Validating an Observation Protocol for Structured Roles in Cooperative Learning

Morgan M. Fong, Liia Butler, Hongxuan Chen, Geoffrey L. Herman
University of Illinois, Urbana-Champaign
{mmfong2, liiamb2, hc10, glherman}@illinois.edu

***Abstract*—This Research Full Paper presents an observation protocol to explore group processing in cooperative learning. Use of structured roles, such as Process Oriented Guided Inquiry Learning (POGIL) and pair programming, can help facilitate cooperative learning and help courses scale to large classroom sizes, decrease attrition and failure rates, and improve student performance. Our observation protocol was created to capture how groups work together in online, POGIL-inspired activities. A team of graduate student researchers developed the observation protocol for a variety of courses by observing three different computer science courses during the Spring 2021 semester. A total of 77 groups across all three courses were recorded, and percent agreement using a subset of the recordings suggested good interrater reliability (91.29%). We also extend a previous equality metric to quantify the rates of student participation, and found that it offered good differentiation between groups where one student contributed the most and groups where students contributed equally. We present example applications of our observation protocol related to general participation trends, the kinds of contributions students make, student-student bonding, and help-seeking patterns. Finally, we discuss future directions for use of our coding scheme as well as implications for implementing structured role-based cooperative learning online in the future.**

***Index Terms*—cooperative learning, observation protocol, undergraduate, computer science, online learning**

I. Introduction

Use of structured roles to facilitate cooperative learning is an evidence-based practice that has been shown to improve student performance, attitude, and persistence [1]–[3]. The combination of structured roles and activities also helps build students' process skills including communication and metacognition [4]. While these benefits have been shown in a variety of disciplines [5], [6], most prior work has focused on highly detailed, time-intensive qualitative analysis or individual differences in learning gains or affective qualities. This paper presents an observation protocol to bridge the strengths of qualitative and quantitative measures and to create a shared tool that can be used across modalities and cooperative learning implementations. The observation protocol can help researchers and practitioners answer questions such as the following: Do students participate equally in groups? Do students' contributions match their assigned role? How do students connect and bond with each other in groups? How do students decide seek help in groups?

This material is based upon work supported by the National Science Foundation under Grant No. DUE 21-21412 and Graduate Research Fellowship Program. This work is also supported by the Strategic Instructional Innovations Program in the Grainger College of Engineering at the University of Illinois, Urbana-Champaign.

II. Literature Review

A. Cooperative Learning and Structured Roles

Cooperative learning is an evidence-based, active learning technique that has been shown to improve student performance, attitude, and persistence [1]–[3]. Cooperative learning centers around small groups working together to learn [7] and promotes positive interdependence and accountability [8]. Structured roles are one way to promote these key qualities of healthy cooperation, while minimizing problematic group dynamics, such as freeloading. For example, Jigsaw assigns students different readings, which they then share out to their group. Jigsaw promotes accountability by having each student be responsible for their own reading, which in turn promotes positive interdependence by each student needing all other students' segments to be successful [9]. Pair programming, a technique borrowed from industry programmers, has pairs of programmers share a computer and uses the "driver" (who types code on the shared computer) and "navigator" (who guides the driver) roles to separate responsibilities [10]. Pair programming promotes accountability by separating access to the shared computer, which in turn promotes positive interdependence by having the pair communicate and work on a shared end product. Process Oriented Guided Inquiry Learning (POGIL) further separates responsibilities [11]. For example, the manager keeps the group on task and ensures everyone contributes; the recorder shares their screen and inputs answers; and the reflector ensures all students understand what's going on. Similar to Jigsaw and pair programming, POGIL promotes accountability by separating responsibilities and access to resources, which in turn promotes positive interdependence by having groups work together on a common task.

Compared to individual or traditional learning, structured roles have been shown to improve student performance [5], [9], [12], [13], interpersonal skills [9], affect and attitudes [13], and self-efficacy [6] at a variety of educational stages and across science, technology, engineering, and math (STEM) disciplines. However, the benefits of structured roles depend on careful implementation. Prior work recommends small and diverse (e.g., racial and gender composition; prior experience) groups [4], [14]–[16]. Additionally, individuals should be

graded on both individual and group performance [17]. Furthermore, roles should be rotated to expose all group students to different skills and to avoid stereotypical role adoption (e.g., women frequently taking on the recorder role) [18], [19]. Yet even when implementations follow these guidelines, inequitable group dynamics can still emerge [20].

B. Observing Cooperative Learning

Due to the COVID-19 pandemic, students in our context worked online in groups with little direct instructor interaction. Thus, we aimed to capture general group processes between students instead of domain-specific or whole classroom practices. Additionally, the protocol needed to be applicable to both online and in-person settings to account for shifting course modalities. Many of the studies cited earlier rely either on quantitative data sources such as surveys (e.g., [6]) and test results (e.g., [12]) or qualitative data sources such as ethnographic observations (e.g., [20]) and interviews (e.g., [19]). Thus, a common observation protocol provides a shared tool that bridges the strengths of qualitative and quantitative methods by allowing for quick analysis of group dynamics while still allowing for detail and depth.

