luke (%)	Total (%)
Question	*Question*	6.7	10.0	27.3	19.4	18.1
	Respond to question	6.7	0.0	13.6	11.1	9.6
Discussion	*Agreement*	13.3	0.0	0.0	8.3	6.0
	Disagreement	0.0	0.0	4.5	8.3	4.8
	Explanation	53.3	60.0	36.4	36.1	42.2
	Suggestion	20.0	30.0	18.2	13.9	18.1
	Reminder	0.0	0.0	0.0	2.8	1.2
	Antagonistic	0.0	0.0	0.0	0.0	0.0

Table 4. Each student's contribution and teacher's feedback on collaboration in Study 1.

Student pairs	Contribution (%)	Teacher's feedback on collaborate (f)	Time to solve the problem (min:sec)
Sandy	60.6	5	5:30
Anthony	39.4		
Dorothy	59.2	1	3:53
David	40.8		
Melony	41.7	0	2:12
Rupert	58.3		
Clara	44.4	3	14:00
Luke	55.6		

captured students' explanations of what they were doing when working on their own screen. This resulted in several modifications so that it would align with the 2 C setting. The modified scheme had three major categories: Question, Discussion, and Manipulation. *Question* and *respond to questions* are under the category Questions. *Agreement, disagreement, explanation, suggestion*, and *antagonistic action* are under the category Discussion. Manipulation had the sub-categories: *edit with consent* and *edit without consent* (Table 1). The modified scheme was tested through pilot (dual) coding 24% of the data and then revised again, resulting in the final coding scheme (Table 1). We used time-based video segmentation to code where we divided videos of each dyadic collaboration into ten-second intervals and coded each interval with conversations categories, where the categories are not mutually exclusive. We finally dual coded 21% of the video data (using different data was used at each step of coding) that had an average kappa of the interrater agreement was 0.847. All the subcategories had a kappa above or equal to 0.70 except for *disagreement* which had a kappa of 0.66 (Table 1). The coders resolved disparities through discussion, and one of the coders then coded the rest of data. We then reflected on the coded data through template analysis and found several themes which we then related to the characteristics of Mercer's types of talk.

For the quantitative analysis, the coded frequencies of sub-categories from the data were used to corroborate the qualitative case studies and thematic analysis. Log data from NetsBlox activity that recorded the sequences of programming steps were also used to determine each partner's contribution to the code by calculating the total number of moves each partner had (Table 2). Both data sources were derived in parallel, then synthesized, to support findings from the discourse analysis (Tables 3 and 4).

Although there was not an explicit feedback intervention in this study, we felt it was important to capture what feedback teachers did provide. Thus, one coder open-coded instances of teacher feedback on collaboration (Table 4).

Results

We first provided quantitative descriptive findings, then a qualitative study of cases based on our interpretation of students' use of conversation categories during collaboration which subsequently followed by thematic findings.

Among all categories (Table 3), the most frequent sub-category was *an explanation* (42.2%); *question* (18.1%) and *suggestion* (18.1%) were used moderately; *respond to question* (9.6%), *agreement* (6%) and *disagreement* (4.8%) were minimally used, and *antagonistic action* was not identified.

In this 2 C setting, students edited without their partner's verbal consent almost 100% of the time.

Dorothy and David

Dorothy and David both actively searched for blocks that would help to solve the problem, which they did in about 4 minutes. David gave a lot of *suggestions* and Dorothy seemed to sincerely respond to the *suggestions* and *explanations* by acknowledging and implementing David's proposals. They also used *questioning*, and both were observed to *respond to each other's questions*. A seemingly concerned Dorothy said, "I found the 'when,' but I don't know" and David peeked at Dorothy's screen to respond and said, "Oh I think I know what it is, maybe it's the 'if' one" and then corrected his idea to "no, if else." This pair listened to each other's *explanations* closely by stopping their own work and looking at each-others screen while also responding very frequently. To confirm what they were supposed to do, Dorothy read the instruction "if touching sprite then stop." David took the hint and said, "oh, stop," but then *suggested*, "maybe we should take out this 'forever' then."

Although David did many of the editing tasks, the pair communicated through *suggestions* and *explanations* to solve the problem together. Both stayed on task and neither got any feedback from the teacher.

Clara and Luke

Clara initially struggled to log into NetsBlox. During that time (50 seconds), Luke edited the script and when he thought he was finished said, "I did it." The teacher then asked if he worked with his partner, to which Luke said no. At the teacher's prompting, Luke stopped editing and waited for Clara to get ready. It was three and a half minutes into the activity and there had been no conversation within the dyad until this point. Clara then found that her "snowflakes" sprite was not visible and although she asked Luke for help first; then within 2 s, she asked the teacher. After solving the problem, they scrolled through the block categories to search for blocks to solve the problem. Unlike David and Dorothy, this dyad did not explain to each other what they were editing and searching for. After several more minutes, Clara *explained* to Luke that she could not find the exact conditional block that would solve the problem. When she found the "if-else" block, she began to work without any *explanation*. Luke seemed to *agree* with her work by nodding and began to edit but did not *explain* what he was doing. At one point, the pair *disagreed* on the use of one block and, without coming to consensus, continued editing. Because the pair used little *explanation* with each other, they often got confused about each other's work. For example, when Clara *questioned*, "what are you doing?" Luke *explained*, "because it says forever if touching." Then, Luke's screen froze, and he exited from the collaborative space, though Clara continued working on her screen without talking. Meanwhile, Luke sat and watched another group work. After five minutes of sitting, Luke seemed to get agitated, shaking hands and rapidly moving on his chair. He *reminded* Clara to re-link the project with him saying, "Invite me in, invite me in." After several *reminders*, Clara noticed and re-shared the project. After getting back into the project, Luke immediately started editing without *explanation* of what he was doing and finished the project.

Although the log data show that the pair almost equally contributed to the editing, their synchronous collaboration was interrupted because of a technical problem with

Luke's screen as well as the lack of *explanation* from either (Table 4). As Table 3 shows, they are the only pair using most (all but *antagonistic*) of the conversation categories. The teacher's feedback was frequent for this dyad. The teacher encouraged them to communicate with each other three times during this activity and each time the dyad immediately engaged in conversations after the feedback (Table 4).

Anthony and Sandy

From the beginning, Anthony showed enthusiasm for participating in the activity. He said with a heightened voice, "I can't get in," to which Sandy calmly replied, "because I haven't yet invited you." After getting into the project, Anthony said "oh, I know this one, I can figure out how to do this one" and then *suggested*, "when sprite four touches sprite three, sprite three run." Anthony talked a great deal and at a faster pace while offering many *suggestions*; however, Sandy was unresponsive to Anthony's *suggestions*. Noticing the lack of communication, the teacher gave them feedback to divide work between themselves and talk saying "it's good to talk to each other, like, say I'll do this part and you'll do that part." Followed by the feedback, Sandy *explained* what she would do." The teacher tried to reconfirm "so what is he doing?" Sandy did not respond and looked at Anthony. The teacher asked how the pair were planning to work and suggested to work collaboratively. Here, both of them separately talked about certain aspects of the problem. After the teachers' feedback, the pair began talking about the task together. Anthony edited the "forever" block and Sandy disagreed with his edit by saying "I don't think that's 1000" to which Anthony wanted to play the program *suggesting*, "let's reactivate the program." When Sandy resumed editing again, Anthony said, "wait, stop for a sec so I can, I'm trying to retype this." Then Anthony edited the "glide" block and provided an *explanation* of what he was doing.

