

## Engagement in collaboration and teamwork using Google Colaboratory

Alexandra Werth<sup>1,2</sup>, Kristin Oliver<sup>1,2</sup>, Colin G. West<sup>1</sup>, and H. J. Lewandowski<sup>1,2</sup>

<sup>1</sup>*Department of Physics, University of Colorado, Boulder, Colorado 80309, USA and*

<sup>2</sup>*JILA, National Institute of Standards and Technology and University of Colorado, Boulder, Colorado 80309, USA*

Google Colaboratory, or “Colab” for short, is a multiuser, collaborative environment that allows anyone with access to Google and the internet to write and execute arbitrary python code through their browser. With recent calls to increase use of computation in physics education, Colab has the potential to be a valuable tool to allow students to collaboratively code together—particularly in an online environment. Through this work, we examine how student teams navigated collaboration challenges related to using Colab in an online environment to conduct data analysis for a course-based undergraduate research experience in physics. We analyze students’ final written assignment of the course, a “memo to future researchers,” through the framework of socially-shared regulation of learning, to understand the challenges, regulations, and perceived goal attainment students discussed relating to their experience programming in teams online with Colab. We found that students struggled with version control issues when simultaneously writing, editing, and saving their work. This led to the need to use socially-shared regulatory strategies, including assigning and rotating roles from week to week and having clear, regular communication. Highlighting these students’ experiences and their advice to future researchers can help inform instructional guidance on how to best promote productive teamwork in collaborative coding environments both online and in person.

## I. INTRODUCTION

There have been numerous, recent calls to increase computation in physics education [1–9]. However, like many scientific fields [10], computing and computational physics is often viewed by students as a solitary practice [11, 12], particularly because programming done in class is often carried out in isolation. There are many reasons that may contribute to this, but one relevant concern is that many instructors believe students will not learn as much while programming in groups as they would if they completed the assignment alone [11, 13].

Solitary classroom practices, however, lie in stark contrast to industry and professional settings, in which scientists frequently rely on collaborations with colleagues to help them write code, troubleshoot, and keep up to date with the latest programming techniques [11, 14]. Furthermore, collaborative learning (i.e., the educational approach to teaching and learning that involves groups of learners working together to solve a problem, complete a task, or create a product [15]) has a variety of social, psychological, and academic benefits, such as developing a social support system for learners, increasing students' self esteem, promoting critical thinking skills, improving classroom performance, and increasing motivation [15]. In combination, emphasizing collaborative computing in class may not only be beneficial for engaging students in authentic scientific practices [11, 14], but also for learning [15] and retention [13].

Hence, it is important to understand how students collaboratively code in teams, common challenges faced when collaboratively coding, and suggestions for overcoming these challenges. Through this work, we examine a single course in which student teams navigated collaboration challenges related to using Google Colaboratory (Colab) in an online environment to conduct computational data analysis for a course-based undergraduate research experience (CURE) in physics [16]. We analyze one of the final written assignments of the course, a "memo to future researchers," through the framework of socially-shared regulation of learning [17] to understand the challenges students faced, the strategies (or "regulations") used to address those challenges, and the students' perceived goal attainment relating to their experience programming in teams online with Colab. The goal of this work is to highlight these students' experiences and their advice to future researchers in order to inform instructional guidance on how to best promote productive teamwork in collaborative coding environments both online and in person.

## II. BACKGROUND

**The CURE.** Our analysis focuses on students in a newly developed CURE, which is part of the Colorado PHysics Laboratory Academic Research Effort (C-PhLARE) [16]. The CURE is a 15-week course that seeks to answer a longstanding question in solar physics by exploring a proposed mechanism responsible for heating the sun's corona [18]. Small

teams of 3-4 students work to determine if nanoflares are the dominant heating mechanism by first using introductory-level physics and calculus, with basic python data analysis tools, to calculate the total energy of individual flares. They then calculate the solar flare frequency distribution (flare frequency rate versus energy in the long x-ray region) [19–21] using a dataset of over 1000 flares analyzed by student teams, and ultimately determine the power law of this distribution. The Fall 2020 version of this course had 440 students enrolled in 34 lab sections run by teaching assistants. Students predominately were engineering and physical science majors. A further description of the details of the CURE and demographic information of the students can be found in Ref. [16] and Ref. [22].