Researchers have been developing observation protocols to capture group processing and dynamics; however, these protocols may be domain-specific, tend to assume strong instructor presence, and are typically for in-person contexts only. For example, the Reformed Teaching Observation Protocol (RTOP) [21], [22] was developed to capture reformed teaching across multiple educational levels, and contains 25 items each using a 5-point rating scale on lesson design, implementation, content, and classroom culture. Similarly, the Teaching Dimensions Observation Protocol (TDOP) [23], [24] was developed to capture postsecondary instructors' active learning teaching practices, and collects observations on pedagogy, teacher-student interactions, student engagement and tasks, and course delivery methods. Inspired by TDOP, the Classroom Observation Protocol for Undergraduate STEM (COPUS) [25], was developed as part of an institutional change initiative to better understand and evaluate teaching practices. Similar to TDOP, COPUS involves observers recording observations in 2-minute intervals, but simplifies TDOP by reducing the number of categories to "students doing" and "instructor doing," which in turn significantly reduced the training time for new observers. Due to the emphasis on teaching, RTOP, TDOP, and COPUS focus more on the classroom as a whole with greater granularity of codes on lesson design, pedagogy, and presentation styles and coarser granularity on students' actions and participation. On the other hand, the Classroom Observation Protocol for Engineering Design (COPED) [26] was developed to capture students engaging in the engineering design process in K-12 science classrooms. Similarly to TDOP and COPUS, COPED uses 2-minute intervals, but instead focuses on students exhibiting behaviors aligned with engineering design. Similarly to COPED, the Three-Dimensional Learning Observation Protocol (3D-LOP) [27] focused on student learning, but it was instead developed for postsecondary biology, chemistry, and physics classrooms and centered around *what* students learn rather than previous protocols' focus on *how* students learn. Additionally, 3D-LOP defined observations per "segment," i.e., class discussion on a topic, rather than time intervals. Recently, OPTIC was developed to capture POGIL activities in a whole classroom context, but it has not yet been validated [28].

In this paper, we present a complementary observation protocol that focuses on group processing. Thus, we aimed to create an observation protocol that met the following design constraints:

- Applicable for both in-person and online modalities
- Generalizable to different course contexts (e.g., different course structure or content)
- Fine-grained time intervals for fast-paced or dense conversation between students
- Course-grained codes that focus on observable actions and for ease of use

We use activity theory to frame our development of the observation protocol. Through the lens of activity theory, students bring their own knowledge to the activity, and through interaction, work together to transform objects (e.g., programming a solution) [29]. Thus, learning is inherently contextual and social (e.g., constraints of the task, division of labor). Furthermore, the tools used (e.g., autograder) can influence the learning process.

III. Online Cooperative Learning

A. Course Contexts

In response to fully online instruction during the Fall 2020 and Spring 2021 semesters, instructors from three computer science (CS) courses (Computer Architecture, Numerical Methods, and Database Systems) restructured their courses into flipped classrooms with POGIL-inspired, in-class activities as a part of a larger project to study cooperative learning in their classrooms. The activities were POGIL-inspired in that they used a subset of common POGIL roles (manager, recorder, and reflector), and groups were largely self-managed. However, due to constraints of scale (all courses enrolled $\sim$400 students each) and Zoom limitations, other POGIL roles such as having a reporter role (share out their group's findings in a whole class debrief) was not logistically possible. Additionally, the instructors typically started each class with a short lecture that was intended to address common student questions during the activity.

To ensure the observation protocol did not rely on the specifics of a particular course, the protocol was developed in all three courses, and two of the three instructors did not participate in the development of the protocol. One of the three instructors acted as a consultant during the development of the protocol, but did not directly influence what the protocol captured or how it was used. Next, we describe each of the course contexts in greater detail.

Computer Architecture is a 4-credit hour required course for the Computer Science major, and is typically taken by second year undergraduate students. The course met twice

a week for 1.25 hours each. Prior to virtual class meetings on Zoom, students were expected to complete short pre-class assignments individually. During synchronous class times, the instructor gave a brief lecture on the relevant topics and walked through some sample questions. Students spent the majority of class time working in small groups. All questions on the activity counted toward groups' collective grades. Question types included multiple choice, fill-in-the-blank, programming, and open-ended response. Students programmed Verilog, MIPS assembly, and C. Students were able to request help via an online queue. Students worked with the same group for the first half of the semester, then they were offered the option to stay with their group or choose a different group for the second half of the semester.

Numerical Methods is a 3-credit hour required course for the Computer Science major, and is typically taken by third year undergraduate students. Similarly to Computer Architecture, the course met twice a week for 1.25 hours each, and students were expected to complete short pre-class assignments individually before class. Unlike Computer Architecture, students worked in groups once a week with little to no instructor lecturing. The instructor used the second weekly class meeting as a live coding session to walk through more examples of course content with the whole class. The main activity counted toward groups' collective grades, and an additional activity providing more practice or exploring concepts in greater depth was optional. The main activity was divided into questions consisting of short programming prompts (e.g., using floating point, truncation, and rounding as three different implementations of arithmetic, then comparing the effects of each on the precision of bank account transactions). Students programmed in Python using the NumPy library in Jupyter Notebooks. Students were able to request help via an online queue. Students worked with the same group for the first half of the semester, then they were offered the option to stay with their group or choose a different group for the second half of the semester.

Database Systems is an elective 3-credit hour course for the Computer Science major, and an elective 4-credit hour course for graduate students. It is typically taken by fourth year undergraduate students, Master's students, and early-stage doctoral students. Similarly to the previous two courses, the course met twice a week for 1.25 hours each, and students were expected to complete short pre-class assignments individually before class. Similar to Computer Architecture, Database Systems had group activities during both weekly class meetings. At the beginning of each class, the instructor answered student questions, which were submitted via pre-class assignments, and walked through some sample questions. Students then spent the majority of class time working in small groups. All questions on the activity counted toward groups' collective grades. Question types included programming and diagram creation. Students were able to request help via an online queue. Students programmed in SQL, MongoDB, and Neo4j, and developed literacy in other representational forms such as relational algebra. Students worked with the same group for the entire semester as these groups also contributed to a separate, semester-long project.

