International Journal of Computer-Supported Collaborative Learning
https://doi.org/10.1007/s11412-021-09349-3

Prompting collaborative and exploratory discourse: An epistemic network analysis study

Jessica Vandenberg[1] · Zarifa Zakaria[1] · Jennifer Tsan[2] · Anna Iwanski[3] · Collin Lynch[1] · Kristy Elizabeth Boyer[4] · Eric Wiebe[1]

Received: 1 March 2021 / Accepted: 8 July 2021 / Published online: 07 August 2021

Abstract
By encouraging elementary students to work collaboratively, they can gain essential skills such as perspective taking, conflict negotiation, and asking for and receiving assistance. Epistemic Network Analysis (ENA) is an analytic technique that provides an alternative to more typical approaches to analyzing and synthesizing coded dialogue. This study used an easy-to-implement prompting intervention in the context of collaborative (pair) programming with upper elementary students to demonstrate the potential of ENA to understand the impact of the intervention. We found that intervention students—those given empirically-derived prompts in support of collaborative and exploratory talk—asked questions, justified their thinking, and offered alternative ideas in ways that were both qualitatively and quantitatively different from control students.

Keywords Epistemic network analysis · Primary grades · Discourse · Pair programming · Collaboration

Introduction

Successful collaboration comprises discussion, shared decision-making, and joint engagement (Roschelle & Teasley, 1995). One of the affordances of a well-designed Computer-Supported Collaborative Learning (CSCL) environment is that these collaborative processes can be made more obvious to the participants, practitioners, and researchers alike (Shawky et al., 2014). Learners in a CSCL context co-construct their knowledge and regulate their own and others' learning in ways that are often visible; this can occur through the use of shared displays and discussion that makes learners' thinking conspicuous (Miyake, 1997). Central to understanding these processes through researching CSCL environments is the generation of theory-guided insight through deep, synthetic analysis of students' discourse.

✉ Jessica Vandenberg
jvanden2@ncsu.edu

1 North Carolina State University, Raleigh, NC 27695, USA

2 University of Chicago, Chicago, IL 60637, USA

3 University of Wisconsin-Madison, Madison, WI, USA

4 University of Florida, Gainesville, FL 32611, USA

Epistemic Network Analysis (ENA) has recently emerged as a powerful way to analyze this collaborative discourse. Discourse analysis hinges on the assumption that a speaker's words and phrases do not exist in isolation but are emergent and responsive to what both they and their partner have said earlier (Mercer, 2007). Thus, it is critical that coded discourse analytically acknowledge prior coded words and phrases. ENA assumes that temporal connections in coded data can highlight patterns and meaning in student cognition. Moving windows of time can be used to explore co-occurrences of codes within this window. The resulting models permit comparative analysis visually and quantitatively, with the support of direct qualitative evidence. The intervention study implemented here is used to demonstrate the potential of ENA, using an empirically tested and refined coding framework. The ENA research community has shown that this analytic tool is useful for discourse analysis using a range of studies including how teachers talked about technology integration in their classrooms (Zhang et al., 2019), the use of scientific collaborative discourse in gameplay (Bressler et al., 2019), and expressions of complex design thinking by university engineering students (Arastoopour et al., 2016). For our ENA analysis, we used coded student discourse, gathered as they worked on a shared programming task.

By encouraging students to problem solve on a joint task, they must consider their different ideas, varied strategies they might employ, how to overcome conflict, and the processes involved in decision-making. In computer-based settings, students tend to articulate their reasoning more explicitly than in more traditional face-to-face settings (Van Eaton et al., 2015). This articulation is central to collaborative work. Environmental and technological restrictions placed on students may affect how they think and talk about what they are doing and learning, however. A landmark study of computer-based learning (Teasley, 1995) concluded that talk dyads (partnered students required to talk through a scientific reasoning activity) produced more sound hypotheses than no-talk dyads, no-talk alones, and talk alones. It was not the mere presence of another, but rather the active, collaborative activity between the learners that drove higher cognitive activity. Roschelle and Teasley (1995) suggest that using a computer supports collaboration. Learners are able to indicate meaning and intent even when their language is not precise enough to describe their goal. In this type of context, it becomes easier to try out divergent ideas before deciding upon a solution. Finally, sharing of environmental space (i.e., relinquishing control of the mouse or trackpad) offers the opportunity to indicate a willingness to propose or accept new ideas.

Upper elementary students who are participating in such a complex socio-cognitive activity need support and training from teachers to be able to effectively negotiate and reason with varying ideas. We intervened in students' collaboration with simple prompting on strategies for better collaboration while they pair programmed. By exploring upper elementary students' collaborative discourse as they pair programmed, we found unique quantitative and qualitative distinctions between the intervention and control groups. Although the starting point for exploring the students' collaboration was with coding the dialogue using a framework modified from the work of Mercer et al. (1999) and Ruvalcaba et al. (2016), we then deployed ENA to garner more insight into the interplay of conversational codes. The ENA findings permitted robust quantitative analysis and descriptive analysis of the resulting visualizations allowing us to highlight narrative-driven distinctions in collaboration between the groups. This analysis provided us with information on how pairs used sets of conversational codes in close temporal proximity when sharing and implementing ideas together. By looking at the varying combination and frequencies of co-occurrences of codes used by pairs, we were able to effectively evaluate their collaboration level.

What follows is a brief literature review of ENA, which will be used to provide additional insight into the analysis of the coded discourse. Mercer and Littleton (2007) framework related to types of talk, which will be used to structure the analysis of the resultant discourse and collaboration through pair programming, is used to illuminate the way prompting may enhance the learning opportunities of this collaborative activity for elementary students.

Related literature

Epistemic network analysis

ENA combines traditional qualitative and quantitative methods with modern computing and data analytics to aid researchers in deep meaning making with their data (Shaffer et al., 2009). More specifically, ENA is a data analysis method centered on dimensionality reduction, modeling connections between concepts in coded data. ENA can easily produce visual networks of the conceptual connections. In summary, each concept, or meaningful feature in a dataset, represents a node in the network (Shaffer, 2017; Shaffer et al., 2016). Whenever this feature is present in a segment of data, that data is coded accordingly. ENA then uses this coded data to generate ENA networks based on the co-occurrence of codes in the dataset. By quantifying these co-occurrences, weighted network models are created in which the edge (or line thickness of the connection between nodes), the node size of a given code, and the node location relative to other nodes all can provide insight into the data (e.g., Arastoopour Irgens et al. 2015). Networks can be created for any unit of interest within the data, giving one the power to both visually and numerically compare various units in a more fine-grained manner.

ENA was created originally as a method to study discourse, cultural, and cognitive patterns, specifically within the domain of the Learning Sciences, Computer-Supported Collaborative Learning, and Learning Analytics (Shaffer et al., 2009; Shaffer, 2017; Shaffer et al., 2016). Shaffer (2006) maintains that epistemic frames represent the "linkage between practices and ways of knowing, but at the level of the local cultures developed by individual communities of practice" (p. 232). ENA was developed to operationalize these epistemic frames based on the knowledge that frequent co-occurring concepts in discourse indicate meaningful connections. Theoretically and mathematically, ENA can be seen as an evolution of an earlier generation of data visualization and graph-theoretic techniques in knowledge engineering, such as multidimensional scaling (MDS) and Pathfinder networks (Branaghan, 1990; Goldsmith et al., 1991).

In this study, we use ENA to analyze conversational data among pairs of students collaborating on solving a programming problem. Here, our rubric based on Mercer's framework (Mercer, 2000), forms the network nodes while both the experimental condition (prompt/no prompt) and time (week of the intervention) are the independent variables by which the network graphs are compared. We are not alone in our utilization of Mercer and ENA; in fact, Knight et al. (2014) used components of Mercer's typology with ENA and found that ENA "offers a representational tool for scalable interpretation of epistemic commitments, and that the notion of connections in epistemic frames is a productive characterization of epistemic commitments, offering more insight towards close qualitative [sociocultural] discourse analysis than simpler coding methods" (p. 157). This insight is gained by systematically looking across codes within time windows and then weighting the co-occurrences in a way that structures the results through dimensionality reduction in a visualization around selected variables of interest for sets of comparisons. It is our goal to build on the prior work of others' coding frameworks and further demonstrate the utility of using ENA for discourse analysis.

Mercer framework on types of talk

Mercer (2002) proposed three types of talk that define constructs in collaborative conversation, identifying different social ways of thinking. Mercer's framework among others is used to explore language artifacts (i.e., spoken utterances) of dyadic collaboration in a classroom. In collaborative activity, three types of conversations can occur, which Mercer termed Cumulative, Disputational and Exploratory talk. In Cumulative talk, speakers positively share and show a desire to understand each other without any critical challenges, whereas Disputational talk is characterized by unresolved disagreement and individualized decision making. In Exploratory talk, posited as the most productive type of talk, participants constructively engage in critical conversations, which lead to improved reasoning and conceptual understanding (Bennett & Cass, 1989; Fernández et al., 2001; Wegerif, 2005).

Mercer posits that Exploratory talk does not naturally occur, but rather needs to be facilitated through instruction, scaffolding, and prompting (Rojas-Drummond & Mercer, 2003). This explains our previous studies where we found that, in regular pair programming settings, elementary students tend to use significantly less Exploratory talk than Cumulative and Disputational talk (Zakaria et al., 2019). The use of prompts is common in classrooms and often takes the form of procedural, reflection, or elaboration prompts (Xun & Land, 2004). Procedural prompts help students complete specific tasks and learn new and/or use cognitive strategies. The types of prompts used in the current study align most closely with procedural prompts. With regards to designing classroom interventions, this is a minimalist approach, with only nominal training needed, and one that permits easy implementation by educators.