Unlike Dorothy and David who had more *explanation* (60% of the pair's conversion), Sandy and Anthony (36.4% of the pair's conversation) did not explain as often what they were doing unless prompted to do so by the teacher. The teacher gave feedback five times during their collaboration. They demonstrated all but three conversation categories (not *agreement, reminder* and *antagonistic*) during their collaboration (Table 3).

Melony and Rupert

Suggestions and *explanations* from Rupert were frequent during this collaboration. Although Rupert made *suggestions*, he did not seem to expect responses. Instead, he edited the blocks without waiting for Melony's confirmation. Rupert said, "so first we have to do the 'if'" and then edited the script. Again, he *suggested*, "so we can just do 'glide' then." At one point he *suggested* splitting up the work saying, "I'll do the top one and you can do the bottom," but edited both parts. When Melony began to edit two minutes into the activity, Rupert said, "I got it" and finished the work. Melony almost never talked or edited the code blocks during that time.

Rupert is the only student from this sample who participated in two sessions, thus he had prior experience solving a similar problem. Table 3 shows that, compared to other pairs, this pair used the lowest amount of *questioning*. Although there were many instances of *explanation* and *suggestion*, most were uttered by Rupert.

Themes found in dyadic collaboration

Analyzing the conversation categories, we could see that students were sometimes Exploratory by being constructively critical–providing productive challenge–about their collective work, and sometimes Cumulative by only confirming partners' work. Four themes emerged for the thematic template analysis on the coded videos. The first resonates with Exploratory talk where students critically question and discuss with each other; the following two are characterized by one student dominating and are less generative. The fourth pattern indicates that with teacher feedback, students could be encouraged to be mutually responsive.

a) Question and discuss while responding. We found this to be the most active collaboration mode. This is reflected in much of David and Dorothy's conversations. In this type of collaboration, partners respond to each other's explanations, suggestions, or questions with additional suggestions or explanations. While this type of questioning might be characterized as Exploratory talk, the coding scheme did not allow us to differentiate its definitive characteristics, as we did not have specific categories such as use of challenging questions, justifications, or sharing alternative ideas to be able to characterize it as Exploratory.

b) Edit without responding. In this type of collaboration, partners only use explanations and suggestions while working independently. We observed one partner explaining what they were doing or making suggestions without the presence or expectation of any response or one partner being passive by only listening or even ignoring their partner. We could see this in much of Melony and Rupert's collaboration, as Rupert took charge of the activity and Melony provided little input. As Rupert had previous experience in programming, Melony might have considered Rupert the authority in the dyad (Lewis & Shah, 2015). This type of conversation where one person is trying to keep an assertive relationship is closer to Cumulative talk in IDZ. However, in Cumulative talk, collaborators do respond to each other through agreements to suggestions. Since the 2 C setting allows partners to work simultaneously, dominating partners may assume they are collaborating through explanation and suggestion without having to confirm or negotiate with their partner.

c) Does not question, discuss, or respond. Sometimes there was little or no communication between partners. Even when a partner asked questions, explained or suggested, the other partner continued to edit without responding. Both Melony-Rupert and Clara-Luke dyads showed this characteristic in many instances. Although there was no Disputative talk like antagonistic actions or disagreements between partners, it was not productive, as partners did less knowledge sharing.

d) More conversation with teacher feedback. We found that when teachers reminded students to collaborate, students often improved their level of productive conversation, if only briefly. Asking questions, explaining their actions and making suggestions were common immediately after teacher feedback. During the total time the dyads took to solve the problem, Anthony-Sandy was reminded five times to collaborate, Luke-Clara three times, Dorothy-David once while Melony-Rupert were never reminded (Table 4). In Anthony and Sandy's collaboration, we observed how they became engaged in conversation every time the teacher reminded them. These findings parallel general findings in the literature that we need to explicitly teach students how to collaborate and model patterns of productive collaboration (Mercer et al., 2004, 1999; Rojas-Drummond & Zapata, 2004).

Discussion

In this first study, we wanted to investigate the collaborative talk students engaged in during 2 C and, in doing so, refine our coding instrument.

Study 1 offered a preliminary validation of methods of analyzing 2 C field data using the Ruvalcaba et al. (2016) coding scheme. The scheme, refined through trial coding sessions, helped us to parse students' conversation into a detailed analyzable format. However, there were two major challenges with the coding method we used. First, it was difficult to differentiate Cumulative and Exploratory talk, which restricted us from identifying productive conversation during collaboration. Thus, for Study 2, we needed to explore how to operationalize these two types of talk through the coding scheme. The second challenge was with the technique we used to segment the video data. Prior research using analyzing conversation has employed both time-based (such as the type we used in Study 1) (e.g., Baines et al., 2009; Ruvalcaba et al., 2016) and event-based video segmentation to code (Schegloff, 1991). Although the 10-s interval-based coding provided a systematic method for quantifying the conversations, we lost important contextual and qualitative information. In Study 2, we switched to event-based coding, where we coded turns of talk by each partner – allowing us to code with higher specificity and granularity in utterances.

The themes derived in Study 1 showed, as would be expected, that dyads varied in terms of their level and quality of discourse. Some dyads were not responding to or discussing work productively with their partners. This lack of task-related dialogue may be related to the 2 C setting where two sets of computers lowers the need to negotiate changes to the code. Thus, pairs may not be leveraging the potential benefits of collaborative work through verbally negotiating the problem-solving process. Previous research on students' computer-based collaboration using platforms that provided opportunities to divide and conquer tasks show that partners may vary in patterns of interaction, marked by individual rather than joint work (Barron, 2000). In our previous study comparing 1 C and 2 C, we found that students' collaboration can turn into cooperation by working separately on individual sections, thus restricting the opportunity to work collaboratively on coding challenges (Zakaria et al., 2019). Although cooperative work through a division of labor is useful in some contexts, a primary goal of pair programming in an educational context is the shared work on a common problem that can both further understanding through negotiated meaning (intersubjectivity), and the ways in which ZPD can be leveraged through dyad members' unique abilities (Fawcett & Garton, 2005). Dyads who had difficulty with collaboration tended to increase their talk when the teacher reminded the dyad of the expectations for collaboration. However, teacher feedback was not consistent in this study as we did not purposefully implement a structured support system for students. The teacher merely reminded students to collaborate without providing explicit guidelines on how to collaborate, sometimes even asking students to divide work among themselves. Studies point to the problem of learning to collaborate versus learning through collaboration and suggests that teachers often may not understand the mechanisms of productive collaboration or do not recognize the necessity to instruct students on the norms of collaboration (Murphy & Henessey, 2001). To be effective, teacher feedback needs to specifically target learning strategies that cater to the task on hand (Chan & Lam, 2010; Hattie & Timperley, 2007; Zimmerman & Kitsantas, 2002).

Thus, for Study 2, we incorporated an instructor-driven support system for students to encourage the use of quality Exploratory talk through collaborative modes. The main goal of Study 2 was to explore how explicitly guiding students in productive collaboration and consistently giving feedback on the quality of collaboration can impact the quality of their discourse.

Study 2

Guided by the insights and subsequent recommendations that emerged from Study 1, our goal was to examine if a structured feedback intervention could help students use more Exploratory talk in their conversation. Accordingly, in Study 2, we explored the following research question: *What role does feedback play in students' collaborative talk during 2 C pair programming?* A secondary goal for this study was to continue to review and modify the coding scheme to identify Mercer's (2002) three types of talk more accurately. Thus, the focus is on how feedback does (or does not) assist with prompting students' collaborative problem solving and help counteract the tendencies to work individually without communication as seen in some dyads in Study 1.