All group activities started with the Manager Report (submitted by the manager) and ended with the Reflector Survey (submitted by the reflector). In the Manager Report, students self-assigned themselves to either the manager, recorder, reflector, or contributor roles. The contributor role was only present in groups of 4, and pairs were allowed to assign themselves to multiple roles (e.g., one student acted as recorder and reflector, and one student acted as manager). In all courses, students were encouraged to rotate roles throughout the semester, and instructors included role rotation as part of students' participation grades. All group activities were hosted on PrairieLearn [30], an online learning management system, that provided automatic feedback to student submissions as well as autograders for programming questions.

In summary, all courses used the same set of POGIL roles, included role-based components on each activity, and hosted activities on PrairieLearn. Aside from these factors, all three courses varied in terms of how often groups met, how many times groups switched, and the kinds of activities groups worked on. This enabled us to observe a wide variety of student behaviors within groups, leading to an observation protocol that can be applicable to many contexts.

B. Group Formation

During the first two weeks of the semester, due to fluctuating enrollment, groups were randomly assigned to Zoom breakout rooms. During these weeks, we collected informed consent and demographic data according to our IRB-approved process. After the first two weeks, students were allowed to pick a group of their choosing or were assigned to a group. All courses used groups of size 3-4. For students in Computer Architecture and Numerical Methods, a research assistant not affiliated with either course formed groups for students who did not pick a group. Group formation incorporated students' self-reported gender to ensure women and nonbinary students were not in the minority to minimize the chances of stereotypical role adoption [18], [19] and consent data to maximize the number of groups where all students consented to participate in the study for observation purposes. Because groups needed to be formed manually and quickly, we were unable to incorporate other students' self-reported identities. Minor adjustments to groups were made on a case-by-case basis (e.g., student signed up with a group but this choice was not reflected in the roster; student's group members did not appear for one class). After 6-8 weeks, students in Computer Architecture and Numerical Methods were offered the option of working with the same group, form a different group, or be assigned to a different group for the rest of the semester (15 weeks long in total). The same research assistant repeated the group formation process for students who opted to be assigned to a different group. For groups with a mix of students wanting to stay and leave, the research assistant tried to keep students who wanted to stay in the same group together, and assigned students who wanted to leave to a different group. For students

in Database Systems, a teaching assistant randomly formed groups for students who did not pick a group. The teaching assistant did not have access to student demographic or consent data. Students in Database Systems stayed in the same group for the whole semester as these groups also contributed to a separate, semester-long project.

IV. Creating the Observation Protocol

A. Observing Group Activities

Our protocol aimed to capture general group processes that allowed for ease of use and simple aggregation and analysis. To systematically record group activities, we designed codes for various activities of interest. The first prototype of our code book was designed shortly before the Spring 2021 semester based on intuitions of what we anticipated students would do. For example, we expected that students would ask questions and type answers, and we initially set a 1-minute time interval for observations. Following activity theory [29], we iterated on and revised the code book based on actual observations. For example, we originally limited "read" to reading the question prompt, and had separate codes for "explain" (explaining a concept or answer) and "solve" (offering a solution). After a few pilot observations at the beginning of the semester, we expanded "read" to be inclusive of anything visible on the screen (e.g., students also read autograder feedback) or stated intention to read. On the other hand, we collapsed "explain" and "solve" into just "explain" since differentiating them was difficult and combining them was simpler to code. We intentionally sought to keep the number of codes minimal for ease of use and reduce strains on observers' working memory. On the other hand, we added codes such as "info" based on observations of students seeking resources or help and "confirm" based on observations that students asked each other for confirmation on their ideas (e.g., students would offer an answer, immediately followed by "..., right?" to which other students would respond in agreement or disagreement). Additionally, the 1-minute time interval was too coarse to capture faster-paced or denser interactions, so we switched to 30-second time intervals for finer-grained observations that were still manageable for observers. Table I reflects the latest version of the code book.

The coding process itself was iterative. For each 30-second interval, we identified and classified each student's contributions using the code book, and within each interval, one kind of activity was only recorded once for each student to keep the observation protocol at a high level and for ease of recording observations. For example, if a student asked two questions within one interval, we only recorded "ask" once for this student. See Table II for sample observations from two observers.

TABLE I
Code book used to observe group activities

Code	Definition
ask	Person asks a question
contribute	Person asks group or student to contribute [Aligns with manager role]
check	Person asks group or student if they understand [Aligns with reflector role]
confirm	Person asks for confirmation (e.g., "... right?", "Does this look correct?")
y/n	Person provides short response to "ask" or "confirm" (e.g., "yeah," "no")
type	Person is visibly typing or annotating the screen [Aligns with recorder role]
read	Person is audibly reading or says they will read something on the screen
explain	Person explains concept or answer, may or may not be incorrect or in response to "ask"
casual	Person expresses emotion (e.g., "Yes! Full points!") or talks about non-activity related topic
info	Person says they will search or actually searches for information in lecture slides, course forum, etc. Includes suggesting to ask or actually asking for help from course staff

V. Quantitative Validation of the Observation Protocol

A. Data Collection

In the Spring 2021 semester, we aimed to code different groups to explore the variety of group dynamics and for validation of the protocol. For each class meeting, one of the three graduate student researchers virtually visited a group where all students consented to participate in the study and had not been previously visited. Due to the limited number of groups that were observable, a few groups were revisited toward the end of the semester or had two researcher visitors. These revisited or double-coded groups were not included in the validation of the protocol or analysis.