Mercer's work is a prominent guiding theory for the analysis of discourse and is often operationalized through a coding rubric applied to the transcript of dialogue. The use of such coding frameworks as an analytic tool to investigate collaborative discourse, including our own, has a long history in CSCL research. That history includes the work of Polo et al. (2016), Hennessy et al. (2016), Bungum et al. (2018), Nikolaidou (2012), Fisher (1993), Asterhan et al. (2014), Fernández et al. (2001), and T'Sas (2018). Studies that employed Mercer's framework to analyze students' conversations during pair programming have shown that elementary and middle school students commonly use Cumulative talk more than the other types of talk (Zakaria et al., 2019; Campe et al., 2020). Findings from the study by Rojas-Drummond and Mercer (2003) showed that teachers do not usually instruct students in effective collaborative talk. Their findings showed that, when taught, students did improve in using Exploratory talk (Rojas-Drummond & Zapata, 2004). In guiding teachers to facilitate Exploratory talk, Mercer et al. (1999) list a set of ground rules. Research asserts that, with teachers' support, students are able to participate in more Exploratory talk by co-constructing new ideas, and by challenging and appropriating each other's ideas to critically and productively contribute to the collaborative activity (Warwick et al., 2013).

Pair programming

Many researchers in computer science (CS) education agree on the general educational benefits of pair programming. At the center of pair programming research is the knowledge that such collaborative work has the ability to drive quality discourse. As such, we use pair programming as a rich context for gathering information on how young students not only talk through their thinking, but build upon each other's contributions in ways that push the dyad's thinking and learning forward. Studies have found that pair programming—conceptualized as two programmers working on the same coding problem—enhances students' overall enjoyment of and confidence in programming (McDowell et al., 2006), increases computer science competence for females and lower ability students (Maguire et al., 2014), some improvement in CS self-efficacy (Davidson et al., 2010), and higher grades (Mendes et al., 2005). An emerging approach, called two-computer pair programming (2CPP; (Tsan et al., 2020)) is being utilized with elementary students. At its core, 2CPP is when two programmers work on independent computers and in close proximity, but on a shared virtual workspace; updates to the code on one computer's workspace appear on the other's almost instantly. Collaboration works best when both partners feel capable of proposing ideas and having those ideas acknowledged (Barron, 2003). Raising group awareness by prompting students on ways to communicate productively can effectively resolve such imbalanced communication (Gweon et al., 2006) and can provide guidance to students on how to interact in a collaborative environment (Deiglmayr & Spada, 2010). By arming students with language and skills reflective of meaningful contributions to collaborative work, we hope to address some of the problems others have found when examining elementary students' collaboration in similar environments.

Current study

The previously published research supports the view that ENA is an effective method to analyze student discourse, where the cognitive connections students make and the language they use are of central interest. We know that technology use can enhance students' social interaction with others—either within the classroom or across the world—and access to information (Cress et al., 2015). The context of these social interactions can vary from collaboratively writing a play to debugging a computer program. Our interest in traditional pair programming is motivated by a desire to foster young students' collaborative and critical thinking abilities and because we know this context typically encourages students to verbalize their thinking, talk through their confusion, and to collaboratively make decisions. Our study investigates a prompting protocol implemented by the teacher–a simple but potentially powerful means of improving the quality of collaboration. In our work, Mercer's (2000) collaborative talk framework guided the development of a coding scheme—categories that underscored the role of decision-making and conflict in collaborative work—which we then analyzed using ENA. The utilization of the Mercer (2000) typology—nested within sociocultural theories—alongside the analysis approach of ENA is purposeful. Both posit that language and connections are meaningful and that the establishment of common ground between participants is important. As such, we approach this study with the goal of answering these two questions:

1. To what extent does the discourse of student dyads differ when a subset of students are prompted to employ collaborative talk strategies?
2. What patterns of collaborative talk, as explained by Epistemic Network Analysis, appear when students receive prompts?

Methodology

Participants

The participants were 62 fourth grade students (9-10 years old) from a single suburban elementary school in the southeastern United States. Four classes of students attended a technology class once-a-week, in which this study was nested. In that context, the teacher designed a pair programming project that used Scratch. Students were introduced to the Scratch platform at the beginning of fourth grade and used it for approximately 6 months before this intervention. The expectations were as follows: 1) students worked in teacher-assigned dyads, 2) the dyads negotiated whose riddle project they were going to program first, 3) each student had access to a computer, and 4) both students were expected to produce the same output. The students wrote riddle poems about objects 'hidden' in the scene they created and then wrote code to reveal the objects associated with each line as it came up in the poem. The weekly lessons occurred for four weeks. Six dyads were selected based on video and audio clarity and analyzed fully here. It must be noted that these data have been quantitatively analyzed using different methods and research questions, and reported in a previous paper (Zakaria et al., 2021); however, as we are moving forward to modify the prompts for further intervention, it is crucial for us to explore the data qualitatively, using Epistemic Network Analysis, to identify the nuances of the collaborative practices of the dyads.

Block-based programming environment: Scratch

Scratch (Resnick et al., 2009) is a block-based programming environment that is built for novices and has become popular for teaching programming to younger students. In block-based programming, users drag separate blocks of code from the left side of the interface

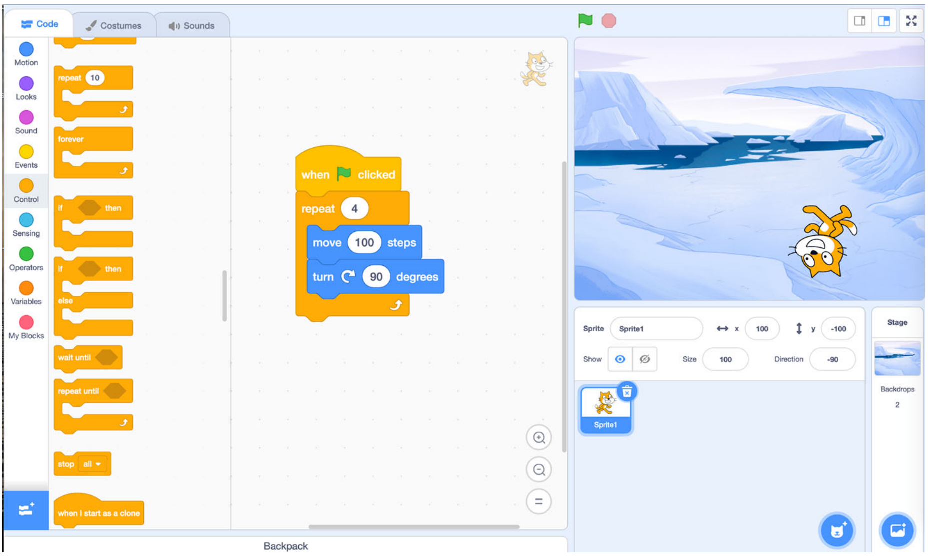

Fig. 1 Scratch programming environment interface

into the center work area. Each code block represents a single command and they fit together like puzzle pieces to create a script (i.e., program) as shown in Fig. 1. The script consisting of blocks of code can control sprites, which are characters, such as the Cat (Fig. 1) on the stage, which is located on the right side of the interface. The stage is where the users can see what happens when they run their scripts. In a typical programming task, dyads would be asked to collaborate together to make sprites do specific actions on the stage.

Procedure

Two of the four classes were given collaborative prompts during their programming task; henceforth, the students who received the prompts are deemed the intervention group and those who did not are deemed the control group. We designed the prompts based on Mercer's ground rules for collaboration, which focus on the essential characteristics of Exploratory talk (i.e., challenging partners with questions, sharing alternative ideas, justifying ideas, or disagreeing with justification) (Mercer & Littleton, 2007). The pre-written prompts were physically handed to the dyads on a sticky note and students were asked to read them aloud to the researcher. To make the prompts more relevant to the students' work, the researcher also added verbal reinforcements which included contextual examples for the prompts relating to what the students were doing at those moments. They were provided twice during the programming activities, approximately 10 minutes apart. Each time the researcher observed what the students were doing, handed the sticky notes to be read aloud, and then provided the additional verbal reinforcements. In line with Mercer's (2000) research, these prompts emphasized *exploratory* talk—suggesting alternative ideas, challenging a partner with questions, disagreeing, and justifying ideas. For example,"While coding this sprite, one strategy you can use right now is to share alternative ideas with each other." Moreover, the prompts encouraged students to put forth good effort and to try diverse strategies. For example, as a contextual example for a prompt, the researcher added "I see one of you is suggesting an edit to this code block. You can also try to challenge this suggestion by asking a question with a 'why' so that she can justify her suggestion." Table 1 displays the content of the sticky note prompts and the researcher's verbal reinforcements.

Consenting students wore headsets attached to video cameras, thus permitting us to capture quality audio and video of the collaboration. Six dyads, three from the intervention group and three from control classrooms, were selected for analysis. These dyads were randomly selected from a subset of those with the clearest audio and video recordings and who were present as complete dyads all four days of the intervention. Recordings were transcribed verbatim.

Analyses

We used a mixed methods approach to analysis (Creswell & Clark, 2017). First, we used a discourse analytic methodology to qualitatively code students' talk (Hardy et al., 2004). Then we applied ENA and statistical analysis to compare students' talk across treatments. To code students' conversation we needed a framework that could identify students' different types of talk. We started with Ruvalcaba's (2016) deductively determined codes, which were developed to analyze conversation in the context of pair programming. However, we could not successfully distinguish the occurrence of Exploratory, Cumulative, and Disputational talk using Ruvalcaba's codes. After multiple iterations, we developed a set of codes that were able to identify Mercer's (2000) original three types of talk in the context of pair

Table 1 Prompts, by day

Day	Sticky note prompts	Verbal reinforcement
1	• Try challenging your partner's idea by asking questions.	• Questions with a 'why' like 'why do you want to do that' would help to challenge each other's ideas.
	• Put effort on sharing alternative ideas.	• Sharing alternative ideas to your partner would help to create the best idea together.
2	• Questions asking for explanations help you decide the best action.	• [Researcher provided contextual examples from what they were doing at that moment]
	• Justify your idea with "because".	• 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	• Check yourself how much you are challenging your partner.	• Do you have a question to ask right now?
	• Check yourself how much you are sharing alternative ideas.	• Do any of you have a different idea to share now?
4	• Ask your partner for justification for an idea.	• Can any of you ask for a justification right now?
	• The more you practice coding with collaborative strategies, you get better at it.	• What strategies make good collaboration? (If no response, told the list of the three strategies – challenging with questions, sharing alternative ideas, justifying own ideas)

With each verbal reinforcement, researcher provided contextual examples from what the pairs were doing at those moments

programming. This was pilot tested and refined (Zakaria et al., 2019, 2021), resulting in the final framework that appears in Table 2.