**Google Colaboratory.** In order for students to perform the analysis described above, Colab [23], which was released to the public in October 2019, was used throughout the CURE. Similar to Jupyter Notebooks [24], Colab notebooks allow one to combine executable code and rich text in a single document. However, it has some advantages over other programming environments because it is free, easily shareable, and, importantly for instructors, runs off of a Google server, so it does not require a fast, expensive GPU nor any setup on a student's personal computer. In addition, one of the greatest advantages is the ability to store Colab Notebooks in a Google Drive account and easily share them with others, allowing collaborators to leave comments in and edit the notebook. Given the apparent value of collaboration when learning programming [13, 14, 25–27], Colab has significant potential as a tool to allow students to collaboratively code together, including in online spaces. However, as we shall observe, it is important to note that Colab currently has version control issues if multiple people are editing the code at the same time (see Sec. IV).

**Relevant course goals.** There are two explicit features of the goals of this course relevant to this paper. First, despite regularly using Python, the CURE was explicitly *not* designed to teach students to code more generally. No prior knowledge of programming was expected from the students; thus, students learned Python the same way many scientists first learn a new programming language, by looking at examples of the code in action and then modifying it as necessary to learn how it functions and adapt it to their needs. This idea was messaged to students throughout the semester. Second, it *was* an explicit objective of the course that students should have productive and enjoyable teamwork experiences [16, 28]. To facilitate this, there were several educational interventions in the course such as purposeful group formation—students were assigned to teams such that they had *similar* levels of prior coding experience—and a teamwork training lab session, amongst other elements [16, 28]. Likewise, students were explicitly given tools to guide their collaborative coding experience using Colab. In one of the assignments they were told, "...to prevent version control issues, do not have multiple people editing code in Google Colab at the same time." Some "team roles" were also suggested,

such as a checker (who checks to make sure all group members understand the concepts and the group's conclusions), coder (who is in charge of writing new ideas/code in Colab), and questioner (who raises counter-arguments and introduces alternative solutions). Students were encouraged to "*rotate roles after each week so everyone has a chance to contribute.*" Because teamwork concepts were emphasized throughout all aspects of the course, no single element of the class can be identified as either a necessary or sufficient cause of the findings seen in this study. However, these are important examples of how one can potentially cultivate and encourage positive collaborative coding experiences.

**Prior findings related to teamwork.** Previous work studying this CURE [16, 28] found that the vast majority of students in the Fall 2020 semester described their teamwork experiences positively, with over 80% reporting that teamwork was fun, helped them stay motivated, helped them learn, and allowed them to conduct the research more successfully [16]. Additional work to further understand the team dynamics through socially-shared regulated learning theory found that students overwhelmingly achieved their goals (85.6%) and felt that their team played a positive role in their success (82.9%), possibly because the majority of the class did not find "social" teamwork challenges particularly difficult to overcome [28].

**Collaborative learning.** As described above, collaborative learning (CL), or grouping students for the purpose of achieving a learning goal, was a key component of the CURE. CL has been widely researched and advocated for since the 1980s [29] as an instructional method in which students work together in groups toward a common goal. In CL, learners are responsible for achieving their personal goals, as well as the goals of others, and therefore, students must help other students to be successful [30]. CL environments not only help students improve classroom performance related to specific content learning goals [15], but also develop valuable collaborative skills in of themselves [31]. Additionally, CL can improve various psychological and social attributes, (for example, by increasing motivation) [15] and has specifically been shown to improve persistence in STEM courses [32]. While CL environments are becoming more prevalent throughout all of academia, STEM labs in particular tend to emphasize and foster group work. In fact, it has been suggested that one of the primary goals for undergraduate physics labs should be to develop "interpersonal communication skills" through "teamwork and collaboration" [33].