Upon entering the Zoom breakout room, we asked if the group was comfortable with being recorded, and only proceeded to record and perform observations if all students agreed. Each member of the research team was responsible for individually coding their observations live as they visited a group and used Zoom's local recording feature to capture audio and screensharing for later validation of the protocol. These recordings were uploaded to a secure Box folder that was only accessible to students of the research team and inaccessible to instructors. If a student decided afterwards that they did not want to be recorded, we deleted the associated recordings and did not include their group's data in the validation of the protocol or analysis. To perform live observations, the observer had both the Zoom breakout room and observation protocol[1] visible to capture spoken codes (e.g., "explain," "ask") and visual codes (e.g., "type," "info"). The observer also used a repeating timer to notify when to move on to the next interval. Of the 77 recordings collected from the Spring 2021 semester, 41 of the recordings were from Computer Architecture, 19 from Numerical Methods, and 17 from Database Systems.

[1]See https://tinyurl.com/BlankObservationProtocol for a blank version. The first tab is to be filled in by the observer (i.e., student's assigned roles, time intervals, and space for notes). The second tab holds definitions of the codes and simple formulas for aggregation of codes by role.

TABLE II
SAMPLE OF OBSERVATION PROTOCOL CODING. INTERVALS ARE IN 30-SECOND INCREMENTS.

	Observer 1's Coding				Observer 2's Coding			
Interval	**Manager**	**Recorder**	**Reflector**	**Instructor**	**Manager**	**Recorder**	**Reflector**	**Instructor**
0	read	info type			read explain	info type		
1	ask explain	info	explain confirm		ask explain	info ask	explain confirm	
2		casual info type	explain			casual info type	explain	

B. Exclusion Criteria

We set the following exclusion criteria because we were interested in real-time group dynamics and validating the observation protocol:

- Groups that completed a significant amount of the activity before class were excluded
- Groups that failed to share their screen were excluded
- Groups that mostly spoke a language other than English were excluded
- Groups where two or more students' voices were indistinguishable were excluded
- Groups where the corresponding recording was lost were excluded

Of the 61 recordings that passed the criteria, 32 of the recordings were from Computer Architecture, 17 from Numerical Methods, and 12 from Database Systems.

C. Interrater Reliability

From the recordings that passed the exclusion criteria, a subset of 22 recordings (out of 61; 36%) were selected using stratified random sampling where we stratified based on course and original observer to ensure all courses and observers were proportionally represented in the subset. The three graduate student researchers re-coded all recordings in the subset to account for differing start times between audio and screen recordings and live observations. Because our coding followed a one-to-many relationship (see Table II for an example), neither Krippendorff's α nor Fleiss' κ were appropriate interrater reliability metrics, so we instead relied on a simpler percent agreement, and set a target of $>80\%$ agreement.

We calculated percent agreement as follows: For each 30-second interval per student per pair of observers, agreement and disagreement were tallied based on the presence or absence of a code (i.e., agreement if present in both observers' observations, disagreement if present in one but not the other). Silence (no codes applied to a 30-second interval) was treated differently for a more conservative metric (i.e., agreement if both observers did not apply any codes). In some groups, a student would request help via an online queue, resulting in a member of the course staff to visit for a few minutes to address questions. Since course staff would have been marked as silent for most of the group activity (or all of the group activity if students did not request help), the "Instructor" column was not considered in calculating percent agreement to avoid inflating the metric.

For example, Table II shows the codes applied for two different observers on the same group of 3 students from Computer Architecture. In interval 0 (0-29 seconds of the recording), the observers have 4 agreements ("read" present for manager; "info" and "type" present for recorder; silence for reflector; instructor is ignored) and 1 disagreement ("explain" present for manager in Observer 2's but not in Observer 1's). In interval 1 (30-59 seconds of the recording), the observers have 5 agreements ("ask" and "explain" present for manager; "info" present for recorder; "explain" and "confirm" present for reflector) and 1 disagreement ("ask" present for recorder in Observer 2's but not in Observer 1's). In interval 2 (60-89 seconds of the recording), the observers have 5 agreements (silence for manager; "casual," "info," and "type" present for recorder; "explain" present for reflector) and 0 disagreements. Across these 3 intervals, there was a total of 14 agreements and 2 disagreements, resulting in 14/(14+2)=87.5% agreement.

Across all 22 recordings in the subset, percent agreement was 91.29%, indicating good interrater reliability. The observers revisited recordings where percent agreement was $>80\%$, but $<85\%$, and discussed discrepancies in interpretations of the codes to further clarify definitions. These clarifications are reflected in Table I.

D. Quantifying Equality of Participation

To quantify participation, for each student for each 30-second interval, we assign a 1 if the interval has at least one code applied and 0 if there were no codes applied (i.e., silence). For example, Table II shows both the manager and reflector would have 2 out of 3 intervals count for participation

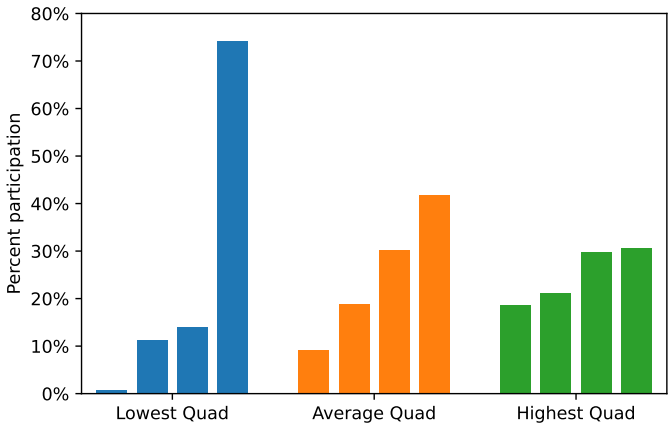

Fig. 1. Breakdown of student participation in groups that had the lowest ($e = 0.23$), average ($e = 0.67$), and highest ($e = 0.86$) equality metric for Quads (groups of 4). Students within groups are shown increasing order of percent participation from left to right.