In this coding framework we clustered codes under the types of talk. *Self-Explanation*, *coordination*, *suggestion*, and *simple question* comprise Cumulative talk. These four codes relate to the definition of cumulative talk wherein pairs attempt to share and understand each other without being constructively critical. *Higher order question*, *alternative idea*, *justification*, and *disagreement with justification* reflect the characteristics of Exploratory talk such as critically challenging a partner and sharing alternative hypotheses with ones partner. It is important to distinguish some of these codes from the Cumulative codes. *Higher order question* is substantively different from *simple question* in that this type of question challenges the partner explicitly by asking for reasoning. *Alternative idea* is a variation of *suggestion*. However, it is always an alternative suggestion to what the partner already suggested or is doing in the programming environment. To capture exploratory talk, it is important for us to distinguish these two types of *suggestions*. Finally, *coordination* is used when students monitor their progress involving positive or negative assessment. However, this is task oriented and does not involve being constructively critical about the partner's thoughts. Thus, it is considered as a code reflecting Cumulative talk. We used a turn of talk as the unit of coding. Turn of talk refers to *the systematic allocation of opportunities to*

Table 2 Modified mercer framework

Type of talk	Category (Visualization code)	Description	Example
Exploratory	Higher Order Question (HigherOrder)	Student questions to challenge their partner's ideas. These questions should be asking for reasoning	"Why did you move it?" "What happens if you keep it that way"
	Alternative Idea (AlternativeIdea)	Share an idea as an alternative to what the partner was suggesting or editing	"No, Let's change it to blue" "Okay, I clicked it, five seconds is too long, no seconds"
	Justification (Justification)	Student justifies their idea or change with reasons. Or evaluates a step as reasoning.	"It has to be the glide block 'cause it needs to move"
	Disagreement + Justification (Justification_Disagreement)	Disagreement with justification included.	"No, because we need to see the cat move slower"
Cumulative	Self-Explanation (SelfExplanation)	Student explains what they are doing or thinking while working on their own computer.	Uttering to self: "Okay, so move 10 steps." "I've to do the sailboat"
	Coordination (Coordination)	Monitoring task or group progress. Positive and negative assessment of task or group progress.	"You're always one step ahead. You're good." "Wait for me please" "Marshall! Focus!"
	Suggestion (Suggestion)	Student suggests or shares an idea for the next step. Also includes reminders.	"You have to change the number to 80."
	Simple Question (SimpleQuestion)	Question about a process or fact.	"Which one?" "Are you trying to change the number?"
Disputational	Disagreement-Justification (NoJustification_Disagreement)	Disagreement without inclusion of any justification.	"No" "That's wrong"

talk and the systematic regulation of the size of those opportunities (Schegloff, 1991). Each turn of talk was tagged such that any appearance of a code in that turn of talk was noted, and codes were not mutually exclusive. Surpassing the standard of dual coding 20% of the dataset (Syed & Nelson, 2015), two pairs of coders dual tagged approximately 46% of the video transcripts (N=24). We calculated kappa for each conversational category separately as the coding of each category was mutually exclusive. The average kappa for all the categories was 0.796 (min=.749, max=.857); the remaining transcripts were solo tagged by a single coder.

The second step in our analysis was a statistical one. Due to differences in programming session lengths (M=26.1, Min=18.7, Max=31.9 minutes), we calculated a normalized score of the total of tagged categories per minute for each dyad. We then used this rate of each category to explore differences in talk between the intervention and the control group.

The final step in our analysis was ENA. The coded dialogue from Step One was analyzed using ENA (Shaffer, 2017) and visualized using the Web Tool (Marquart et al., 2018). Performing ENA analysis begins with making decisions around the three basic ideas that generate an ENA network model: units, conversations, and codes.

First, we chose our units. Units are the portions of the data around which one would like to model connections. A network will be generated for each unique unit, but the power of ENA lies within its ability to combine networks of units, enabling one to make claims about larger groups within the study. To generate units, we selected 3 columns in our coded data in a hierarchical manner: Condition, Week, and Dyad. In this layout, Condition is at the top of the hierarchy, and thus the primary unit about which we wished to generate and analyze networks. This unit hierarchy allowed us to analyze, both visually and statistically, elements of the two conditions: Intervention and Control.

We then selected our conversations. Conversation parameters define which sections within the data ENA will make connections within. This parameter ensures that connections are not drawn between coded lines that do not hold meaning. For example, in our study, we would not want connections made between turns of talk by dyads that did not represent the same Condition. For this reason, we selected our conversation parameter as 'Condition.' Moreover, we used a moving stanza window with a window size of six. The moving stanza window is a method used to model the recent temporal context (Siebert-Evenstone et al., 2017). With this parameter setting, ENA uses a moving window in which a chosen number of turns of talk can be seen at once. If a co-occurrence of codes occurs within this window, it is counted. Then the window slides to include a new turn of talk and discards the turn of talk that has been within the window longest. This method naturally models the temporal connectedness of recently uttered dialogue while not attempting to connect dialogue stated much farther apart. For example, if a student within a dyad makes a statement that aligns with the code *suggestion* and then twenty minutes later, makes a statement that is coded as *justification*, instinctively, one would not think this student cognitively connected these two codes from these two turns of talk.

Within the network, each node represents a code. When ENA networks are generated, a dimension is created for every pairwise combination of the codes one selects. Even with a modest number of codes, the combinatorial math leads to a very large number of possible dimensions to visualize. For this reason, we used the Single Value Decomposition (SVD) optimization algorithm for dimensionality reduction and to rotate the model such that the dimension along the x-axis is the one that explains the greatest variance among the units, and the dimension along the y-axis explains the second greatest amount of variance; this process is similar to that which occurs with principal components analysis (Arastoopour Irgens et al., 2015).[1] With this in mind, it is important to pick codes with some thoughtfulness, as to not miss a potentially highly-variant dimension. For our analysis, we chose the codes listed in Table 2.

The analysis was structured based on the semantic scheme of the visualizations. First, the size of a node displays how prevalent the corresponding code was relative to all code co-occurrences (see Fig. 2 for varied sized nodes). Second, the thickness of the connecting lines between codes, in turn, indicates the relative frequency of the code pair counts as determined by the moving stanza window technique (see Fig. 3 for differences in thicknesses between

[1] See Shaffer et al., (2016) for a more detailed explanation of the mathematical models utilized in this analytic approach; see Arastoopour Irgens et al. (2015) and Sullivan et al. (2018) for examples of this kind of analysis

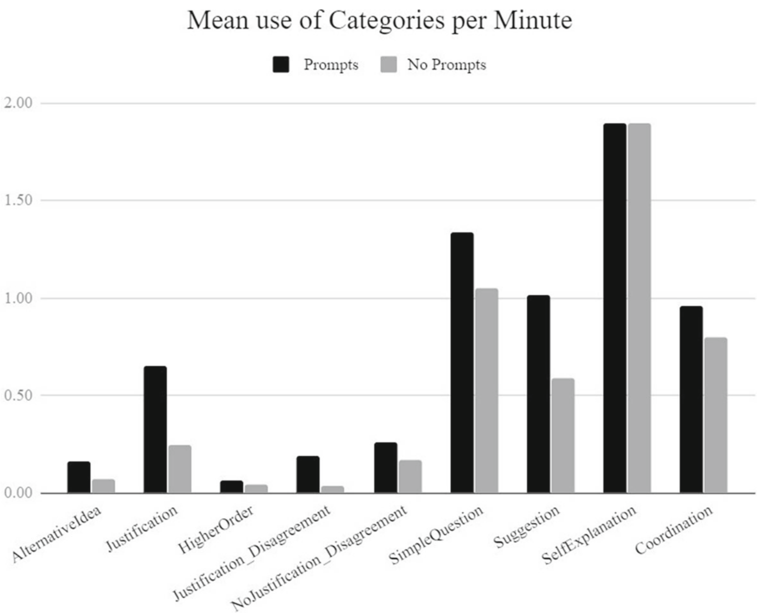

Fig. 2 Epistemic network analysis: Node size difference

nodes). Finally, the placement of the nodes in the space also holds meaning. Nodes that are near to each other in the space explain the variance among the units in a similar manner. Node placement uses an optimization routine that minimizes the overall distance of nodes to a centroid of all the nodes. Thus codes ending up in a similar region have a relationship based on their proximal location in Cartesian space (Shaffer et al., 2016). For example, see Fig. 4 for two *clusters* in which the left hand side of the space with the rectangle represents Exploratory talk and the right hand side of the space with the oval represents Cumulative talk.

Results

Statistical results

We first investigated the overall rates of students' types of talk. Figure 5 presents the distribution of the mean rate of conversation categories. For both groups, *self-explanation* was the most commonly coded talk type used during students' collaboration. *Simple question* was the next highest category. *Coordination* and *suggestion* were moderately used. However, exploratory talk categories—*justification*, sharing of *alternative ideas*, and *higher order questioning*—were minimally used by students. Campe et al. (2020) and Zakaria et al. (2019) found that students at this age tend to use less exploratory talk and more cumulative talk. As we were interested in exploring if prompting would help students increase their use of exploratory talk during their collaboration, ANOVAs were used to examine if there were differences in conversation categories between the intervention and control groups.