Method

The study took place at a suburban elementary school in the Southeastern United States. The students were from four classrooms but taught by the same technology teacher following the same curriculum. Study 2 made use of Scratch (Resnick et al., 2009), as the teacher and students were most familiar with this environment. This variation of 2 C with Scratch had unlinked workspaces which did not allow for a synchronized workspace. Dyads were seated side-by-side (per the 2 C setting) where they were instructed to create identical programs on each of their computers in separate workspaces. The teacher directed the task that required the pairs to collaboratively create a game wherein the user follows rhyming clues to click on certain sprites that move or speak in response.

In this intervention study, we randomly selected two classrooms to constitute our feedback group while the other two were our control group. The participants were 62 (control = 30; intervention = 32) fourth grade students (9–10 years) who formed 31 dyads (formed by the teacher). In-depth analysis was conducted on 12 randomly selected

Table 5. Expectations for collaboration presented to the students.

Control and intervention group	Additional terms for the Intervention group
(1) This work needs to be happening collaboratively.	(1) Information is shared openly.
(2) Each partner is responsible for the coding, problem-solving, and debugging.	(2) Listen to each-other and consider each-others suggestions.
(3) Make sure you both agree on your plans before you make changes.	(3) Actively encourage your partner to contribute to the discussion.
(4) The partners talk through the problem the entire time.	(4) Welcome challenges from your partner.
	(5) Challenge ideas with questions.
	(6) Justify your ideas to your partner.
	(7) Appreciate your partner's effort.

Table 6. Feedback Structure.

Session	Serial	Feedback
1	First	Sticky note: Try challenging your partner's idea by asking questions. Verbal addition: Questions with a "why" like "why do you want to do that" would help to challenge each other's ideas.
	Second	Sticky note: Put effort on sharing alternative ideas. Verbal addition: Sharing alternative ideas to your partner would help to create the best idea together.*
2	First	Sticky notes: Questions asking for explanations help you decide the best action. Verbal addition:*
	Second	Sticky note: Justify your idea with "because". Verbal addition: When you have an idea or you are answering a question try to provide a reason. Like "I think we should change the color of the text because it is too bright."
3	First	Sticky note: Check yourself how much are you challenging your partner. Verbal addition: Do you have a question to ask right now?*
	Second	Sticky note: Check yourself how much you are sharing alternative ideas. Verbal addition: Do any of you have a different idea to share now?*
4	First	Sticky note: Ask your partner for justification for an idea. Verbal addition: Can any of you ask for a justification right now?*
	Second	Sticky note: The more you practice coding with collaborative strategies, you get better at it. Verbal addition: What strategies make good collaboration? (If no response, told the list of the three strategies – challenging with questions, sharing alternative ideas. justifying own ideas)

*Provided contextual examples from what they were doing at that moment.

participants forming 6 dyads, three from each group (10 girls and 2 boys) from all the pairs that had recordings of good video and audio clarity. Activity sessions were once per week for four weeks (average of 26 active minutes/week) and the dyads were the same across all the sessions. We used video cameras with synced headsets to capture audio and video of the dyads. Verbatim transcripts of pairs' verbal interactions were created.

At the beginning of each session, we verbally presented to the whole class the expectations for collaboration to both the control and intervention groups (see Table 5). In the feedback group, the terms also included information on what constitutes good collaboration. As the students programmed, we provided pre-structured feedback to students in the feedback group focusing on the vital characteristics of Exploratory talk (Mercer, 2002) (i.e., *challenging partners with questions, sharing alternative ideas, justifying ideas, or disagreeing with justification*). Informed by growth mindset theory (Dweck & Leggett, 1988), the importance of effort and use of appropriate strategies was incorporated into the feedback as well. The feedback–a different set statement for each session–was provided on a sticky note twice per session (10 minutes into the activity and then 10 minutes after the first feedback) by the first author (Table 6), with students asked to read the feedback aloud from the note upon delivery. In addition, at the time the sticky note was provided additional verbal feedback was also given related to students' current collaboration practices. It took around 30 seconds to a minute to complete each feedback.

In the last week of the intervention, all participants (n = 32) in the intervention group completed anonymous reflections on the feedback. They responded to three questions (*Was the feedback helpful? What was helpful? and What was not helpful?*) about the feedback on a printed questionnaire.

Data coding and analysis

Based on our pilot study and Study 1, we developed a new coding scheme (Table 7). Our revisions resulted in new categories reflecting Exploratory talk: *sharing alternative ideas*,

Table 7. Conversation categories in the coding scheme and the inter-rater agreement (kappa) for Study 2.

Type of Talk	Categories	Description	Example utterances
Cumulative Talk	*Agreement*	Show explicit agreement with partner's idea, suggestion, or change. k = .847	"Yeah, good idea", "Okay".
	Self-Explanation	Starts explaining themselves what they are doing or thinking while working on own platform. Like a think aloud. k = .716	"Okay, so move 10 steps.", "I've to do the sailboat"
	Suggestion	Student makes a suggestion or shares an idea for the next step. Also, reminders would fall under suggestion. k = .767	"You have to change the number to 80." Words like "we/you/I can", "we/you need", "let's", "we/you have to"
	Controlled Direction	Gives step-by-step direction to their partner. k = .799	"Glide 2 sec, then go back to the sailboat",
	Seeking help	Student directly or indirectly seeks help from partner. k = .734	"I'm confused", "What shall we do?"
	Simple question	Question about a process or fact. (any type of questions which are not higher order) k = .903	"That?", "Which one?" "What are you doing?,"
	Coordination	Monitoring task or group progress, Coordination of task or group process. k = .822	"You're always one step ahead," "Wait for me please," "Marshall, focus!"
Exploratory talk	*Higher-order question*	Student questions to challenge their partner's ideas. These questions should be asking for reasoning. k = .811	"Why did you move it?", "Why?", "What happens if you keep it that way?"
	Alternative idea	Share an idea as an alternative to what the partners was suggesting or editing. k = .695	"No, Let's change it to blue", "May be try three seconds"
	Justification	Students justify their idea with reasons. Or evaluates a step as reasoning. May include words like "because"/"cause". k = .773	"Five seconds is too long", "It has to be the glide block cause it needs to move"
	Disagreement (followed by justification)	Shows disagreement with partner. k = .854	"No", "But that's gonna be like little weird.", "I don't think that's gonna.
Disputational talk	*Disagreement* (with no justification)	Shows disagreement with partner. k = .854	"No", "Nope, I don't want it that way."
Other	*Off-task*	Utterances which are not task oriented. k = .908	"I feel like I am getting stalked", (any talk outside of group)"
	Other	Utterances that do not fit into any other codes. k = .692	

k = average kappa of each category between the coders. Kappa range from .60 to 1 throughout all the categories.

asking *higher-order questions* (questions that challenge partner's ideas such as "why" questions), giving *justification;* three categories to Cumulative talk: *seeking help, coordination, controlled direction* and we considered conversation as Disputational talk when *disagreements* occurred but that did not include any *justifications*. We added *Disagreement with justification* to Exploratory talk. We removed two categories *editing with consent* and *editing without consent*, as they never got used in Study 1. We believed these added categories would help us to differentiate Cumulative and Exploratory talk much more explicitly, and thus better be able to identify productive conversation during collaboration. Each category was placed into one of Mercer's (2002) three types of talk (Table 7). In terms of feedback identification, as the feedback statements were pre-structured and time-specific, we decided not to code them.