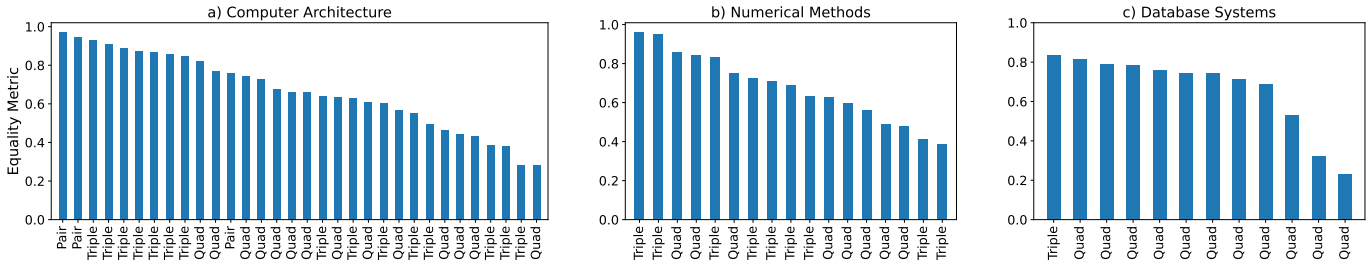

Fig. 2. Equality metric for groups in a) Computer Architecture, b) Numerical Methods, and c) Database Systems, in decreasing order. "Pair" means 2 students were present, "Triple" means 3 students were present, "Quad" means 4 students were present.

(intervals 0 and 1 for the manager; intervals 1 and 2 for the reflector).

Occasionally, the "info" code was present for multiple, consecutive intervals because a student had a resource visible on their screen, so we only counted the "info" code as participation once per time a resource was visible. For example, Table II shows a recorder with the "info" code present for 3 consecutive intervals. After reviewing the screen recording, the "info" code referred to the same resource, thus it was removed from intervals 1 and 2. This meant that the recorder's participation count was 2 intervals for Observer 1 (no codes present in interval 1) and 3 intervals for Observer 2 ("ask" code still present in interval 1). Alternatively, if a student pulled up a resource on their screen ("info" present), hid it ("info" not present), made it visible again ("info" present), and had no other codes applied, the student would have 2 intervals for participation. Similarly, if a student pulled up a resource on their screen ("info" present), then brought another resource in ("info" still present), and had no other codes applied, the student would also have 2 intervals for participation.

We extend the metrics for determining equality of participation from [31], which was based on number of student submissions in a group, to instead use the number of intervals where students participated. For each group, we calculate the equality of participation using the following equation:

$$e = 1 - \frac{\sigma_p}{\sigma_{max}} \tag{1}$$

Where σ_p is the standard distribution of number of non-empty 30-second intervals across all students in the group, and σ_{max} is the maximum possible standard distribution of number of non-empty 30 second intervals across all students in the group. Figure 1 shows the rates of participation for the groups with the lowest, average, and highest equality metric for Quads (groups of 4). The equality metric across all groups ranged from 0.23-0.97, which provides evidence for the wide variety of group dynamics we observed. For example, in the Quad with the lowest e, the recorder (right-most blue bar in Figure 1) seemed to drive conversation (and typing) for the majority of the time with occasional input from the other members. Similarly, in the Quad with an average e, the contributor and recorder (right-most orange bars in Figure 1) contributed roughly equally, but much more than the manager and reflector. Lastly, in the Quad with the highest e, all students seemed engaged and participated throughout the activity.

Figure 2 shows the equality metric calculated for all groups in each course. Across all 3 courses, we observed wide variation in the equality metric $(M = 0.669, SD = 0.189)$, which matched our informal observations of how students did or did not participate.

VI. Potential Uses of the Observation Protocol

In this section, we present examples of how our observation protocol can be useful to researchers and instructors and expand on future directions.

A. Examining Participation Trends and Role Alignment

The research team noticed differences between groups where all students participated and appeared actively engaged

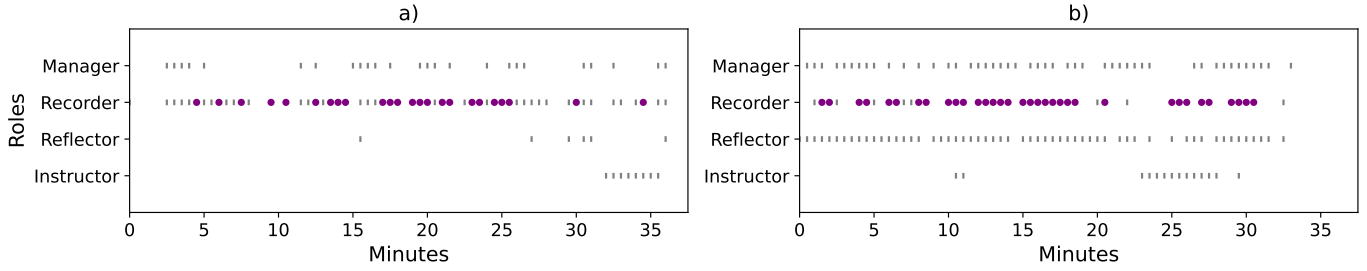

Fig. 3. Common participation trends where a) one student does not seem to be participating as much as the other two and b) where all students seem to be participating equally. Gray ticks indicate at least one code applied to a student during a 30-second interval. Purple dots indicate presence of a role-aligned code (see Table I).