Repeated measures ANOVA on four data points per dyad showed no significant difference in overall mean rate of conversation between the two groups. A MANOVA test, with

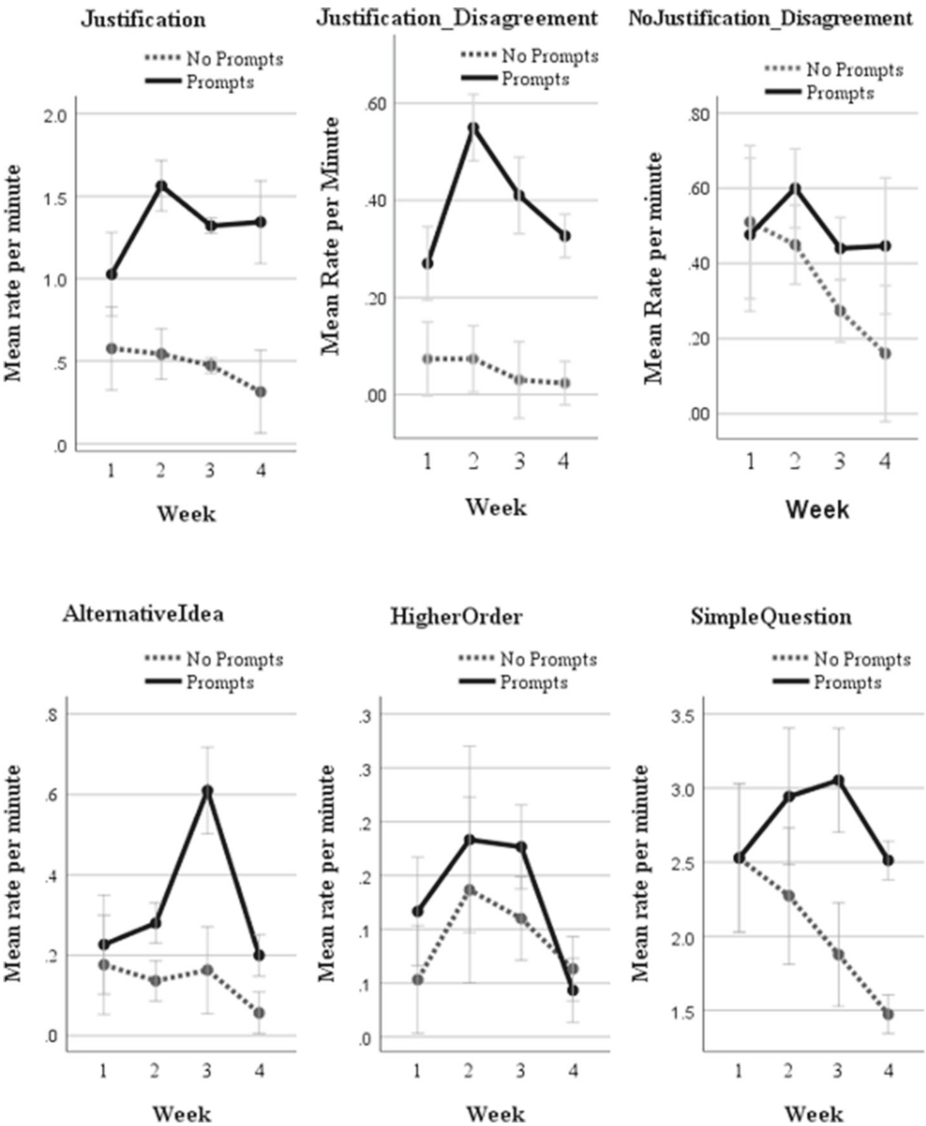

Fig. 3 Epistemic network analysis: Edge thickness

all the categories as DVs, showed a significant difference in the use of conversation categories between the groups (IV), F (14, 9) = 11.564, p < 0.001; Wilk's Λ = .114, $\eta\rho2$ = .881. To explore each category separately, we then examined the data with univariate ANOVAs Table 3.

Among the exploratory categories, *justification* F(1,22)= 38.647, p< 0.01, $\eta\rho2$= .637 and *alternative idea* F(1,22)= 6.877, p=.016, $\eta\rho2$= .238 showed significant differences between the intervention and control group where the intervention group was higher (see Fig. 6). Although a substantial part of the prompting and pre-activity instruction emphasized asking challenging questions (i.e., 'why' questions seeking justification) of the partner, both groups

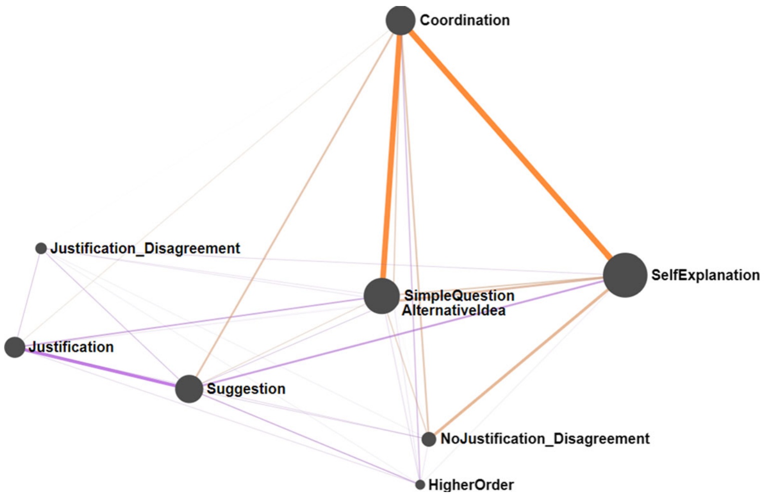

Fig. 4 Epistemic network analysis: Node placement

used a minimal number of *higher order questions* and did not show any differences. In contrast, the use of *simple questions* was significantly higher in the intervention group, F(1,22)= 6.988, p=.015, $\eta\rho 2$= .241.

ENA visualizations and descriptive analysis results

ENA revealed patterns of meaningful discourse not made obvious by the statistical results above. This is not surprising as ENA is predicated on highlighting the frequent connections among the types of talk student dyads made as they worked collaboratively.

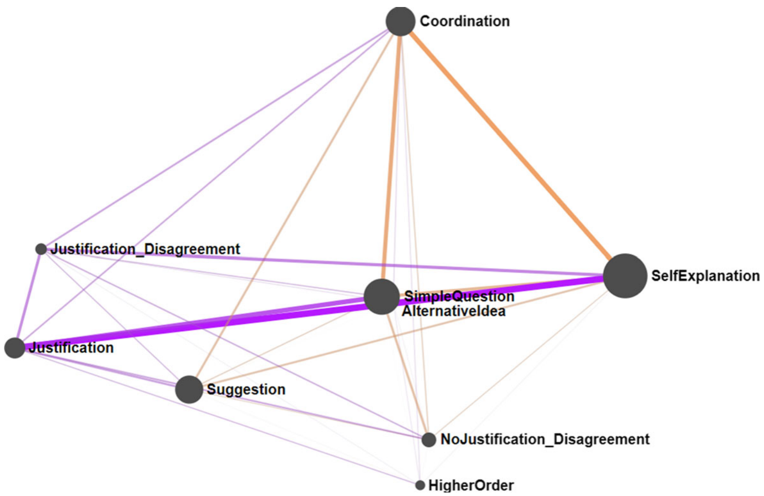

Fig. 5 Mean rate of conversation categories by experimental condition

Table 3 ANOVA results for each talk type category between groups

	Dependent Variable	F	*p*	Partial Eta Squared
Exploratory	HigherOrder	1.009	.326	.044
	AlternativeIdea	6.88	.016	.238
	Justification	38.647	.000***	.637
	Justification_Disagreement	43.281	.000***	.663
Cumulative	SelfExplanation	.606	.445	.027
	Coordination	2.700	.115	.109
	Suggestion	17.036	.000***	.436
	SimpleQuestion	6.988	.015**	.241
Disputational	NoJustification_Disagreement	1.931	.179	.081

For all of the following models (Figs. 7 through 10), orange edges, or lines, represent the control students and purple represent intervention students. Thicker, darker lines indicate more co-occurrences of the two codes the lines connect, whereas thinner, lighter lines indicate fewer co-occurrences of the codes. These models represent the difference between the control and intervention students, so where orange is dominant, that simply indicates that control students offered proportionally more co-occurrences of the noted codes; it does not mean that intervention students did not utter such statements. Moreover, for the excerpts provided, we include only the relevant codes given the co-occurrences under consideration. That is, one turn of talk may have multiple codes associated with it, but we only note the pertinent co-occurring codes in the following tables.