**Challenges with collaborative learning.** Despite the benefits, CL can pose an array of challenges both for students and instructors alike. A study by Le et al. [34] found that students and instructors cited four common obstacles to collaboration: students' lack of collaborative skills, social loafing [35], competence status, and friendship. For students, CL can be more challenging to navigate than independent learning because students need to overcome additional challenges that only emerge due to the social nature of the environment [17, 36, 37]. For example, students participating in

group work may have irreconcilable personal goals [38, 39], differing styles of work and communication [40], differing levels of commitment, concentration, or standards [41], and differing levels of prior knowledge, understanding of concepts, or power [42]. Likewise, instructors often face challenges monitoring CL environments to ensure productive group work (e.g., managing on-task behaviour, managing group-work time, providing relevant materials, assigning individual roles, and establishing teamwork beliefs and behaviours) [34, 43] and often do not have training on the best ways to address such challenges or promote and manage CL environments generally [34].

Challenges with CL are particularly present in computation and programming [11–13, 26], possibly because, when programming in a group, there can only be one "driver" (i.e., one who has control of the mouse/keyboard and is writing the design or code) at a time [13, 26]. This leads instructors to be concerned that students programming in groups may not learn as much as they would if they completed the assignment alone [11, 13], with non-drivers getting a "free ride" [11]. This sentiment is echoed in student beliefs, as many student programmers equate working together with cheating [27]. In fact, however, it has been shown that cheating is often reduced when collaboratively programming [25].

These challenges are further exacerbated in the physics classroom, where the integration of computation into the curriculum is still fairly new [9]. A national survey of physics faculty [9] found that formal classes in computational physics are absent from most departments, although the vast majority physics graduates report programming in their jobs [44]. There have been growing calls to increase computation in undergraduate coursework [1–9], but, as these are primarily new courses, there is a steep learning curve for instructors and effective practices still need to be developed [8].

**Evaluation of collaborative learning, socially-shared regulation of learning theory.** Instructors and researchers face a lack of assessment tools to measure collaborative performance [45], and often struggle to evaluate how students learn and interact in complex collaborative environments [46, 47]. One theoretical framework that *has* been used to better understand CL environments is regulated learning theory [46]. Self-regulated learning (SRL) refers to a cycle in which a student engages in setting goals and selecting strategies, implementing strategies, monitoring their performance, and then reflecting and adapting [48]. However, by definition, SRL involves only the individual. Additional theories, co-regulation of learning (CoRL) and socially-shared regulation of learning (SSRL) have added to SRL to represent this cyclical process in pair and group learning environments [49]. In this work, we focus on primarily SSRL, in which several individuals regulate their activity in a genuinely shared way [17]. For example, the challenges students face during the "implementing strategies" stage of the learning cycle may be collective (e.g., "we have trouble using this technology as a team") and thus, the regulations of these challenges can be socially-shared as well (e.g., "we need to work together to determine

a more effective distribution of roles").

### III. METHODOLOGY

This work analyzes the “memo to future researchers,” a written assignment at the end of the Fall 2020 term. We asked students to write a one-page memo introducing the next class of researchers to the project. The memo served as a chance for students to engage in metacognitive reflection and in the authentic research practice of “handing off their experiment to the next researcher.” A vast majority of the students (405/440) completed the memo. The prompt asked students to (1) describe the project such that someone new can understand what we are doing, (2) summarize what conclusions we can draw from our research results, (3) discuss their personal experience (e.g., What did they learn this semester? Do they have any advice they would give them about the analysis, working with Colab, or teamwork?) and (4) suggest ideas for how this project could continue in the future.

To analyze the memos, we used a multi-level coding scheme. We started with an *a priori* codebook containing seven codes: affect, authenticity, coding, community, identity, learning and teamwork, which reflected our motivations for the course. We then isolated the teamwork codes and applied another *a priori* codebook to those codes which had four main codes (1) goals, (2) challenges, (3) regulations, and (4) perceived goal attainment. This second codebook was based on regulated learning theory and the AIRE questionnaire [17, 28]. A further discussion of our use of the AIRE questionnaire for our coding scheme can be found in Ref. [28]. Each of these four new codes had a number of sub-codes created both *a priori*, again based on the AIRE questionnaire, and emergently during the coding process. AW and KO independently coded two distinct subsets of the responses using the second codebook (22 teamwork responses) and determined the percent agreement [50] between the two raters to be 93.2%. After establishing inter-rater reliability, the entirety of the data set was divided and coded by AW and KO using the codebook. All additional emergent codes added after the initial inter-rater reliability were discussed and agreed upon by AW, KO, and the rest of the research team.