Two pairs of coders, dual coded 47% of the data using the final coding scheme (Table 7). We coded each partner's turns of talk in a conversation (Schegloff, 1991). The categories were not mutually exclusive; thus, *kappa* was calculated for each category separately. Also, kappa was computed for each coder pair then averaged to provide a single index of interrater reliability (Light, 1971). The resulting average *kappa* indicated substantial agreement (*kappa*: average = 0.795, min = .749, max = .857) (Landis & Koch, 1977) (see Table 7).

Due to differences in session lengths, we normalized the data by calculating the total of each coded category per minute as a proportion of overall talk for that dyad. This normalized score of each category from video data was then used to explore significant differences in collaborative talk between the groups.

Results

To answer the research question, we utilized quantitative methods to directly compare the level of discourse from the dyads' collaboration between conditions. Figure 2 presents the overall distribution of the mean rate of conversation categories for each group. *Self-explanation* was most common throughout students' collaboration for both groups. *Simple question* (any questions not challenging the partner) was the next highest category. *Coordination, suggestion*, and *seeking help* were moderately used. However, categories that characterize Exploratory talk–*justification, sharing alternative idea*, and *higher-order question*–were minimally used by all students.

Each data point represents the mean rate of each conversation category used by dyads each week. In terms of growth in conversation rate throughout the study, a repeated measures ANOVA on four data points (one per week) per dyad (24 total across the sample) showed no significant difference in mean rate of conversation categories between feedback and control groups. Figure 3 shows progressions of estimated mean rate of Exploratory categories and *simple questions* from week one to four indicating there is growth in some of these categories in the first two or three weeks but then a decrease in the last week. We then followed up with a MANOVA test to find if there were significant mean differences between the groups regardless of time. Because of the small sample of data used, we also conducted tests to check if the assumptions of MANOVA were met. We found Cronbach's *alpha* of .781 indicating that the categories were sufficiently correlated to be considered as dependent variables in MANOVA. As the sample sizes were equal for both groups, we assumed the homogeneity of covariance was met. Shapiro-Wilk statistics

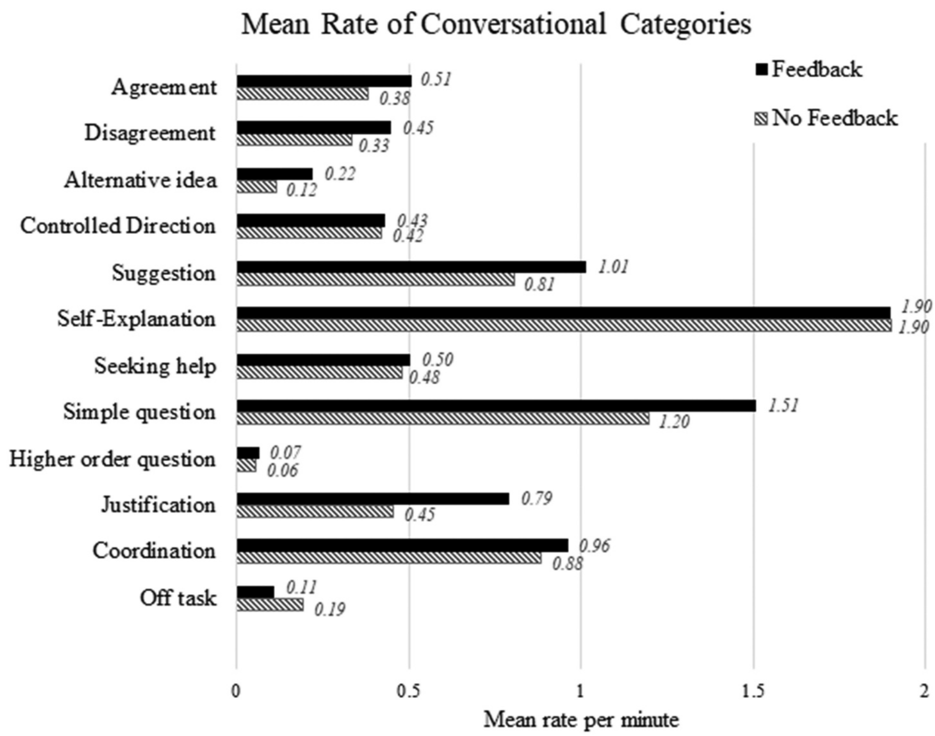

Figure 2. Mean rate of conversational categories per minute in both feedback intervention and control groups.

for the dependent variables (conversation categories) shows that except for *off-task, controlled direction* and *self-explanation* categories, all other conversation categories were normally distributed (Table 8). Research shows that when the assumption of normality is untenable, but the homogeneity of covariance is met, the parametric statistic is more robust, and even slightly outperforms the nonparametric statistic in terms of Type I error rate (Finch, 2005). Thus, we conducted a MANOVA that showed a significant difference in the use of conversation categories (all the categories as DVs) between the groups (IV), $F(14, 9) = 9.090$, $p = 0.001$; Wilk's $\Lambda = .066$, $\eta_p^2 = .934$. Because MANOVA statistics did not show which conversation categories were significantly different and to what level, we explored each category separately with Kruskal-Wallis H tests, a non-parametric alternative of univariate ANOVA (Table 9). Among the exploratory categories, *justification*, $\chi2(1) = 13.026$, $p = 0.00$ and *sharing alternative idea*, $\chi2(1) = 3.992$, $p = 0.046$ were significantly higher in the feedback group. However, despite that a substantial part of the feedback and pre-activity instruction emphasized asking challenging questions (i.e. "why" questions seeking justification) to the partner, students in both groups used minimal *higher-order questions* and did not show any significant differences. In contrast, the use of *simple questions* was significantly higher in the feedback group, $\chi2(1) = 5.749$, $p = 0.017$.

Because we were interested in the impact of instructor-directed feedback on student-student conversation, immediate uptake by students was examined. We found 26% of the