Further, along the x-axis, a Mann-Whitney test showed that Intervention students (Mdn=-1.17, N=12) were statistically significantly different at the alpha=0.05 level from Control students (Mdn=1.00, N=12 U=10.00, p=0.00, r=0.86). This significant difference highlights

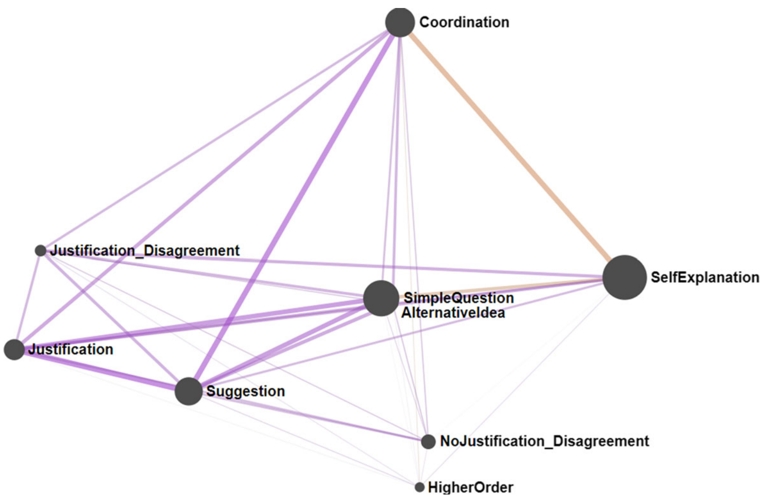

Fig. 6 Mean rate of exploratory categories and simple question

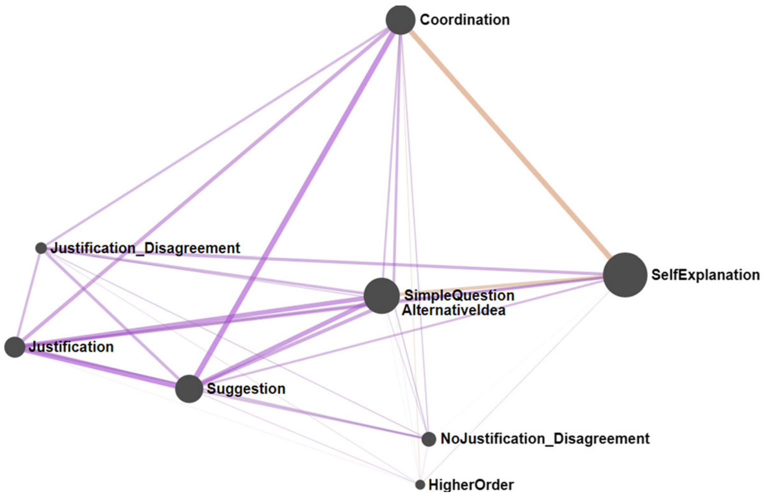

Fig. 7 Week 1 of Intervention. Purple lines/edges represent intervention students; orange lines/edges represent control students. Node size indicates how prevalent the corresponding code is relative to all code co-occurrences

the spread along the x-axis of such codes as those representing students' use of *justification*—on the left hand side of the space—or *self-explanation*—on the right hand side of the space, and the difference in patterns of their occurrence between the two groups. Recall that *justification* is an Exploratory type of talk, whereas *self-explanation* is a Cumulative type of talk. Despite having a small sample size, this robust nonparametic test provides further support for our use of ENA.

Exploratory-coded results

Justification and simple question In the intervention group, *justification*—an Exploratory category—co-occurs with most of the other Exploratory and Cumulative codes. Rather than detail each co-occurrence, we will highlight illuminating examples, those that seemingly do not belong together or which tell a distinctive story. Of particular interest is the co-occurrence of *justification* and *simple question* (see Figs. 8 though 10). *Simple questions* generally do not require a detailed and reasoned response and yet the intervention students engaged in more elaborate explanations for their decisions when asked process-oriented or fact-based questions. Recall that there was a statistically significant difference between intervention and control students in their use of *justification* and *simple questions* (see Fig. 6). An excerpt from the intervention students is provided in Table 4.

This exchange details how the act of finding the perfect background and the *simple question* of "what about…?" launched the dyad into justifying their suggestions. A possible explanation for many of these co-occurrences could be that intervention students emphasized providing *justification* to the extent that even for *simple questions* they diligently came up with justifying answers. Moreover, the students justified their suggestions many turns in a row, which indicates a more meaningful exchange than having many turns of agreement or disagreement without any *justification*.

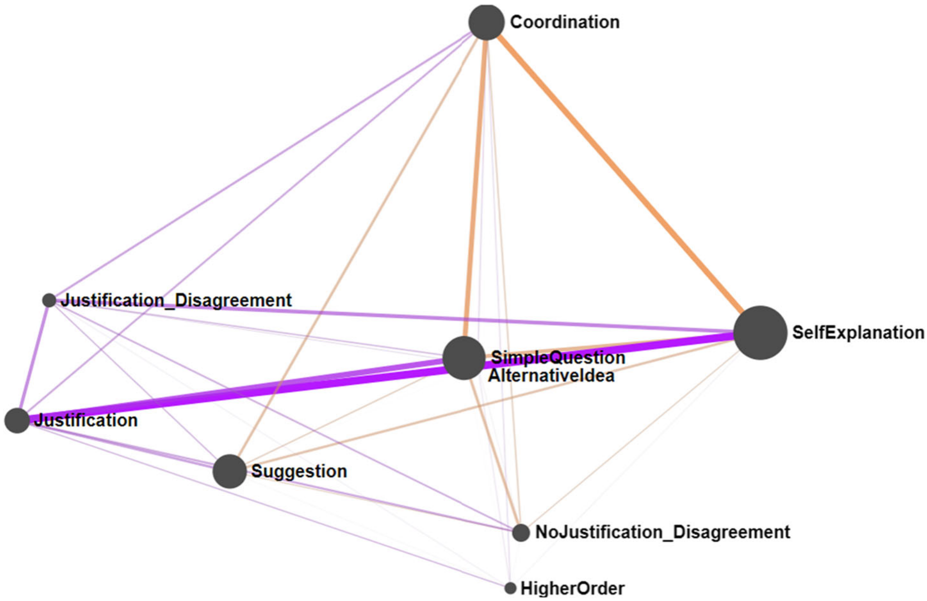

Fig. 8 Week 2 of Intervention. Purple lines/edges represent intervention students; orange lines/edges represent control students. Node size indicates how prevalent the corresponding code is relative to all code co-occurrences

Justification and coordination Similar to the co-occurrence pattern noted above, intervention students likewise paired *justification* and *coordination* statements across the intervention (see Figs. 9 and10). This is noteworthy largely given the steady increase in these co-occurring statements, as seen in the increasingly darker and thicker purple line over the four weeks. The intervention students seemed to provide extra reasoning whilst ensuring the efficient execution of group tasks. One such excerpt appears in Table 5.

This dyads' exchange highlights their coordinating efforts—seen here as attempts to ensure their code and their process are the same—interspersed with *justification* for their

Table 4 Intervention group: Justification and simple question excerpt

Student	Turn of Talk	Co-occurring Code(s)
1	Or I feel like we should do something more, like not blue sky?	——
2	Yeah, what about….?	Simple Question
1	It's too plain.	Justification
2	It's more for love. I know it's Valentine's day coming up but, no. Stripes.	Justification
1	No. Because that would look too busy-	Justification
2	But it would be confusing and stuff and that's what it's supposed to be. It's supposed to be hard. How about stripes, or? Nah.	Justification

—— Indicates this turn of talk was not tagged with either Simple Question nor Justification

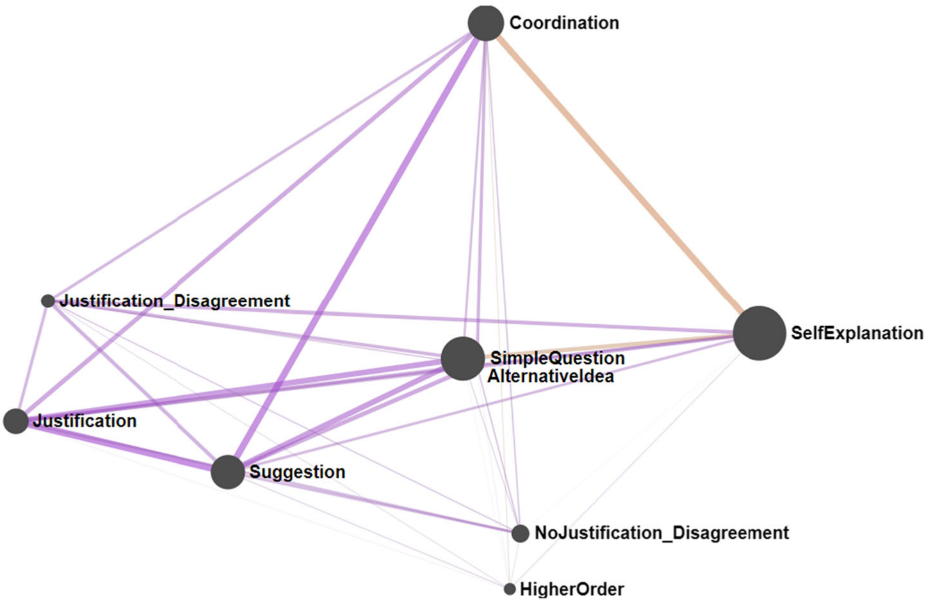

Fig. 9 Week 3 of Intervention. Purple lines/edges represent intervention students; orange lines/edges represent control students. Node size indicates how prevalent the corresponding code is relative to all code co-occurrences

individual actions. It is important to reiterate that, under this pair programming setting, students did not share the same virtual computer workspace, therefore they may have felt the need to coordinate more to create the same product together. The finding that intervention students show many co-occurrences of *justification* with *coordination* is expected as justifying ideas, actions or suggestions were prompted by the research.

Justification with disagreement and self explanation Intervention students also provided reasoning within their disagreement statements, coded here as *justification_disagreement*, and although there was not a statistically significant difference in the use of this code by the two groups of students (see Fig. 6), intervention students used this type of talk more readily. These types of utterances co-occurred with *self-explanation* more so than with the control students (see Figs. 8 through 10). This is an intriguing finding because utterances coded as *self-explanation* are those in which students are explaining what they are doing or thinking while working on their own computer. These types of utterances rarely require input from a partner, especially in this CSCL configuration where each student has their own device, let alone detailed and reasoned responses. Yet, the intervention students offered more of these paired statements across all four weeks. We are unable to say definitively if these self-explanations are intended solely for the student who utters them or for both partners. One example appears in Table 6.