### IV. RESULTS AND DISCUSSION

The “memos to future researchers” assignment showed a consistent theme—collaboratively coding in Colab clearly posed a challenge to many students and their teams. Together, of the 217 students who mentioned teamwork challenges in their memos, 161 (74.2%) referenced technological challenges stemming from using Colab. Many students also recommended a variety of productive regulations for the “future researchers” to try that worked successfully for them, most involving socially-shared regulations (SSRs) such as working as a team and rotating roles. We provide exam-

TABLE I. Code counts of common regulations to challenges with Colab in their memos.

Common regulations to challenges with Colab	Number of students (n = 70)
Rotated roles / divided the work	38
Utilized Zoom features	15
Communicated with team	7
Worked together	3

ples detailing the challenges faced by students using Colab and the numerous productive solutions (regulations) to those challenges suggested to the future researchers.

Many of the challenges stemmed from Colab’s *dissimilarity* to more familiar Google products. Similar to Google Docs, Colab allows the user to create a shareable link to allow others to view, edit, and comment on the notebook. Unlike Google Docs, however, Colab has significant version control issues when multiple students are writing, editing, and saving at the same time. Students explicitly discussed how Colab negatively impacted their ability to work as a team: “*One of the most challenging aspects of this entire course was not anything rigorous like a difficult concept or topic, it was actually Colab. This software is very new and buggy and does not allow for streamlined collaboration. It was very hard to simultaneously work as a group on a single Google Colab document because it does not act like a standard Google Doc. When two or more people are typing on it at the same time, the document has a very hard time saving the work that multiple people have done.*” Another student explained to a future researcher, “*...only one person can physically write on it [Colab] at a time or else the entire system freaks out.*” One student expressed that the challenges of working with Colab as a team “*made it difficult to ensure the entire research team was on the same page*” (i.e., a social teamwork challenge) and required increased regulations, because students need to “*clearly and thoroughly communicate with each other.*”

Seventy of the 161 students who referenced challenges with Colab in their memos also brought up regulations they used to overcome challenges. Table I shows that the most common regulation reported by students was dividing up the work as a team and assigning clear team roles—a SSR: “*...one member of my team would screenshare a google colab document over zoom. They would be the person writing the code, while the other gave their input. Afterwards one member would comment the code and the other would write descriptions of the code.*” Assigning clear “roles” while collaboratively coding has been recommended an effective and authentic way to code in pairs or groups. Work by McDowell et al. [13] suggest that this can even be done successfully with one member of the group as the “driver” who actively creates code and controls the keyboard and mouse while the “non-drivers” constantly review the code for errors in the implementation. Students from the CURE suggested something

similar: “I would suggest sharing the notebook, but have one person do all the typing on the notebook for that week... However, to make sure everyone contributes to the project, I would rotate who edits the notebook per week.” A second student wrote: “...I recommend one person be the scribe for the document while the other group members tell what the person to write.” However, a key component of successful group coding is that the group regularly rotates roles [13], as expressed by this student: “[rotating roles] not only solves issues with Colab, but also allows everyone to get practice leading and using their skills.” In contrast, assigning roles does not always equate to dividing the work. One student discussed working together for overcoming differing prior coding experience: “...I think the work can be made easier by working as a group on each problem and code cell, as opposed to assigning individual tasks to each team member. This allows for greater collaboration and it also helps any student who may not be as comfortable with coding or with the physics concepts be able to understand what is going on and not leave them behind...” While it is not clear whether this student’s team took on specific roles, it is clear that they did not divide up the work into subsections for each student to complete independently.

In the face of the challenges posed by Colab, the students relied heavily on their teams to overcome this obstacle, demonstrating their collaborative skills: “It is very important to be patient and respectful with the people you are working with especially if this is their first-time coding. Google colab can be very frustrating and confusing for a first-time user, so make sure to help your teammates succeed.” Note the empathy expressed by this student, emphasizing being “patient and respectful” especially of “first-time” coders or peers who many have differing levels of prior coding experience. In addition, this student implies that one of their goals was for everyone on the team to succeed—a sign of successful SSR. Emphasizing the importance of having shared team goals throughout the coding process, another student wrote: “If everyone knows what exactly is being achieved, especially on google colab, then the process will be smooth, and your results will be of high quality. I really can’t stress enough on the team aspect of this project.” Many students brought up the need for increased communication, both in regards to working with Colab, as seen in quotations such as (1) “Colab is not as simple, my best recommendation is to communicate with your team and don’t rely on Colab to save all your work.” (2) “My team did an excellent job of communicating to each other in every lab section...” (3) “Lack of consistent communication leads to a degradation of the team’s goals and makes the process much more difficult for the one or two people who are invested in the research.”