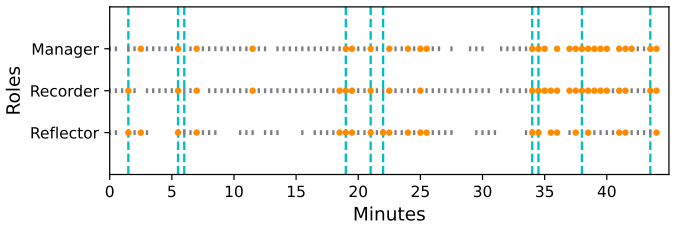

Fig. 4. Common pattern of "casual" codes occurring after submission in response to autograder feedback and non-activity related talk towards the end. Orange dots indicate the presence of "casual." Vertical, dashed lines indicate an interval where a student submitted a solution to the autograder.

and groups where one student seemed to participate much less. For example, Figure 3a shows a group of three students where contributions were primarily from the manager and recorder ($e = 0.38$). In contrast, Figure 3b shows a group of three where all students contributed throughout the session ($e = 0.89$). From these observations, the 0.51 difference in the equality metric between these two groups can help instructors quantify differences in participation.

Across most groups, role alignment (see Table I for which codes aligned with which role) seemed strongest for the recorder. For example, both groups in Figure 3 show the presence of role-aligned codes for the recorder only. This was perhaps due to the recorder's role (sharing their screen and typing answers) being most visually salient, whereas the manager's role (ensuring everyone contributes) becomes less salient when students are unable to see each other's facial expressions or gestures in Zoom breakout rooms. Alternatively, the lack of alignment for the manager and reflector may indicate a gap in how instructors referred to roles during class meetings.

Aggregating across groups, instructors and researchers may be interested in how the course they are observing compares to others. For example, upon visual inspection of Figure 2, the Database Systems course seems to have a lower equality metric on average. However, a one-way ANOVA revealed that there was not a statistically significant difference in the average equality metric across all three courses ($F(2, 58) = [0.023], p = 0.98$) at the level $\alpha = 0.05$.

B. Moments of Team-Bonding

Conversations not explicitly about the activity's content were coded as "casual." We observed that the "casual" code usually happened as the group submitted answers to the autograder and at the end of class. These times provided moments for team-bonding. Figure 4 shows a group that exhibited this behavior. For example, after re-watching the recording for the moments after the group's first attempt was correct, the recorder said, "Yay," the manager said, "Very nice," and the reflector said, "First try, so good." As the group wrapped up the activity, they talked about their exams, quizzes, emotional experiences in other courses, and had fun with submitting joking feedback on the Reflector Survey.

C. Help-seeking Patterns

The research team noticed two common types of help-seeking patterns. In the first scenario, students would get stuck on a question, then discuss asking for help via the online queue. One student would then offer to join the online help queue (i.e., digitally raise their hand), followed by instructor presence (see Figure 5a).

In the second scenario, the recorder would split their screen to have the activity on one half and a resource on the other (see Figure 5b). After re-watching the associated recording, the group appeared to be stuck at multiple points throughout the activity, which was part of the reason why the resource was left visible on the screen for the first 25 minutes. As instructors, we may want to encourage the group to ask for help more often, and as a result see more instructor presence from the observation protocol. In future work, we may try to integrate the online queue with PrairieLearn or incorporate encouragement in automated feedback to help lower the effort required to ask for help in an online setting.

VII. Discussion

A. Potential Differences in Online Settings

In-person settings can make gender and ethnicity more visually salient, increasing the risk of stereotypical roles, frequency of microaggressions, and potential for stereotype threat. Online settings may decrease these risks due to visual anonymity when video is turned off. In our context, many students we observed defaulted to leaving their camera off, which showed a default Zoom profile (i.e., name or first initial against a solid color background) in the breakout room. The research team did not observe microaggressions between students, but due to our group selection process and virtual presence, we cannot say with certainty that no microaggressions or other undesirable behaviors ever occurred.

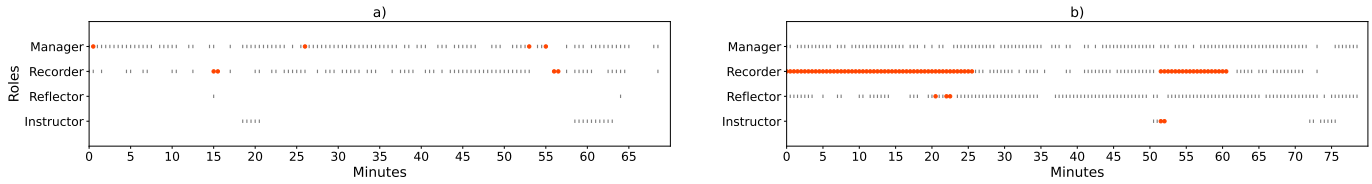

Fig. 5. Two common types of help-seeking patterns: a) asking for help via the online queue followed by instructor presence and b) referencing a resource that was visible on the shared screen. Red dots indicate the presence of "info."