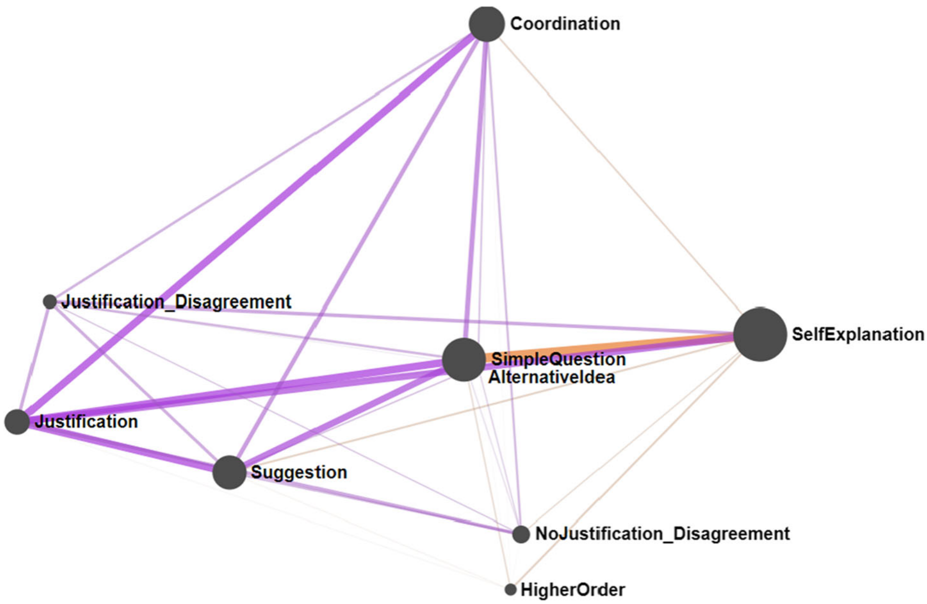

Fig. 10 Week 4 of Intervention. Purple lines/edges represent intervention students; orange lines/edges represent control students. Node size indicates how prevalent the corresponding code is relative to all code co-occurrences

This excerpt underscores the ways students receive *self-explanation*, generally observed as self talk with no overt attempt to engage the partner, as passive input. In this case, the students intermittently spoke their thinking aloud and voiced their actions, with the partner hearing and responding to the utterance negatively but with justification. What we are unable to delineate is when students self-explain as a way of simply verbalizing to themselves and when they self-explain in hopes of having their partner respond. Absent any traditional conversational markers such as asking a question, saying the partner's name, or

Table 5 Intervention group: Justification and coordination excerpt

Student	Turn of Talk	Co-occurring Code(s)
1	No sing! [Laughs]	——
2	It's on the stage...	Justification
1	No, I'm going to make it size twenty-five, so then it's like actually..	Justification
2	Oh my goodness, oh my goodness!	——
1	[Student], what did you put?	Coordination
2	I put three twenty-five. [Laughter] Now we have to make it dance.	Coordination
1	All right, so... look.	Coordination
2	Oh, my goodness... I think I wanted it to sing, that's what I wanted it to do. Yep, sing, then change color, then hide.	——
1	No, let's do a dance, 'cause it's on a stage.	Justification

—— Indicates this turn of talk was not tagged with either Coordination nor Justification

Table 6 Intervention group: Justification with disagreement and self-explanation excerpt

Student	Turn of Talk	Co-occurring Code(s)
1	Okay. So it says when clicked, start sound growl, wait 30 seconds or for 30 seconds.	Self-Explanation
2	Wait 30 seconds	Self-Explanation
1	Because this goes. . . .	—
2	No don't do this for p. How do you delete it? Delete block. And then, we have to go this page. Let's just do it with roar because then it will be easy. Let me find, wait.	Self-Explanation, Justification with Disagreement
1	Wait until. . . . Yeah, wait until in control.	Self-Explanation
2	Wait until...no don't do wait until because. Just press wait 30 seconds.	Justification with Disagreement
1	If wait 30 seconds, it gets here?	Self-Explanation

— Indicates this turn of talk was not tagged with either Justification with Disagreement nor Self-Explanation

looking at the partner, we feel we can only discuss the quality of the dyad's collaboration when the partner acknowledges the *self-explanation*.

Cumulative-coded results

Coordination and self explanation Control students offered more *self-explanation* and *coordination* utterances, although these co-occurring statements decrease across the four week intervention. Figures 4 through 7 clearly illustrate this change. The week 1 model (Fig. 7) conveys this through the thick orange line between *coordination* and *self-explanation*. This line thins and lightens as the intervention continues. By examining the students' co-occurring utterances, coded as *self-explanation* and *coordination*, we are better able to surmise both why the control students offered these utterances and why they decreased over time. One excerpt from a dyad during week 1 of the intervention appears in Table 7.

This exchange among two control students highlights their use of *self-explanation*—generally conceived of as self-talk or narrating one's actions or thoughts—and *coordination*—comments aimed at the organization of task or group process. This excerpt begins with Student 1 commenting on the fish sprite on her partner's screen. Student 2 then narrates her own actions for a different sprite. Student 1, while looking at her own screen comments on the fish sprite, then looks at her partner's screen and intervenes to prevent her from deleting the entire project. Both students then turn their attention to their own screens and verbalize their actions, with Student 1 asking for technical task assistance from her partner in the final line.

We believe that without the strategies provided in the prompts, the control students defaulted to verbalizing their individual work and superficially checking in with their partner. As the intervention continued, control students' mean rate of talk across all talk types dropped (Fig. 6). It appears as though, minus the prompts, the control students suffered from a lack of drive to talk to one another.

Table 7 Control group: Coordination and self-explanation excerpt

Student	Turn of Talk	Co-occurring Code(s)
1	That fish is so cute.	—–
2	And then we're going to go to the costume.	Self-Explanation
1	Look at the fish. Oops. What are you doing? No, no, no, don't do it. Don't do anything. Don't do anything. Don't do anything. No, then you're going to erase the whole thing. Okay good.	Coordination
2	Okay, I don't want any...	Self-Explanation
1	And the apple. Yes, I need the apple here. You go to costume...	Self-Explanation
2	Oh, there's two fishbowls. We had a fish in-	—–
1	How'd you make the bite out of the apple here?	Coordination

—– Indicates this turn of talk was not tagged with either Coordination nor Self-Explanation

Coordination and simple question Quantitative results indicated that control students uttered significantly fewer *simple question*-coded statements than the intervention students (see Fig. 6). However, ENA results showed that across the first two weeks of the intervention (Figs. 7 and 8), control students offered more co-occurrences of *simple questions* and *coordination* utterances than the intervention students. In the final two weeks (Figs. 9 and 10), the intervention students offered more. That both control students paired *simple questions* with *coordination* statements and that a reversal occurred in which group made more *simple question* and *coordination* co-occurring statements is best explored through students' spoken words. An excerpt from a control dyad appears in Table 8.

This dyad's exchange demonstrates several *simple questions* that were also coded as *coordination*. The students' line of questioning served simultaneously to organize their programming work and to align their task progress. Compare the Control Group (Table 8) to the following intervention dyad (Table 9) who also had several dual coded simple question and coordination turns of talk.

The difference in how the control and intervention students expressed *simple question* and *coordination* utterances is a qualitative one, but may also reflect the intervention students' repeated exposure to strategy prompts. More specifically, the intervention students

Table 8 Control group: Coordination and simple question excerpt

Student	Turn of Talk	Co-occurring Code(s)
1	Can I choose the background?	Simple Question, Coordination
2	Wait one second.	Coordination
1	Oh I got one.	—–
2	Wait.	Coordination
1	Is this one good?	Simple Question
2	Yeah, perfect. Wait, let me just get … Wait, what if I search up camping?	Simple Question, Coordination

—– Indicates this turn of talk was not tagged with either Coordination nor Simple Question

Table 9 Intervention group: Coordination and simple question excerpt

Student	Turn of talk	Co-occurring Code(s)
1	Wait 10 seconds and then change color? To what?	Simple Question, Coordination
2	Color effect 25. Wait, or do you think we should do...	Simple Question, Coordination
1	Wait.	Coordination
2	Look. They have all these different ones, look.	Coordination
1	Wait, what? But we want to change color, I thought.	Simple Question, Coordination
2	Well, I think change color...	——
1	What do you think?	Simple Question, Coordination
2	It won't change color. How do you make it all go? And then it will hide. When this sprite clicked... Okay. Show. Okay, do you want to start hiding the animals?	Simple Question, Coordination

—— Indicates this turn of talk was not tagged with either Coordination nor Simple Question

asked for their partner's opinion, for clarification, and for different ideas from the partner, all of which promote the three characteristics of exploratory talk mentioned in the prompts. Control students, however, asked more polar questions (i.e., seeking yes or no answer).

Disputational-coded results

No justification disagreement and self explanation Control students offered more co-occurring *self-explanation* and *nojustification_disagreement* statements in three of the four weeks, with the strongest appearance of these codes occurring in week 1 (Fig. 7). An excerpt appears in Table 10.

In this excerpt, Student 1 mostly engages in self talk; she is looking at her own screen where she is searching for a specific background option. She does not directly respond to her partner's question of a second choice, but when her partner describes the horse sprite she has on her own screen, Student 1 responds negatively without providing any reasoning. This matters in terms of what we expect from students. Ideally, students would elaborate on their ideas with *justification*; the partner needs to positively welcome different ideas and the

Table 10 Control group: No justification disagreement and self-explanation excerpt

Student	Turn of Talk	Co-occurring Code(s)
1	Just search forest. That's what it is.	Self-Explanation
2	Yeah it doesn't have a stable. There's no stable on this. So what would be the second choice?	No Justification Disagreement
1	The forest, it's called forest. It's really pretty, there's a bunch of...	Self-Explanation
2	A horse galloping on the beach.	Self-Explanation
1	No.	No Justification Disagreement

dyad would discuss them. Here, however, the students arbitrarily provide options without further discussion and reject their partners' suggestions without saying why.

Discussion and conclusion

The ENA visualizations, in conjunction with descriptive analysis alongside statistical analysis of the ENA quantitative models, have permitted us to build upon well established coding frameworks currently in use by the CSCL research community. ENA provides additional insight to historical methods of utilizing these coding frameworks. ENA's model combines structured comparisons of the prevalence of codes, their relationship to other codes, and co-occurrences of codes in windows of time. The utility of each of these pieces of information, individually and collectively, has been demonstrated in the analysis above.