## V. CONCLUSIONS AND KEY TAKEAWAYS

Previous work found that collaborative coding in physics is important to consider as both an authentic scientific practice, a means to develop essential collaboration skills, and

as a way to increase learning [11, 14, 25, 26]. These considerations are particularly relevant given the numerous, recent calls to increase computation in physics education [1–9]. However, CL, and in particular team programming on a single computer, can pose a variety of challenges for students and instructors. For example, many students in the course expected Colab to operate like more familiar Google products and thus ran into frustration when faced with version control issues. Despite this frustration, most students overcame the challenges by using SSRs, such as dividing and rotating roles, working together closely, increasing communication, and being respectful of one another. From our analysis of the “memos to future researchers,” we detail two key takeaways that may help guide future instruction of collaborative coding, particularly when using Colab.

**Message early and often.** Message about (1) the expected challenges of working in Colab, (2) the importance of relying on one’s team for overcoming these challenges, and (3) the authenticity and usefulness of collaborative coding. This type of messaging can set students’ expectations and communicate the motivation for the collaborative nature of the course. One can provide this messaging through an interactive “training” notebook that is designed to demonstrate common pitfalls of collaborative coding and provides suggestions for how to overcome those challenges through teamwork. It may be useful also to emphasize the “authentic” nature of challenges like this—professional researchers are often frustrated with the limitations of the tools they have to work with, and students should not view such challenges as an artifact of a limited or inauthentic classroom experience, particularly in a lab or CURE.

**Suggest team roles.** Clear, rotating roles can be a key component of productive collaborative coding. Although some student teams may discover the best roles through experimentation, many students will have little-to-no formal training on working in a collaborative environment [34] and in particular may lack prior experience with collaborative coding. Therefore, suggesting productive team roles, such as those listed in Sec. II or Ref. [13], gives students a more direct opportunity to learn and practice collaborative coding skills. In addition, concerns about social loafing and cheating are often expressed by students and instructors alike during group work [27]—suggesting roles can help normalize one person “driving” the coding process while everyone else still actively contributes. However, it is essential that the roles rotate amongst the team members, although monitoring this can be challenging for instructors. Some suggestions are to enlist the help of teaching assistants or undergraduate learning assistants, if available, or have students document authorship and detail author contributions at the end of each assignment.

## ACKNOWLEDGMENTS

This work is supported by STROBE NSF STC, Grant No. DMR-1548924 and PHY-1734006.