B. Exploration Made Possible

Our results highlight the range of possible questions about group processes that our observation protocol can answer in both online and in-person settings. For example, we observed many instances of positive, "casual" talk across groups, and informal analysis of survey data indicated students generally enjoyed working with their groups. This speaks to the need for similar kinds of codes in other observation protocols. While TDOP includes the "humor" code for instructors [24], there is no equivalent for students, and other protocols that have student-focused elements center around how students engage with course content (e.g., RTOP [22], COPUS [25], COPED [26], 3D-LOP [27]) or may incorrectly characterize students as not participating (e.g., OPTIC [28]).

C. Implementing Structured Roles Moving Forward

Based on the role alignment we observed as well as informal analysis of survey data, the three POGIL roles we used did not seem to be helpful for groups that were satisfied with their working style. Part of this may have been due to COVID as students seemed more understanding of the impact of the pandemic on their peers' participation. Other students cited additional cognitive load (e.g., paying attention to roles distracted from the question they were trying to solve). We think that perhaps a pair programming style with one driver and multiple navigators may be easier for students to adopt while still scaling the number of groups to a manageable number for limited course staff resources. As all courses switched to hybrid offerings starting in the Fall 2020 semester, we aim to further explore the differences in structured role implementations across online and in-person modalities.

D. Adjusting the Observation Protocol for Other Contexts

While our observation protocol was developed with POGIL in mind, we believe the protocol can be easily adapted to different structured role implementations or added to other observation protocols. To adapt to other structured role implementations, for example, other POGIL implementations use a "reporter" role, where one student shares their group's findings in a whole class debrief. In this scenario, we may add a "share" code that captures the expected role-based behavior. Alternatively, we could change which role aligns with which code or remove codes for other structured role implementations. For example in pair programming, the "check" and "explain" codes could instead align with the navigator role, and the "contribute" code may no longer be necessary. For other observation protocols, many of the previously cited protocols explicitly code for when students work together (e.g., COPUS), thus our protocol can add another layer of depth by providing a protocol to "switch to" when investigating how students work together in the context of a classroom.

VIII. Limitations

Due to limitations of our IRB procedure, we were not allowed to record students' faces, so we were unable to incorporate students' gestures or facial expressions. Most students left their camera off by default, but this also meant we were unable to differentiate between students who chose not to participate and those who were not given an opportunity to participate. Additionally, students may have changed their behavior due to having a researcher in the breakout room with them; however the presence of "casual" codes in our data seems to indicate otherwise.

From the observers' perspective, recording observations was easier with the use of two monitors (i.e., one for the Zoom breakout room and repeating timer, one for the observation protocol) but still tricky, especially in conversation-dense groups of 3-4. Informal debriefs between observers indicated that "type" was difficult to detect for small edits (e.g., fixing a typo), and spoken codes (e.g., "explain," "ask") were difficult to record in instances of overlapping talk. Furthermore, we were unable to compare observers' original coding with re-coded versions because of differing start times (i.e., a small difference between the start time of the live observation and recording could cause cascading differences), so we are unable to speak to the accuracy of live observations.

Throughout the semester, all students were incentivized with extra credit to fill out surveys that included measures of sense of belonging, how students took up POGIL roles, and peer reviews. While these surveys provide important data to supplement our observations, they are out of scope for the present study.

IX. Conclusion

In this paper, we presented our observation protocol that was inspired by past observation protocols and grounded in actual classroom observations to better understand how students work together in structured role-based cooperative learning. A team of three graduate student researchers iteratively refined the protocol and validated it using recordings of groups across three CS courses. Percent agreement indicated good interrater reliability, and our use of an equality metric captured variations in student participation. For researchers and practicitioners, our protocol can show that whether students align with their assigned roles, whether participation is unequal across students within a group, whether there are moments for team-bonding, and how students seek help.

Acknowledgments

The authors would like to thank students of the Computers and Education research area for their invaluable feedback on earlier drafts of this work. The authors would also like to thank the Statistical Consulting Services at the University of Illinois, Urbana-Champaign for their advice on interrater reliability metrics.

References

[1] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, "Active learning increases student performance in science, engineering, and mathematics," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8410–8415, 2014.

[2] E. Kyndt, E. Raes, B. Lismont, F. Timmers, E. Cascallar, and F. Dochy, "A meta-analysis of the effects of face-to-face cooperative learning. do recent studies falsify or verify earlier findings?," *Educational Research Review*, vol. 10, pp. 133–149, 2013.

[3] J. M. Cámara-Zapata and D. Morales, "Cooperative learning, student characteristics, and persistence: an experimental study in an engineering physics course," *European Journal of Engineering Education*, vol. 45, no. 4, pp. 565–577, 2020.

[4] D. W. Johnson, R. T. Johnson, and K. A. Smith, *Cooperative learning: increasing college faculty instructional productivity*. No. 4 in ASHE-ERIC higher education report, School of Education and Human Development, George Washington University, 1991.

[5] R. S. Moog, J. N. Spencer, and A. R. Straumanis, "Process-oriented guided inquiry learning: POGIL and the POGIL project," *Metropolitan Universities*, vol. 17, no. 4, pp. 41–52, 2006.

[6] A. Yadav, C. Mayfield, S. K. Moudgalya, C. Kussmaul, and H. H. Hu, "Collaborative learning, self-efficacy, and student performance in CS1 POGIL," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pp. 775–781, ACM, 2021.

[7] J. van der Linden, G. Erkens, H. Schmidt, and P. Renshaw, "Collaborative learning," in *New Learning* (R.-J. Simons, J. van der Linden, and T. Duffy, eds.), pp. 37–54, Springer Netherlands, 2000.