We start our discussion of the results by focusing on what the two groups had in common. Each dyad's aim was to accomplish a programming project with an identical outcome while working on their own screens. As such, this external instructional constraint may be the reason why many of the code co-occurrences in both groups included *self-explanation*; they simply intended to keep one another appraised of their progress. Moreover, both groups frequently used *simple questions*, despite the intervention students receiving prompts encouraging them to use *higher order questions*. Research has shown that young students struggle to participate in argumentative collaboration (Bell, 2004). It could be that although the intervention students were prompted to ask one another challenging questions, they were unable to translate that request into the target *higher order questions* types. From an instructional standpoint, students likely need more practice developing their ability to ask and answer challenging questions. To sustain students' practices to ask such a question, more explicit instruction and dialogue modeling ought to be provided to students.

Although *simple questions* are more frequent than *higher order questions*, there are qualitative differences in the content of the *simple questions* being asked by the two groups; ENA's descriptive evidence outlined in the results helped highlight these qualitative differences. The intervention students' *simple questions* typically served to align the dyad's work (ie., "Do you think we should do one?" or "What about this?"), whereas the control students' *simple questions* generally sought a yes or no answer (ie., "Do you like this background?" or "Is this okay?"). The former type of *simple questions*, those more readily used by intervention students and evident in the ENA models, were likely to result in further collaborative talk.

In the absence of prompts, the control groups' discourse is qualitatively different, as noted in the ENA findings above. The control students utilized talk types *self-explanation*, *coordination*, and *nojustification_disagreement*) that did not engender elaborative conversations with their partner, thereby likely not enhancing the quality of their collaboration. The intervention students, however, more frequently asked *questions* of each other, proposed *alternative ideas*, and *justified* their thinking. We acknowledge that codes and co-occurrences appear in both intervention and control students' networks; it is the quality of the dyads' use of those codes that makes them different.

The use of *justification* statements, independently or with disagreement, appears to be a distinctive difference between the two groups—namely, as the results detail, that intervention students used *justification* more so than control students and in ways that were different from control students. Regarding *justification* and *coordination* co-occurrences, we posit the following. *Coordination* is difficult in collaborative activities, especially in this 2CPP configuration where the students have to look at their partners' screen to keep track of what is happening. They have to regularly verbally communicate to remain *coordinated*. However, it is not just enough to *coordinate* their actions, because it is likely that each student has their own idea of where they would like their project to go; therefore, they must also verbally negotiate the changes that should be made. The *justification* action, by showing responsiveness to the partner, indicates that both of the students are participating and have equal say in what their project will look like. Intervention students had higher frequency of the *justification* and *coordination* co-occurrence than control students. Moreover, ENA indicated that the students differed in how they *coordinated* their efforts, and this, we maintain, was likely supported by the prompting the intervention students received.

A further distinction between the intervention and control students is their use of *justification_disagreement* and *self-explanation* statements. Although the majority of the *self-explanation* utterances appear to be of students thinking aloud and not necessarily speaking to their partner, their partners still heard them and voiced their opinions. Instead of simply ignoring, agreeing, or disagreeing with the utterances, the intervention partners often *disagreed* and *justified* their thoughts. Previous work by Barron (2003) found that young students' collaborative processes were often more productive when the partners accepted suggestions or took the time to discuss them rather than rejecting or ignoring them. This is especially important because the *justification* made by the intervention students aligned with Barron's (2003) finding in that they were providing more of their thoughts to their partner. These thoughts also allowed for their partner to further engage in a coordination-centered conversation regarding the direction they wanted their projects to head. We assert that the prompting intervention students received likely increased their capacity to ask questions, challenge, and verbalize their thoughts, thereby engendering differences in the two ENA models.

We posit that prompting has built a higher level collaboration between the partners and has infused in them a set of responsibilities regarding not only what they need to do with the task but also how they need to talk to each other. For them, the intervention students, *self-explanation* was used as a think-aloud for problem solving that resulted in raising the quality of communication and problem-solving of the intervention group relative to the control group. The quality of team communication encouraged problem solving and collaboration among team members (Tseng et al., 2009). In the control group, there was not as high a level of collaboration, minimal determined task and talk responsibilities, and therefore less of a likelihood of forming elements of a shared mental model. As such, the use of *self-explanation* co-occurring with *coordination* helped keep the intervention students together and focused on the shared task. Moreover, this heightened collaboration may have evoked the belief within the intervention dyads that they could safely ask questions of, and challenge, one another. In fact, Frisby and Martin (2010) found that student rapport—which they operationalize as a positive connection and interaction—engenders quality peer participation in classroom activities.

Mercer et al. (1999) maintained that exploratory talk—marked by instances of critical challenge, but which occur in task-beneficial ways—is the most beneficial of the talk types and that students are capable of learning how to talk in ways that support such productive discourse. A simple intervention such as the one executed here is one way to support students' acquisition of these types of talking skills. Analyzing these interactions is complex, though. Discourse is dynamic and robust analysis requires both a framework that captures nuances of students' talk, but also a methodology that can express the meaningful relationships between utterances. ENA was found to be very capable of meeting these requirements.

In particular, ENA was able to uncover patterns of meaningful discourse that the quantitative data based on simple code counts did not reveal. For example, in the quantitative analysis it was found that the frequency of *simple question* was higher in intervention group but did not provide further details on how that utterance was used. ENA results showed that it often co-occurred with *coordination* in ways that may indicate the intervention students had a heightened desire to incorporate their partner's input.

Our findings provide notable implications for system design and practice. Our results indicate that even relatively simple prompting by teachers was effective at elevating the level of student discourse. In particular, we found that intervention students used collaboratively beneficial *justification*-type talk more so than control students. Moreover, intervention students asked more *simple questions* than control students. Findings such as these could be used with a real-time adaptive system such as that developed by Adamson et al. (2014) that prompted students to articulate their thinking to peers. With the ability to analyze students' discourse in real-time with feature detectors, the system might prompt students to explain why they disagree, to provide reasoning for their actions, and perhaps prompt students to talk when they are exceedingly silent or neither student have utilized the laptop for some time. Another possible route to take is to use other system-based prompts that would encourage students to use justification or simple questions. One example could be to prompt them every 10 minutes, as was done in our work. Yet another approach could be to analyze students' code in real time (Marwan et al., 2020; Price et al., 2017), and prompt them when they have completed a task. Both options are desirable for different reasons. The first may have them justify and ask questions while they are working and may help them work towards a consensus. The second may help them reflect on their work.

Not only does this work potentially inform adaptive systems that would help support students directly and thus lessen the burden of teachers, but these systems can also provide information directly to teachers to inform their practice. More specifically, teachers could make use of these data by discerning the discursive needs of their students; knowing which students use which types of talk and perhaps that some students rarely use beneficial types of talk may help inform future lessons. Such lessons, based on our findings here, might include practices around academic argumentation and consensus building. These are important because asking challenging *higher order questions* and proposing *alternative ideas* is difficult for many students, and 'groupthink' can be problematic as groups make decisions (Baron, 2005).

With all studies, this one experienced limitations. First, we limited our sample size based on clarity of audio and video due to the time-intensive nature of qualitative analysis. However, future work should expand the sample size to increase the diversity of networks we and others can explore in ENA. Second, we did not assess changes in students' collaboration, nor their attitudes toward it, using an established instrument that might show between- and/or within-subjects differences as a result of the intervention. Third, as this study does not include data involving any other type of pair programming configuration, we are unable to make statements about the interaction of this unique 2C pair programming configuration and prompting on students' collaboration. Future work should explore different pair programming configurations and prompting. Fourth, out of concern that students may have viewed our prompting as intrusive, future efforts ought to prioritize seamless distribution of prompts.

Additional research should focus on collaboration-building between student programmers in hopes of determining how collaboration and the development of mutual understanding may contribute to academic risk-taking, decision-making, and student outcomes.

As noted above and in established research, when students critically challenge one another their exploratory talk drives their cognition (Colfer, 2011). It will be important that future research investigates not only effective ways to integrate exploratory talk in children's learning experiences but how their discourse shows markers of exploratory talk. By looking for these markers, researchers and practitioners can know better how to support the prompting of such constructive talk. ENA could be more widely explored as a tool for this type of CSCL work. One of the affordances of ENA is in the ability to create and compare different epistemic models, thereby permitting quantitative analysis of and direct visual comparisons between individual students and dyadic verbalizations or, as we did here, between different experimental conditions. Future work may harness the efforts around automated talk analysis (Paulus & Wise, 2019) with ENA to produce easy to read and use graphics that could translate into actionable information for researchers and practitioners alike.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant No DRL-1721160. Additionally, this work was funded in part by the National Science Foundation (DRL-1661036, DRL-1713110), the Wisconsin Alumni Research Foundation, and the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin-Madison. The opinions, findings, and conclusions do not reflect the views of the funding agencies, cooperating institutions, or other individuals.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

Adamson, D., Dyke, G., Jang, H., & Rosé, C.P. (2014). Towards an agile approach to adapting dynamic collaboration support to student needs. *International Journal of Artificial Intelligence in Education*, *24*(1), 92–124.

Arastoopour, G., Shaffer, D. W., Swiecki, Z., Ruis, A., & Chesler, N.C. (2016). Teaching and assessing engineering design thinking with virtual internships and epistemic network analysis. *International Journal of Engineering Education*, *32*(3), 1492–1501.

Arastoopour Irgens, G., Shaffer, D. W., Swiecki, Z., Ruis, A., & Chesler, N.C. (2015). Teaching and assessing engineering design thinking with virtual internships and epistemic network analysis. *International Journal of Engineering Education*.

Asterhan, C. S., Schwarz, B. B., & Cohen-Eliyahu, N. (2014). Outcome feedback during collaborative learning: Contingencies between feedback and dyad composition. *Learning and Instruction*, *34*, 1–10.

Baron, R. S. (2005). So right it's wrong: Groupthink and the ubiquitous nature of polarized group decision making. *Advances in Experimental Social Psychology*.

Barron, B. (2003). When smart groups fail. *The Journal of the Learning Sciences*, *12*(3), 307–359.