- 
- [1] E. Behringer, AAPT recommendations for computational physics in undergraduate physics curricula, *Bulletin of the American Physical Society* **62** (2017).
- [2] R. Landau, Computational physics: A better model for physics education?, *Computing in science & engineering* **8**, 22 (2006).
- [3] R. H. Landau, Computational physics education; why, what and how, *Computer physics communications* **177**, 191 (2007).
- [4] R. H. Landau, J. Paez, and C. C. Bordeianu, A survey of computational physics, in *A Survey of Computational Physics* (Princeton University Press, 2011).
- [5] C. Henderson and M. H. Dancy, Impact of physics education research on the teaching of introductory quantitative physics in the united states, *Physical Review Special Topics-Physics Education Research* **5**, 020107 (2009).
- [6] A. L. Graves and A. D. Light, Hitting the ground running: Computational physics education to prepare students for computational physics research, *Computing in Science & Engineering* **22**, 50 (2020).
- [7] R. F. Martin, Undergraduate computational physics education: uneven history and promising future, in *Journal of Physics: Conference Series*, Vol. 759 (IOP Publishing, 2016) p. 012005.
- [8] M. D. Caballero, N. Chonacky, L. Engelhardt, R. C. Hilborn, M. L. del Puerto, and K. R. Roos, Picup: A community of teachers integrating computation into undergraduate physics courses, *The Physics Teacher* **57**, 397 (2019).
- [9] M. D. Caballero and L. Merner, Prevalence and nature of computational instruction in undergraduate physics programs across the united states, *Phys. Rev. Phys. Educ. Res.* **14**, 020129 (2018).
- [10] J. Osborne, S. Collins, M. Ratcliffe, R. Millar, and R. Duschl, What ideas-about-science should be taught in school science? a delphi study of the expert community, *Journal of research in science teaching* **40**, 692 (2003).
- [11] P. Maguire, R. Maguire, P. Hyland, and P. Marshall, Enhancing collaborative learning using pair programming: Who benefits?, *All Ireland Journal of Higher Education* **6** (2014).
- [12] K. Redmond, S. Evans, and M. Sahami, A large-scale quantitative study of women in computer science at stanford university, in *Proceeding of the 44th ACM technical symposium on Computer science education* (2013) pp. 439–444.
- [13] C. McDowell, L. Werner, H. E. Bullock, and J. Fernald, Pair programming improves student retention, confidence, and program quality, *Communications of the ACM* **49**, 90 (2006).
- [14] C. McDowell, B. Hanks, and L. Werner, Experimenting with pair programming in the classroom, in *Proceedings of the 8th annual conference on Innovation and technology in computer science education* (2003) pp. 60–64.
- [15] M. Laal and S. M. Ghodsi, Benefits of collaborative learning, *Procedia-social and behavioral sciences* **31**, 486 (2012).
- [16] A. Werth, C. G. West, and H. J. Lewandowski, Impacts on student learning, confidence, and affect and in a remote, large-enrollment, course-based undergraduate research experience in physics, *Phys. Rev. Phys. Educ. Res.* **18**, 010129 (2022).
- [17] H. Järvenoja, S. Volet, and S. Järvelä, Regulation of emotions in socially challenging learning situations: an instrument to measure the adaptive and social nature of the regulation process, *Educational Psychology* **33**, 31 (2013), <https://doi.org/10.1080/01443410.2012.742334>.
- [18] J. A. Klimchuk, On solving the coronal heating problem, *Solar Physics* **234**, 41 (2006).
- [19] H. S. Hudson, Solar flares, microflares, nanoflares, and coronal heating, *Solar Physics* **133**, 357 (1991).
- [20] N. B. Crosby, M. J. Aschwanden, and B. R. Dennis, Frequency Distributions and Correlations of Solar X-ray Flare Parameters, *Solar Physics* **143**, 275 (1993).
- [21] A. Veronig, M. Temmer, A. Hanslmeier, W. Otruba, and M. Messori, Astronomy & Astrophysics Temporal aspects and frequency distributions of solar soft X-ray flares, *Astronomy & Astrophysics* **382**, 1070 (2002).
- [22] R. L. Hughes and S. K. Jones, Developing and assessing college student teamwork skills, *New directions for institutional research* **2011**, 53 (2011).
- [23] G. LLC, [Google Colaboratory faq page](#) (2020).
- [24] Jupyter, [Jupyter](#) (2022).
- [25] L. Williams and R. L. Upchurch, In support of student pair-programming, *ACM SIGCSE Bulletin* **33**, 327 (2001).
- [26] L. Williams, R. Kessler, W. Cunningham, and R. Jeffries, Strengthening the case for pair programming, *IEEE Software* **17**, 19 (2000).
- [27] L. A. Williams and R. R. Kessler, The effects of " pair-pressure" and " pair-learning" on software engineering education, in *Thirteenth conference on software engineering education and training* (IEEE, 2000) pp. 59–65.
- [28] A. Werth, K. Oliver, C. G. West, and H. J. Lewandowski, Student engagement with teamwork in an online, large-enrollment course based undergraduate research experience in physics, *Physical Review Physics Education Research*, in preparation (2022).
- [29] K. A. Bruffee, Collaborative learning and the " conversation of mankind", *College English* **46**, 635 (1984).
- [30] A. A. Gokhale, Collaborative learning enhances critical thinking, (1995).
- [31] D. Boud, Making the move to peer learning, *Peer learning in higher education: Learning from and with each other* **1**, 20 (2001).
- [32] L. Springer, M. E. Stanne, and S. S. Donovan, Effects of small-group learning on undergraduates in science, mathematics, engineering, and technology: A meta-analysis, *Review of Educational Research* **69**, 21 (1999), <https://doi.org/10.3102/00346543069001021>.
- [33] J. Kozminski, H. J. Lewandowski, N. Beverly, S. Lindaas, D. Deardorff, A. Reagan, R. Dietz, R. Tagg, M. Eblen-ÂZayas, J. Williams, R. Hobbs, and B. Zwickl, *AAPT Recommendations for the Undergraduate Physics Laboratory Curriculum Subcommittee Membership*, Tech. Rep. (American Association of Physics Teachers Committee on Laboratories, 2014).
- [34] H. Le, J. Janssen, and T. Wubbels, Collaborative learning practices: teacher and student perceived obstacles to effective student collaboration, *Cambridge Journal of Education* **48**, 103 (2018).
- [35] B. Latanè, K. Williams, and S. Harkins, Many hands make light the work: The causes and consequences of social loafing, *Journal of Personality and Social Psychology* **37**, 822â832 (1979), <https://doi.org/10.1037/0022-3514.37.6.822>.
- [36] S. Volet, M. Summers, and J. Thurman, High-level co-regulation in collaborative learning: How does it emerge and how is it sustained?, *Learning and Instruction* **19**, 128 (2009).