[8] R. E. Slavin, "Cooperative learning," *Review of Educational Research*, vol. 50, no. 2, pp. 315–342, 1980. Publisher: American Educational Research Association.

[9] E. Aronson, "Building empathy, compassion, and achievement in the jigsaw classroom," in *Improving Academic Achievement: Impact of Psychological Factors on Education* (J. Aronson, ed.), Educational Psychology Series, pp. 209–225, Academic Press, 2002.

[10] L. Williams, R. Kessler, W. Cunningham, and R. Jeffries, "Strengthening the case for pair programming," *IEEE Software*, vol. 17, no. 4, 2000. Conference Name: IEEE Software.

[11] H. H. Hu and T. D. Shepherd, "Using POGIL to help students learn to program," *ACM Transactions on Computing Education*, vol. 13, no. 3, pp. 1–23, 2013.

[12] K. Doymus, "Teaching chemical equilibrium with the jigsaw technique," *Research in Science Education*, vol. 38, no. 2, pp. 249–260, 2008.

[13] K. Umapathy and A. D. Ritzhaupt, "A meta-analysis of pair-programming in computer programming courses: Implications for educational practice," *ACM Transactions on Computing Education*, vol. 17, no. 4, pp. 1–13, 2017.

[14] K. A. Smith, "Cooperative learning: Making "groupwork" work," *New Directions for Teaching and Learning*, vol. 1996, no. 67, pp. 71–82, 1996.

[15] P. Heller and M. Hollabaugh, "Teaching problem solving through cooperative grouping. part 2: Designing problems and structuring groups," *American Journal of Physics*, vol. 60, no. 7, pp. 637–644, 1992.

[16] L. K. Michaelsen and M. Sweet, "The essential elements of team-based learning," *New Directions for Teaching and Learning*, vol. 2008, no. 116, pp. 7–27, 2008.

[17] R. E. Slavin, "Group rewards make groupwork work," *Educational Leadership*, vol. 48, no. 5, pp. 89–91, 1991. Publisher: Association for Supervision & Curriculum Development.

[18] L. Meadows and D. Sekaquaptewa, "The influence of gender stereotypes on role adoption in student teams," in *2013 ASEE Annual Conference & Exposition Proceedings*, ASEE Conferences, 2013.

[19] D. Doucette, R. Clark, and C. Singh, "Hermione and the secretary: how gendered task division in introductory physics labs can disrupt equitable learning," *European Journal of Physics*, vol. 41, no. 3, pp. 1–20, 2020.

[20] N. Shah and C. M. Lewis, "Amplifying and attenuating inequity in collaborative learning: Toward an analytical framework," *Cognition and Instruction*, vol. 37, no. 4, pp. 423–452, 2019.

[21] D. Sawada, M. Piburn, J. Turley, K. Falconer, R. Benford, I. Bloom, and E. Judson, "Reformed teaching observation protocol (RTOP) TRAINING GUIDE," Tech. Rep. IN00-2, Arizona State University, 2000.

[22] D. Sawada, M. D. Piburn, E. Judson, J. Turley, K. Falconer, R. Benford, and I. Bloom, "Measuring reform practices in science and mathematics classrooms: The reformed teaching observation protocol," *School Science and Mathematics*, vol. 102, no. 6, pp. 245–253, 2002.

[23] M. T. Hora, A. Oleson, and J. J. Ferrare, "Teaching dimensions observation protocol (TDOP) user's manual."

[24] M. T. Hora, "Toward a descriptive science of teaching: How the TDOP illuminates the multidimensional nature of active learning in postsecondary classrooms," *Science Education*, vol. 99, no. 5, pp. 783–818, 2015.

[25] M. K. Smith, F. H. M. Jones, S. L. Gilbert, and C. E. Wieman, "The classroom observation protocol for undergraduate STEM (COPUS): A new instrument to characterize university STEM classroom practices," *CBE—Life Sciences Education*, vol. 12, no. 4, pp. 618–627, 2013.

[26] L. B. Wheeler, S. L. Navy, J. L. Maeng, and B. A. Whitworth, "Development and validation of the classroom observation protocol for engineering design (COPED)," *Journal of Research in Science Teaching*, vol. 56, no. 9, pp. 1285–1305, 2019.

[27] K. Bain, L. Bender, P. Bergeron, M. D. Caballero, J. H. Carmel, E. M. Duffy, D. Ebert-May, C. L. Fata-Hartley, D. G. Herrington, J. T. Laverty, R. L. Matz, P. C. Nelson, L. A. Posey, J. R. Stoltzfus, R. L. Stowe, R. D. Sweeder, S. H. Tessmer, S. M. Underwood, M. Urban-Lurain, and M. M. Cooper, "Characterizing college science instruction: The three-dimensional learning observation protocol," *PLOS ONE*, vol. 15, no. 6, pp. 1–20, 2020.

[28] "POGIL | OPTIC. https://pogil.org/pogil-tools/optic."

[29] D. H. Jonassen and L. Rohrer-Murphy, "Activity theory as a framework for designing constructivist learning environments," *Educational Technology Research and Development*, vol. 47, no. 1, pp. 61–79, 1999.

[30] "Prairielearn. https://www.prairielearn.org/pl."

[31] G. L. Herman, Y. Jiang, Y. Jiang, S. Poulsen, M. West, and M. Silva, "An analytic comparison of student-scheduled and instructor-scheduled collaborative learning in online contexts," in *2022 ASEE Annual Conference & Exposition Proceedings*, ASEE Conferences, 2022.