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.

Branaghan, R. J. (1990). Pathfinder networks and multidimensional spaces: Relative strengths in representing strong associates. In *Ablex series in computational sciences. Pathfinder associative networks. Studies in knowledge organization* (pp. 111–120). Ablex Publishing.

Bressler, D. M., Bodzin, A. M., Eagan, B., & Tabatabai, S. (2019). Using epistemic network analysis to examine discourse and scientific practice during a collaborative game. *Journal of Science Education and Technology*, *28*(5), 553–566.

Bungum, B., Bøe, M. V., & Henriksen, E.K. (2018). Quantum talk: How small-group discussions may enhance students' understanding in quantum physics. *Science Education*, *102*(4), 856–877.

Campe, S., Denner, J., Green, E., & Torres, D. (2020). Pair programming in middle school: variations in interactions and behaviors. *Computer Science Education*, *30*(1), 22–46.

Colfer, C. (2011). What types of talk are boys and girls using when engaged in a collaborative design and technology task? PATT 25: CRIPT8 pp 113.
Cress, U., Stahl, G., Ludvigsen, S., & Law, N. (2015). The core features of cscl: Social situation, collaborative knowledge processes and their design. *International Journal of Computer-Supported Collaborative Learning*, *10*(2), 109–116.
Creswell, J. W., & Clark, V. L. P. (2017). *Designing and conducting mixed methods research*. Los Angeles: Sage Publications.
Davidson, K., Larzon, L., & Ljunggren, K. (2010). Self-efficacy in programming among sts students. *Retrieved August*, *12*, 2013.
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.
Fernández, M., Wegerif, R., Mercer, N., & Rojas-Drummond, S. (2001). Re-conceptualizing "scaffolding" and the zone of proximal development in the context of symmetrical collaborative learning. *The Journal of Classroom Interaction* pp 40–54.
Fisher, E. (1993). Distinctive features of pupil-pupil classroom talk and their relationship to learning: How discursive exploration might be encouraged. *Language and Education*, *7*(4), 239–257.
Frisby, B. N., & Martin, M. M. (2010). Instructor–student and student–student rapport in the classroom. *Communication Education*, *59*(2), 146–164.
Goldsmith, T. E., Johnson, P. J., & Acton, W.H. (1991). Assessing structural knowledge. *Journal of Educational Psychology*, *83*(1), 88.
Gweon, G., Rose, C., Carey, R., & Zaiss, Z. (2006). 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).
Hardy, C., Harley, B., & Phillips, N. (2004). Discourse analysis and content analysis: Two solitudes. *Qualitative Methods*, *2*(1), 19–22.
Hennessy, S., Rojas-Drummond, S., Higham, R., Márquez, A. M., Maine, F., Ríos, R. M., García-Carrión, R., Torreblanca, O., & Barrera, M.J. (2016). Developing a coding scheme for analysing classroom dialogue across educational contexts. *Learning Culture and Social Interaction*, *9*, 16–44.
Knight, S., Arastoopour, G., Williamson Shaffer, D., Buckingham Shum, S., & Littleton, K. (2014). Epistemic networks for epistemic commitments. In *International conference of the learning sciences*.
Maguire, P., Maguire, R., Hyland, P., & Marshall, P. (2014). Enhancing collaborative learning using paired-programming: Who benefits? *AISHE-J: The All Ireland Journal of Teaching and Learning in Higher Education*, *6*(2), 1411–14125.
Marquart, C., Hinojosa, C., Swiecki, Z., Eagan, B., & Shaffer, D. (2018). Epistemic network analysis (version 1.7.0)[software].
Marwan, S., Gao, G., Fisk, S., Price, T. W., & Barnes, T. (2020). Adaptive immediate feedback can improve novice programming engagement and intention to persist in computer science. In *Proceedings of the 2020 ACM conference on international computing education research* (pp. 194–203).
McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, *49*(8), 90–95.
Mendes, E., Al-Fakhri, L. B., & Luxton-Reilly, A. (2005). Investigating pair-programming in a 2nd-year software development and design computer science course. In *Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education* (pp. 296–300).
Mercer, N. (2000). Words and minds–how we use words to think together.
Mercer, N. (2002). *Words and minds: How we use language to think together*. Evanston: Routledge.
Mercer, N. (2007). Sociocultural discourse analysis: Analysing classroom talk as a social mode of thinking. *Journal of Applied Linguistics and Professional Practice*, *1*(2), 137–168.
Mercer, N., & Littleton, K. (2007). *Dialogue and the development of children's thinking: A sociocultural approach*. Evanston: Routledge.
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.
Miyake, N. (1997). Making internal processes external for constructive collaboration. In *Proceedings second international conference on cognitive technology humanizing the information age* (pp. 119–123). IEEE.
Nikolaidou, G. N. (2012). Complus model: A new insight in pupils' collaborative talk, actions and balance during a computer-mediated music task. *Computers & Education*, *58*(2), 740–765.
Paulus, T. M., & Wise, A. F. (2019). *Looking for insight, transformation, and learning in online talk*. Evanston: Routledge.
Polo, C., Lund, K., Plantin, C., & Niccolai, G.P. (2016). Group emotions: The social and cognitive functions of emotions in argumentation. *International Journal of Computer-Supported Collaborative Learning*, *11*(2), 123–156.

Price, T. W., Dong, Y., & Lipovac, D. (2017). isnap: towards intelligent tutoring in novice programming environments. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education* (pp. 483–488).

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & et al (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60–67.

Rojas-Drummond, S., & Mercer, N. (2003). Scaffolding the development of effective collaboration and learning. *International Journal of Educational Research*, *39*(1-2), 99–111.

Rojas-Drummond, S., & Zapata, M. P. (2004). Exploratory talk, argumentation and reasoning in mexican primary school children. *Language and Education*, *18*(6), 539–557.

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

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 computing science education* (pp. 90–95).

Schegloff, E. A. (1991). Conversation analysis and socially shared cognition. In *Socially shared cognition*. Washington: American Psychological Association.

Shaffer, D. W. (2006). Epistemic frames for epistemic games. *Computers & Education*, *46*(3), 223–234.

Shaffer, D. W. (2017). *Quantitative ethnography*. Madison, Wisconsin: Cathcart Press.

Shaffer, D. W., Hatfield, D., Svarovsky, G. N., Nash, P., Nulty, A., Bagley, E., Frank, K., Rupp, A. A., & Mislevy, R. (2009). Epistemic network analysis: A prototype for 21st-century assessment of learning. *International Journal of Learning and Media 1*(2).

Shaffer, D. W., Collier, W., & Ruis, A.R. (2016). A tutorial on epistemic network analysis: Analyzing the structure of connections in cognitive, social, and interaction data. *Journal of Learning Analytics*, *3*(3), 9–45.

Shawky, D., Badawi, A., Said, T., & Hozayin, R. (2014). Affordances of computer-supported collaborative learning platforms: a systematic review. In *2014 international conference on interactive collaborative learning (ICL)* (pp. 633–651). IEEE.

Siebert-Evenstone, A. L., Irgens, G. A., Collier, W., Swiecki, Z., Ruis, A. R., & Shaffer, D.W. (2017). In search of conversational grain size: Modelling semantic structure using moving stanza windows. *Journal of Learning Analytics*, *4*(3), 123–139.

Sullivan, S., Warner-Hillard, C., Eagan, B., Thompson, R. J., Ruis, A. R., Haines, K., & Jung, H. S (2018). Using epistemic network analysis to identify targets for educational interventions in trauma team communication. *Surgery*, *163*(4), 938–943.

Syed, M., & Nelson, S. C. (2015). Guidelines for establishing reliability when coding narrative data. *Emerging Adulthood*, *3*(6), 375–387.

Teasley, S. D. (1995). The role of talk in children's peer collaborations. *Developmental Psychology*, *31*(2), 207.

Tsan, J., Vandenberg, J., Zakaria, Z., Wiggins, J. B., Webber, A. R., Bradbury, A., Lynch, C., Wiebe, E., & Boyer, K.E. (2020). A comparison of two pair programming configurations for upper elementary students. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 346–352).

T'Sas, J. (2018). Learning outcomes of exploratory talk in collaborative activities. Unpublished PhD dissertation, university of Antwerp.

Tseng, H., Ku, H. Y., Wang, C. H., & Sun, L. (2009). Key factors in online collaboration and their relationship to teamwork satisfaction. *Quarterly Review of Distance Education 10*(2).

Van Eaton, G., Clark, D. B., & Smith, B.E. (2015). Patterns of physics reasoning in face-to-face and online forum collaboration around a digital game. *International Journal of Education in Mathematics Science and Technology*, *3*(1), 1–13.

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.

Wegerif, R. (2005). Reason and creativity in classroom dialogues. *Language and Education*, *19*(3), 223–237.

Xun, G., & Land, S. M. (2004). A conceptual framework for scaffolding iii-structured problem-solving processes using question prompts and peer interactions. *Educational Technology Research and Development*, *52*(2), 5–22.

Zakaria, Z., Boulden, D., Vandenberg, J., Tsan, J., Lynch, C., Wiebe, E., & Boyer, K. (2019). Collaborative talk across two pair-programming configurations. In *A wide lens: Combining embodied, enactive extended, and embedded learning in collaborative settings, 13th international conference on computer supported collaborative learning (CSCL)*, (Vol. 1 p. 2019).

Zakaria, Z., Vandenberg, J., Tsan, J., Boulden, D. C., Lynch, C. F., Boyer, K. E., & Wiebe, E.N. (2021). Two-computer pair programming: Exploring a feedback intervention to improve collaborative talk in elementary students. *Computer Science Education*, pp. 1–28.

Zhang, S., Liu, Q., & Cai, Z. (2019). Exploring primary school teachers' technological pedagogical content knowledge (tpack) in online collaborative discourse: An epistemic network analysis. *British Journal of Educational Technology*, *50*(6), 3437–3455.