- [37] H. Järvenoja and S. Järvelä, How students describe the sources of their emotional and motivational experiences during the learning process: A qualitative approach, *Learning and Instruction* **15**, 465 (2005).
- [38] P. C. Blumenfeld, R. W. Marx, E. Soloway, and J. Krajcik, Learning with peers: From small group cooperation to collaborative communities, *Educational researcher* **25**, 37 (1996).
- [39] P. Van den Bossche, W. H. Gijsselaers, M. Segers, and P. A. Kirschner, Social and cognitive factors driving teamwork in collaborative learning environments: Team learning beliefs and behaviors, *Small group research* **37**, 490 (2006).
- [40] S. E. Volet and G. Ang, Culturally mixed groups on international campuses: An opportunity for inter-cultural learning, *Higher education research & development* **17**, 5 (1998).
- [41] L. N. Makewa, D. Gitonga, B. Ngussa, S. Njoroge, and J. Kuboja, Frustration factor in group collaborative learning experiences, *American Journal of Educational Research* **2**, 16 (2014).
- [42] G. Salomon and T. Globerson, When teams do not function the way they ought to, *International Journal of Educational Research* **13**, 89 (1989).
- [43] R. M. Gillies and M. Boyle, Teachers' reflections on cooperative learning: Issues of implementation, *Teaching and teacher Education* **26**, 933 (2010).
- [44] P. Mulvey and J. Pold, Physics bachelors: Initial employment. data from the degree recipient follow-up survey for the classes of 2013 and 2014. focus on., AIP Statistical Research Center (2017).
- [45] P. S. Strom and R. D. Strom, Teamwork skills assessment for cooperative learning, *Educational Research and Evaluation* **17**, 233 (2011).
- [46] S. Järvelä, A. Hadwin, J. Malmberg, and M. Miller, Contemporary perspectives of regulated learning in collaboration, *International handbook of the learning sciences* , 127 (2018).
- [47] C. J. McCarthy, T. A. Whittaker, L. H. Boyle, and M. Eyal, Quantitative approaches to group research: Suggestions for best practices, *The Journal for Specialists in Group Work* **42**, 3 (2017).
- [48] D. L. Butler and P. H. Winne, Feedback and self-regulated learning: A theoretical synthesis, *Review of Educational Research* **65**, 245 (1995), <https://doi.org/10.3102/00346543065003245>.
- [49] E. Panadero and S. Järvelä, Socially shared regulation of learning: A review, *European Psychologist* **20**, 190 (2015), <https://doi.org/10.1027/1016-9040/a000226>.
- [50] K. Gwet, Kappa Statistic is not Satisfactory for Assessing the Extent of Agreement Between Raters, Series: Statistical Methods For Inter-Rater Reliability Assessment (2002).