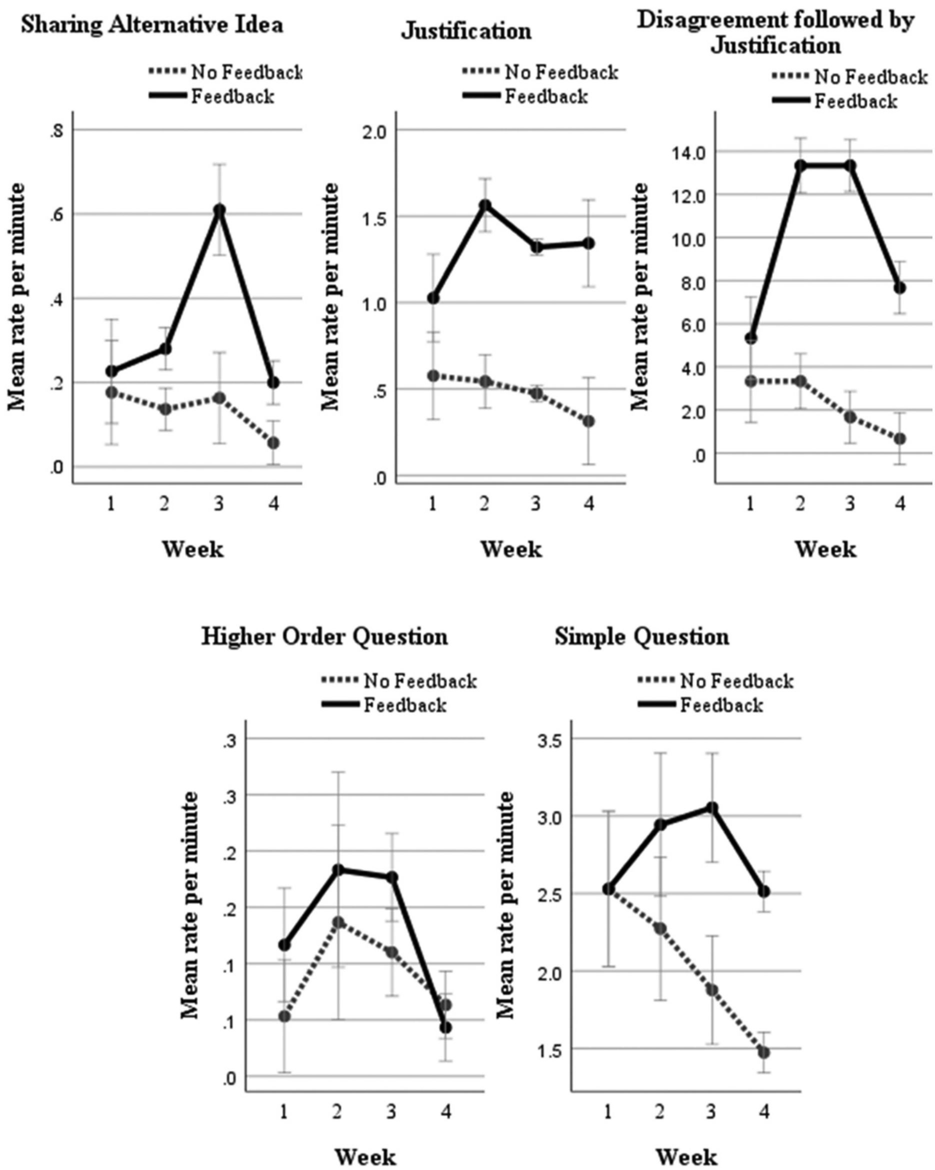

Figure 3. Mean rate of categories reflecting Exploratory talk and *simple question.*

time they used a *higher-order question* within two minutes of receiving feedback about asking challenging questions. For example, right after feedback ("Try challenging your partner's idea by asking questions."), Dyad F1 engaged in the following conversation:

A: *I have a question for you. Why do you want to do that color?*

B: *That would look really weird because it would blend in.*

A: *No, it won't. Because this is dark enough. See?*

Table 8. Shapiro-Wilk test of normality distribution for conversation categories.

Conversation categories	Shapiro-Wilk Statistic	
	Intervention	Control
Alternative idea	.940	.885
Justification	.888	.936
Higher-order question	.935	.901
Disagreement with justification	.964	.911
Disagreement without justification	.953	.879
Agreement	.866	.939
Controlled direction	.972	.872**
Suggestion	.968	.947
Self-Explanation	.898**	.916
Seeking help	.916	.947
Simple question	.974	.948
Coordination	.926	.898
Off-task	.511***	.559***
Other	.920	.903

df = 12

To similar feedback, Dyad F2 responded:

C: *I think we should put "when the sprite clicked" high*. (paused) *Say, "why?"*.

D: *Why?*

C: *Because and then when somebody finds it, then it would disappear.*

We analyzed another sequence of categories that reflects a desired category of Exploratory talk: *disagreement* followed by *justification*. Example from Dyad F3:

E: *Polar bear (suggestion).*

F: *No. (Disagreement) Because it will run off (justification).*

Students in the feedback group had a significantly higher rate of *disagreement* followed by *justification*, $\chi2(1) = 15.460$, $p = 0.00$, than the control group (Table 9).

Table 9. Test-statistics for univariate ANOVA main effects of Feedback intervention.

Main effect (between)	Kruskal Wallis H	*df*	Mean Rank	
			Intervention	Control
Alternative idea	3.992*	1	15.38	9.63
Justification	13.026***	1	17.71	7.29
Higher-order question	.660	1	13.67	11.33
Disagreement with justification	15.460***	1	18.17	6.83
Disagreement without justification	1.022	1	13.96	11.04
Agreement	8.344***	1	16.57	8.33
Controlled Direction	1.842	1	14.46	10.54
Suggestion	10.839***	1	17.25	7.75
Self-explanation	.563	1	13.58	11.52
Seeking help	1.470	1	14.25	10.75
Simple question	5.743**	1	15.96	9.04
Coordination	2.430	1	14.75	10.25
Off-task	1.847	1	10.54	14.46
Other	.008	1	12.63	12.38

*** The mean difference is significant at level .01
**The mean difference is significant at level .05

Disagreement without *justification*, which is considered Disputational talk, did not show significant differences between groups, $\chi2(1) = 1.002$, $p = 0.312$. Among other Cumulative talk categories, only *suggestions* and *agreement* had statistically significant main effects between the groups with the feedback group having the higher rate (Table 9).

Student reflection on feedback

Two pairs of coders open-coded students' reflections on feedback where we found that 90.3% of students reported that feedback was helpful. Investigating the strengths and limitations of the feedback, there were three themes related to strengths. First, most students' found that feedback helped collaboration by reminding them to ask questions, communicate with their partner, and give partner feedback. According to a student (S14), "It was very helpful. I was talking to my partner way more." Second, they felt the feedback helped their motivation and task progression by mentioning that it was "fun," (S21) "encouraging," (S7) and helped them work "faster" (S19) and "stay focused" (S6). Third, a few students found that the feedback enhanced their code practices stating that it "improved work" (S29) and that it "was helpful because if some of the coding was wrong we could help each other to make it right." (S12).

We also found three themes related to limitations. First, some students felt that the feedback lacked the necessary detail, mentioning that there were "not enough details and examples"(S5) and "some of the feedback on the sticky notes did not make sense" (S13). Second, two of the students found the feedback to be a distraction, with one noting "that it would stop us from what we were doing and make us lose our train of thought" (S27). Third, four of the students felt that they already knew the main strategies for collaboration saying, "We already know the target points" (S3). Also, of note, another student mentioned "My partner did not help me. So, no [the feedback was not helpful]!" (S21) serving as evidence that whoever is providing the feedback needs to remind both members of the pair to collaborate.

Discussion

In Study 2, we examined if structured feedback on dyadic collaboration along with explicit instructions on the expected characteristics of collaboration can impact students' level of collaborative conversation in this 2 C setting. The findings suggested that, for this sample, the conversation rate was significantly higher in dyads who were given feedback and instruction. This supports findings in other research that showed feedback on the use of certain strategies and effort improved performance (Dweck & Leggett, 1988), and helped students to incorporate those strategies into their work (Binglan & Jia, 2010; Burnett, 2002).

Perhaps most importantly, nearly all of the Exploratory categories appeared at significantly higher levels in the intervention group. In particular, students who received feedback had higher use of *justification* for the new and *alternative ideas* they shared as well as the *disagreements* that they had. While this might be expected as students were given explicit feedback on their use of *justification*, it is encouraging to see the level of uptake. Since programming problems can have many solutions, it is important for students to explore potential solutions and justify why they think it could be correct. Such critical reasoning processes may help them explore and make rationalized decisions (Besnard & Hunter, 2008). This process is found to be beneficial both for students' learning outcomes (Wegerif & Dawes, 1998) and their problem-solving capacity (Rojas-Drummond & Zapata, 2004).

The use of *higher-order questions*, which would indicate if they were asking challenging questions to each other, was not found to have any difference between the groups. However, use of *simple questions* – questions not challenging the partner, but rather asking about simple information – was found to be used significantly more by the experimental group. Reflecting back on the structure of the feedback system, we believe that our prompt related to using *higher-order questions* may have been unclear to students and they may have merely perceived it as a cue to ask any type of question. Previous literature suggests that feedback can easily be misinterpreted by students due to inadequacy of the feedback or other factors (i.e. consistency, accuracy, or comprehensibility of the feedback) not investigated in this study (Lee, 2008). In further research, we need to carefully word the feedback regarding questioning as well as demonstrate how challenging questions would be utilized in this context. Considering how challenging it is for students at this age to participate in argumentative collaboration (Bell, 2004), it is also plausible that this more cognitively challenging discursive move requires longer and consistent training with the teacher and their students to be utilized effectively.

Discussing with students the feedback prompts they received is likely to provide useful data for future refinement of our feedback strategy (cf., Hedgcock & Lefkowitz, 1994). While students were, overall, positive about the feedback, a few of them felt that the feedback was distracting and/or redundant. These thoughts shared by the students spurred us to reflect on the structure of the feedback, which confirmed that perhaps some of the feedback content could be perceived as repetitive. Although we specifically wanted to reiterate the strategies as a reminder to students, we now believe it may have been more beneficial to make the feedback personalized and adaptive by incorporating examples grounded in their specific programming activity, thus helping students better understand what we were striving for.

In both Study 1 and 2, we found the use of *self-explanation* (*explanation* in Study 1) to be significantly higher than the other categories. The high rate of *self-explanation* was not surprising given the opportunity in these 2 C pair-programming settings to work concurrently (Zakaria et al., 2019), thus heightening the need to coordinate among themselves by explaining what they were working on and thinking. While considered Cumulative talk, we see this as a positive quality as more *self-explanation* often leads to greater understanding (Chi et al., 1994) as well as more communication between partners to monitor their progress.

Reflecting on the revised coding scheme in Study 2, adding new categories helped us to classify Exploratory talk in more detail. Although we needed to sub-code categories like *disagreement* with *justification* when *disagreement* code was immediately followed by *justification* code and *disagreement* without *justification*, it was important to keep primary codes *disagreement* and *justification* separate. Mercer's (2002) framework emphasizes students' ability to disagree with appropriate reasoning as an important characteristic of Exploratory talk whereas only disagreeing without any justifying elaboration is Disputational and considered unproductive. Thus, this differentiation in the coding helped us explore these characteristics in detail. Additionally, analysis segmented by turn of talk in Study 2 helped to get a more accurate count of the number of instances for each category used and to better contextualize the discourse.

Limitations

As with all qualitative analysis, there are risks of interpretation bias, which we diligently tried to minimize through multi-coder corroboration and by striving for a high level of objectivity in our interpretations. The analytic-intensive nature of mixed-methods studies that involve verbatim transcripts and multi-coder concurrence, limited the total number of students we were able to study. We consider such a small sample size as a limitation of Study 2; however, this sample size is common for this type of annotation-intense video analysis work (e.g., Deitrick et al., 2017; Lewis & Shah, 2015). Still, this analysis of the intervention study provided enough statistical power to guide the development of future intervention studies utilizing 2 C pair programming. Additionally, we chose to randomly select dyads from the data that had good quality audio and video rather than developing a purposeful or stratified sample based on factors such as competence level or gender. Our aim was to explore collaborative practices and the impact of feedback on students more broadly in this initial set of studies. However, in our future studies, we intend to pursue research questions that might benefit from such a sampling method.

Future studies should strive to enlarge both the number and diversity of students studied. Another limitation is that we did not measure the existing collaboration norms of the classrooms. Although there was a single teacher of record for the students in each study, these classes all had different primary teachers that they spent most of the school day with. Both studies took place at suburban elementary schools, with Study 1 in an academically gifted classroom while Study 2 was with classrooms of students with weekly exposure to technology/coding experiences. Thus, our populations may not be representative of a larger body of students of this age. Finally, there can also be a possible Hawthorne Effect for the feedback; students with microphone headsets on, cameras in their space, and extra lessons on how to appropriately talk to each other might talk differently as a result of the attention and not the feedback themselves.

Conclusion

Our studies support the notion that we cannot assume that students inherently will know how to engage in productive collaboration practices and discourse. These two investigations demonstrate that before we can begin to design effective student scaffolds, it is imperative to characterize what collaborative discourse looks like in a relatively new and underexplored context, 2 C pair programming with elementary students. Although Mercer's (2002) framework provided an excellent foundation based in constructivist theory, iterative development was needed to create a coding scheme that was contextually responsive enough to provide insight into elementary students' collaborative programming practices. Appropriately, this was achieved by integrating discursive moves specific to programming (Ruvalcaba et al., 2016) with Mercer's overarching framing. Our work demonstrates that before one is able to design thoughtful interventions and support, pilot studies are essential in developing effective methodological frameworks that capture the nuances of collaborative practices and discourse. Results from Study 1 helped gain important insights into students' natural tendencies in collaborative discourse in 2 C learning environments that were critical for developing a more refined and targeted study design. Study 1 showed discourse was highly varied by dyad and characterized mostly by one student explaining what they were doing as they took the liberty to edit their program without engaging in conversation with their partner. We also learned that teacher feedback has a possibility to influence the quality of students' conversation. We used these findings to implement an intervention study that utilized a feedback framework that was specifically designed to engage students in higher levels of collaboration as exemplified in Mercer's Exploratory talk. Findings showed the intervention did influence the small sample of dyads' discourse; however, there are still refinements that need to be made to our feedback system, such as modeling higher-order questioning. These two studies also helped us develop an enhanced analytic approach through our revised coding rubric that captured the more complex patterns and nuances of students' collaborative practices in 2 C environments. We encourage others to utilize or adapt our framework, as more studies in this area are needed with larger and more diverse populations. We hope this work has begun to shed light and start a fruitful conversation on the promise of supporting elementary students who are engaging in collaborative practices in pair-programming contexts so that they can maximize the benefits afforded through these experiences.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Science Foundation [1721000].

Notes on contributors

Zarifa Zakaria is a Ph.D. candidate in Teacher Education and Learning Sciences specializing in Educational Psychology at North Carolina State University. She is working as a research assistant under the supervision of Dr. Eric N. Wiebe at the Friday Institute of Educational Innovations. Her research interests include students' discourse during computer-supported collaboration, understanding students' motivation towards collaboration and ways to support students in collaboration. LinkedIn: https://www.linkedin.com/in/zarifa-zakaria-06450663/

Jessica Vandenberg is a PhD candidate in Educational Psychology at North Carolina State University. She is a College of Education Research Fellow at the Friday Institute of Educational Innovation where her research has largely focused on elementary-aged students' discourse as they learn how to collaborate in programming environments.

Dr. Jennifer Tsan is a postdoctoral researcher in the Department of Computer Science at the University of Chicago. Her research primarily focuses on supporting elementary students learning computer science. She completed her doctoral research on supporting collaborative programming between upper elementary students in the Computer Science department at North Carolina State University.

Dr. Danielle Cadieux Boulden is a Research Scientist at the Center for Educational Informatics in North Carolina State University's Computer Science Department. Her research interests include developing K-12 teacher and student capacity with computational thinking and computer science practices, digital game-based learning, and teacher self-efficacy for technology integration.

Dr. Collin F. Lynch is an Assistant Professor in the Department of Computer Science at North Carolina State University. His research is focused on the development of robust intelligent learning environments and adaptive platforms for Ill-Defined domains such as scientific writing, law, and software development. He completed his doctoral research in Intelligent Systems at the University of Pittsburgh where he was affiliated with the Learning Research and Development Center.

Dr. Kristy Elizabeth Boyer is an Associate Professor in the Department of Computer & Information Science & Engineering and the Department of Engineering Education at the University of Florida. Her research focuses on how natural language dialogue and intelligent systems can support human learning across educational contexts including within and outside the classroom. Her research group builds computational models of the processes and phenomena during dialogue and learning, and these models drive the adaptivity of intelligent systems.

Dr. Eric N. Wiebe is a Professor in the Department of STEM Education at NC State University and Senior Research Fellow at the Friday Institute for Educational Innovation. Dr. Wiebe has worked in computer science education research for the last 20 years. A particular focus has been on computational thinking, CS/CT curricular integration, and collaborative programming in Grades 4-8.

ORCID

Zarifa Zakaria http://orcid.org/0000-0002-0241-4018
Jessica Vandenberg http://orcid.org/0000-0001-6497-1840
Eric N. Wiebe http://orcid.org/0000-0002-5920-5225

References

Adams, D., & Hamm, M. (1998). *Collaborative inquiry in science, math, and technology*. Heinemann, 361 Hanover Street, Portsmouth, NH 03801-3912.

Alpay, E., & Ireson, J. (2006). Self-theories of intelligence of engineering students. *European Journal of Engineering Education*, *31*(2), 169–180. https://doi.org/10.1080/03043790600567027

Bailey, J. (2017). Open Broadcaster Software. *Software, OBS Project*, *62*. https://obsproject.com/

Baines, E., Rubie-Davies, C., & Blatchford, P. (2009). Improving pupil group work interaction and dialogue in primary classrooms: Results from a year-long intervention study. *Cambridge Journal of Education*, *39*(1), 95–117. https://doi.org/10.1080/03057640802701960

Bandukda, M., & Nasir, Z. (2010, June). Efficacy of distributed pair programming. In *2010 International Conference on Information and Emerging Technologies* (pp. 1–6). IEEE.

Barron, B. (2000). Achieving coordination in collaborative problem-solving groups. *The Journal of the Learning Sciences*, *9*(4), 403–436. https://doi.org/10.1207/S15327809JLS0904_2

Beauchamp, M. H., & Anderson, V. (2010). SOCIAL: An Integrative Framework for the Development of Social Skills. *Psychological Bulletin*, *136*(1), 39–64. https://doi.org/10.1037/a0017768
Bell, P. (2004). Promoting students' argument construction and collaborative debate in the science classroom. *Internet environments for science education, 3,* 115–143.
Bennett, N., & Cass, A. (1989). The effects of group composition on group interactive processes and pupil understanding. *British Educational Research Journal*, *15*(1), 19–32. https://doi.org/10.1080/0141192890150102
Besnard, P., & Hunter, A. (2008). *Elements of argumentation* (Vol. 47). MIT press.
Binglan, Z., & Jia, C. (2010). The Impact of Teacher Feedback on the Long-term Improvement in the Accuracy of EFL Student Writing. *Chinese Journal of Applied Linguistics (Foreign Language Teaching & Research Press)*, *33*(2), 18–34. http://www.celea.org.cn/teic/90/10060802.pdf
Bradbury, A., Wiebe, E., Vandenberg, J., Tsan, J., Lynch, C., & Boyer, K. (2019). The Interface Design of a Collaborative Computer Science Learning Environment for Elementary Aged Students. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vols. 63, No. 1, pp. 493–497). SAGE Publications.
Broll, B., Lédeczi, A., Volgyesi, P., Sallai, J., Maroti, M., Carrillo, A., Weeden-Wright, S.L., Vanags, C., Swartz, J.D., & Lu, M. (2017, March). A visual programming environment for learning distributed programming. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 81–86).ACM
Bruner, J. S. (1978). The role of dialogue in language acquisition. In *A. Sinclair, R. J. Jarvelle, & W. J. M. Levelt (Eds.), The child's conception of language* (pp. 241–256). Springer-Verlag.
Burnett, P. C. (2002). Teacher praise and feedback and students' perceptions of the classroom environment. *Educational Psychology*, *22*(1), 5–16. https://doi.org/10.1080/01443410120101215
Calder, N. (2010). Using scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, *15*(4), 9–14. ISSN: ISSN-1326-0286.
Campe, S. et al. (2020). "Pair programming in middle school: variations in interactions and behaviors." Computer Science Education 30(1), 22–46.
Chan, J. C., & Lam, S. F. (2010). Effects of different evaluative feedback on students' self-efficacy in learning. *Instructional Science*, *38*(1), 37–58. https://doi.org/10.1007/s11251-008-9077-2
Chi, M. T., De Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, *18*(3), 439–477. https://doi.org/10.1016/0364-0213(94)90016-7
Cockburn, A. (2004). Crystal clear: *A human-powered methodology for small teams*. Pearson Education.
Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage.
Cutts, Q., Cutts, E., Draper, S., O'Donnell, P., & Saffrey, P. (2010, March). Manipulating mindset to positively influence introductory programming performance. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 431–435). ACM.
De Lisi, R., & Golbeck, S. L. (1999). Implications of Piagetian theory for peer learning. In *A. M. O'Donnell and A. King (Eds.), The Rutgers Invitational Symposium On Education Series, Cognitive perspectives on peer learning* (pp. 3–37). Lawrence Erlbaum Associates Publishers.
Deiglmayr, A., & Spada, H. (2010). Developing adaptive collaboration support: The example of an effective training for collaborative inferences. *Educational Psychology Review*, *22*(1), 103–113. https://doi.org/10.1007/s10648-010-9119-6
Deitrick, E., Wilkerson, M. H., & Simoneau, E. (2017, August). Understanding student collaboration in interdisciplinary computing activities. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 118–126). ACM.
Dewan, P., Agarwal, P., Shroff, G., & Hegde, R. (2009). Distributed side-by-side programming. In *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*. IEEE.
Dweck, C. S. (2013). *Self-theories: Their role in motivation, personality, and development*. Psychology press.

Dweck, C. S., & Leggett, E. L. (1988). A social-cognitive approach to motivation and personality. *Psychological Review*, *95*(2), 256. https://doi.org/10.1037/0033-295X.95.2.256

Fawcett, L. M., & Garton, A. F. (2005). The effect of peer collaboration on children's problem-solving ability. *British Journal of Educational Psychology*, *75*(2), 157–169.

Fernández, M., Wegerif, R., Mercer, N., & Rojas-Drummond, S. (2015). Re-conceptualizing "scaffolding" and the zone of proximal development in the context of symmetrical collaborative learning. *Journal of Classroom Interaction*, *36/37*(1/2), 54–72. https://www.jstor.org/stable/44735711

Finch, H. (2005). Comparison of the performance of nonparametric and parametric MANOVA test statistics when assumptions are violated. *Methodology*, *1*(1), 27–38. https://doi.org/10.1027/1614-1881.1.1.27

Franklin, D., Hill, C., Dwyer, H., Iveland, A., Killian, A., & Harlow, D. (2015, February). Getting started in teaching and researching computer science in the elementary classroom. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 552–557). ACM.

Gweon, G., Rose, C., Carey, R., & Zaiss, Z. (2006, April). Providing support for adaptive scripting in an on-line collaborative learning environment. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (pp. 251–260). Montréal Québec, Canada.

Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in education: A literature review. *Computer Science Education*, *21*(2), 135–173. https://doi.org/10.1080/08993408.2011.579808

Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, *77*(1), 81–112. https://doi.org/10.3102/003465430298487

Hedgcock, J., & Lefkowitz, N. (1994). Feedback on feedback: Assessing learner receptivity to teacher response in L2 composing. *Journal of Second Language Writing*, *3*(2), 141–163. https://doi.org/10.1016/1060-3743(94)90012-4

Infante, C., Hidalgo, P., Nussbaum, M., Alarcón, R., & Gottlieb, A. (2009). Multiple Mice based collaborative one-to-one learning. *Computers & Education, 53*(2), 393-40.

Kavitha, R. K., & Ahmed, M. I. (2015). Knowledge sharing through pair programming in learning environments: An empirical study. *Education and Information Technologies*, *20*(2), 319–333. https://doi.org/10.1007/s10639-013-9285-5

King, N. (2004). Using Templates in the Thematic Analysis of Text. In C. Cassell & G. Symon (Eds.), *Essential guide to qualitative methods in organizational research* (pp. 256–270). Sage.

Lai, A. F., & Yang, S. M. (2011). The learning effect of visualized programming learning on 6 th graders' problem solving and logical reasoning abilities. In *2011 International Conference on Electrical and Control Engineering* (pp. 6940–6944). IEEE.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 159–174

Lee, I. (2008). Student reactions to teacher feedback in two Hong Kong secondary classrooms. *Journal of Second Language Writing*, *17*(3), 144–164. https://doi.org/10.1016/j.jslw.2007.12.001

Lewis, C. M. (2011). Is pair programming more effective than other forms of collaboration for young students? *Computer Science Education*, *21*(2), 105–134.

Lewis, C. M., & Shah, N. (2015). How equity and inequity can emerge in pair programming. In *Proceedings of the eleventh annual International Conference on International Computing Education Research* (pp. 41–50). ACM.

Light, R. J. (1971). Measures of response agreement for qualitative data: Some generalizations and alternatives. *Psychological bulletin*, *76*(5), 365.

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*(1), 63–86. https://doi.org/10.1007/s11412-006-9005-x

Mercer, N. (1995). *The guided construction of knowledge: Talk amongst teachers and learners*. Multilingual matters.

Mercer, N. (2002). *Words and minds: How we use language to think together*. Routledge.

Mercer, N., Dawes, L., Wegerif, R., & Sams, C. (2004). Reasoning as a scientist: Ways of helping children to use language to learn science. *British Educational Research Journal*, *30*(3), 359–377. https://doi.org/10.1080/01411920410001689689

Mercer, N., Wegerif, R., & Dawes, L. (1999). Children's talk and the development of reasoning in the classroom. *British Educational Research Journal*, *25*(1), 95–111. https://doi.org/10.1080/0141192990250107

Murphy, P., & Hennessy, S. (2001). Realising the potential – and lost opportunities – for peer collaboration in D&T setting. *International Journal of Technology and Design Education*, *11*(3), 203–237.

National Research Council. (2013) . *Education for life and work: Developing transferable knowledge and skills in the 21st century*. National Academies Press.

Nawrocki, J. R., Jasiński, M., Olek, Ł., & Lange, B. (2005). Pair programming vs. side-by-side programming. In *European Conference on Software Process Improvement* (pp. 28–38). Springer.

Pifarré, M., Cobos, R., & Argelagós, E. (2014). Incidence of group awareness information on students' collaborative learning processes. *Journal of Computer Assisted Learning*, *30*(4), 300–317. https://doi.org/10.1111/jcal.12043

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. B. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

Rojas-Drummond, S., & Mercer, N. (2003). Scaffolding the development of effective collaboration and learning. *International Journal of Educational Research*, *39*(1–2), 99–111. https://doi.org/10.1016/S0883-0355(03)00075-2

Rojas-Drummond, S., & Zapata, M. P. (2004). Exploratory talk, argumentation and reasoning in Mexican primary school children. *Language and Education*, *18*(6), 539–557. https://doi.org/10.1080/09500780408666900

Ruvalcaba, O., Werner, L., & Denner, J. (2016). Observations of pair programming: Variations in Collaboration Across Demographic Groups. In *Proceedings of the 47th ACM Technical Symposium on Computer Science Education (SIGCSE '16)* (pp. 90–95).

Schegloff, E. A. (1991). Conversation analysis and socially shared cognition. *Perspectives on Socially Shared Cognition*, 150–171. https://doi.org/10.1037/10096-007

Scott, S. D., Mandryk, R. L., & Inkpen, K. M. (2003). Understanding children's collaborative interactions in shared environments. *Journal of Computer Assisted Learning*, *19*(2), 220–228.

Shah, N., Lewis, C., & Caires, R. (2014). *Analyzing equity in collaborative learning situations: A comparative case study in elementary computer science*. International Society of the Learning Sciences.

Shah, N., & Lewis, C. M. (2019). Amplifying and attenuating inequity in collaborative learning: Toward an analytical framework. *Cognition and Instruction*, *37*(4), 423–452. https://doi.org/10.1080/07370008.2019.1631825

Soller, A. (2001). Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education (IJAIED)*, *12* (1), 40–62. https://iaied.org/journal/980

Toerien, M. (2013). Conversations and conversation analysis. In *Uwe Flick The SAGE handbook of qualitative data analysis* (pp. 327–340).

Tsan, J., Rodríguez, F. J., Boyer, K. E., & Lynch, C. (2018, February). I Think We Should . . . : Analyzing Elementary Students' Collaborative Processes for Giving and Taking Suggestions. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 622–627). ACM.

Tudge, J. R. (1992). Processes and consequences of peer collaboration: A Vygotskian analysis. *Child Development*, *63*(6), 1364–1379. https://doi.org/10.2307/1131562

Umapathy, K., & Ritzhaupt, A. D. (2017). A meta-analysis of pair programming in computer programming courses: Implications for educational practice. *ACM Transactions on Computing Education (TOCE)*, *17*(4), 16. https://doi.org/10.1145/2996201

Vygotsky, L. (1978). Interaction between learning and development. In *Readings on the Development of Children* (pp. 34–41).

Warwick, P., Mercer, N., & Kershner, R. (2013). 'Wait, let's just think about this': Using the interactive whiteboard and talk rules to scaffold learning for co-regulation in collaborative science activities. *Learning, Culture and Social Interaction*, *2*(1), 42–51. https://doi.org/10.1016/j.lcsi.2012.12.004

Wegerif, R., & Dawes, L. (1998). Encouraging Exploratory Talk Around Computers. In M. Monteith (Ed.), *IT for Learning Enhancement* (pp. 10–29). Intellect Press.

Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *IEEE Software*, *17*(4), 19–25. https://doi.org/10.1109/52.854064

Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In support of pair programming in the introductory computer science course. *Computer Science Education*, *12*(3), 197–212. https://doi.org/10.1076/csed.12.3.197.8618

Yin, R. K. (2017). *Case study research and applications: Design and methods*. Sage publications.

Zakaria, Z., Boulden, D., Vandenberg, J., Tsan, J., Lynch, C., Wiebe, E., & Boyer, K. E. (2019). Collaborative Talk Across Two Pair-Programming Configurations. In *Proceedings of 13th International Conference on Computer Supported Collaborative Learning (CSCL) 2019* (vol. 1, pp. 224-231). Lyon, France: International Society of the Learning Sciences.

Zimmerman, B. J., & Kitsantas, A. (2002). Acquiring writing revision and self-regulatory skill through observation and emulation. *Journal of Educational Psychology*, *94*(4), 660. https://doi.org/10.1037/0022-0663.94.4.660

Zumbach, J., Reimann, P., & Koch, S. C. (2006). Monitoring students' collaboration in computer-mediated collaborative problem-solving: Applied feedback approaches. *Journal of Educational Computing Research*, *35*(4), 399–424. https://doi.org/10.2190/2G3G-5M86-8474-76NV