Engaging Minds: Impact of Sequential Live Coding on Academic Engagement

Kwansun Cho, University of Florida

Kwansun Cho is an Instructional Assistant Professor of the Department of Engineering Education, in the UF Herbert Wertheim College of Engineering. She has been teaching introductory computer programming courses for engineers. She holds two Masters' degrees in Electrical and Computer Engineering from the University of Florida and Yonsei University, specializing in speech signal processing. Her educational research interests include improved flipped classroom teaching/learning for students, and computer- or web-assisted personalized learning.

Sung Je Bang, Texas A&M University

Sung Je Bang is a Ph.D. candidate in Interdisciplinary Engineering at Texas A&M University, within the Department of Multidisciplinary Engineering. He serves as a graduate research assistant on multiple projects, where he focuses on user experience and psychological aspects of technology. His research interests include artificial intelligence, large language models, user experience design, and engineering education.

Syeda Fizza Ali, Texas A&M University

Syeda Fizza Ali is currently pursuing her PhD in Interdisciplinary Engineering at Texas A&M University. She works as a graduate research assistant at the Department of Multidisciplinary Engineering, where her work focuses on educational technology tools and student engagement. Her research is shaped by a commitment to broadening participation in engineering and fostering learning environments that prioritize student well-being.

Dr. Asefeh Kardgar, Texas A&M University

Asefeh Kardgar is a researcher at Texas A&M University.

Dr. Saira Anwar, Texas A&M University

Saira Anwar is an Assistant Professor at the Department of Multidisciplinary Engineering, Texas A and M University, College Station. She received her Ph.D. in Engineering Education from the School of Engineering Education, Purdue University, USA. The Department of Energy, National Science Foundation, and industry sponsors fund her research. Her research potential and the implication of her work are recognized through national and international awards, including the 2023 NSTA/NARST Research Worth Reading award for her publication in the Journal of Research in Science Teaching, 2023 New Faculty Fellow award by IEEE ASEE Frontiers in Education Conference, 2022 Apprentice Faculty Grant award by the ERM Division, ASEE, and 2020 outstanding researcher award by the School of Engineering Education, Purdue University. Dr. Anwar has over 20 years of teaching experience at various national and international universities, including the Texas A and M University - USA, University of Florida - USA, and Forman Christian College University - Pakistan. She also received outstanding teacher awards in 2013 and 2006. Also, she received the "President of Pakistan Merit and Talent Scholarship" for her undergraduate studies.

Engaging Minds: Impact of Sequential Live Coding on Academic Engagement

Abstract

Computer programming presents conceptually hard concepts to all undergraduate students. Particularly, engineering students struggle with the logic and syntax details of programming languages. Due to the complex nature of the programming courses, instructional methods such as live coding, peer programming, and hands-on exercises have been implemented to improve students' learning outcomes. However, prior literature also notes some issues with such methods, such as passive attention and limited hands-on experience. Considering these issues, a new instructional mechanism, sequential live coding, was developed and utilized in a large R1 university for engineering programming (C++ and Python) courses. Our prior work suggested sequential live coding positively affected students' learning and perceptions of learning programming. However, its impact on students' non-cognitive factors, particularly engagement, is unexplored. Considering the importance of academic engagement for students' deeper understanding of course material, this paper examines the effect of sequential live coding on academic engagement. More specifically, the paper addresses the following research question: How does academic engagement differ between students who participated in sequential live coding and those who did not? To answer the question, we collected data from 63 undergraduate students enrolled in two programming courses, Python and C++. The data were collected pre- and post-manner using a previously validated engagement survey instrument measuring four dimensions of engagement: behavioral, emotional, social, and cognitive. Students reported their engagement levels across 24 items on a six-point Likert scale. In this quantitative study, the data were analyzed using an independent samples t-test between each dimension of engagement. The results indicate a significant difference in students' emotional engagement after using sequential live coding. Also, the descriptive statistics indicate that students who participated had improved engagement in all aspects except cognitive engagement. The study's results highlight that students' engagement mostly declines in conceptually hard courses like programming. However, students who participated in sequential live coding had higher engagement with the course than those who didn't participate. The study's results warrant creating learning environments that foster engagement to improve student's learning outcomes.

Introduction

Undergraduate students generally find computer programming concepts difficult to learn [1], [2] often due to a lack of an appropriate learning environment [4], few opportunities to practice learned materials [5], or a less effective pedagogical approach in the courses [6]. Difficulties in programming courses may also emerge due to the inherent nature of these courses. Mostly, in such courses, students may be required to divide a complex problem into smaller tasks to solve it. Also, programming requires a combination of a hierarchy of smaller skills [1]. For instance, in this hierarchy of smaller skills, students usually learn starting from syntax and gradually build up towards semantics, structure, and coding style [1]. These transitions could be hard for novice students, especially if they struggle with syntax details and the logic of programming languages [1], [7], [8]. Consequently, introductory programming has a high attrition rate among undergraduate students [3].

To combat undergraduate students' struggles with learning programming concepts, researchers conducted studies to develop more intuitive instructional methods [9], [10], [11], [12]. Such methods include live coding [9], [10], pair programming [11], and hands-on activities [12]. Live coding, being one of these instructional methods, is highly effective because it promotes non-cognitive factors (e.g., motivation, engagement) of students' learning outcomes [13], [14], [15], [16]. However, the literature also notes some drawbacks of this approach [9], [17], [18], [19]. The most prominent critique is that live coding can easily create a passive environment where the instructor is coding, and students are passive observers only, not actively participating in the coding process [9], [19].

To address the potential drawbacks of the live coding approach, in this paper, we discuss our novel approach called sequential live coding [20]. This approach requires that peers in the class participate by helping the students code to provide a solution to the problem, thus creating an engaging environment. Sequential live coding promotes student-centered learning and active engagement by asking students to take turns and live code in front of their peers [20]. Considering the interactive nature of this approach, we hypothesize that this approach will improve students' academic engagement. More specifically, the following research question guided this study: How does academic engagement differ between students who participated in sequential live coding and those who did not?

Related Literature

The premise of this study is situated in two fundamental theories: 1) constructivism and 2) self-system motivation theory.

Considering that sequential live coding provides a hands-on learning experience to programming students with interactive feedback from peers and later by the instructor, it creates opportunities for the construction of knowledge through learning by doing [21], [22]. The approach emphasizes the importance of learning with the method of inquiry for students [21]. Also, it allows the students to construct new ideas and concepts based on their ability to connect past and present knowledge [22]. With sequential live coding, students can be encouraged to learn by performing live coding in front of their peers [20], and students can make connections between what they know from the material they studied before the class.

Self-system motivation theory describes students' academic engagement as a fundamental facet of students' academic outcomes [23]. Prior literature describes student engagement as a multidimensional construct, which refers to students' meaningful interaction with the course material [24]. For meaningful interaction, students actively participate in contextually relevant classroom activities. Prior literature describes four dimensions of academic engagement: 1) Behavioral, where students participate, are involved in, and pay attention to classroom materials; 2) Emotional, based on students' affective responses, either positive or negative, towards the activities. 3) Social, where students indicate their level of interaction with their peers while working on activities. 4) Cognitive, where students indicate their willingness to put effort into learning the course material.

Various active learning approaches have been introduced for programming courses, and their effectiveness on student outcomes has been examined [25]. Among these approaches, live coding [9], [10] and pair programming [9] are the most commonly used approaches. Prior literature provides mixed results on the effectiveness of these active learning approaches in programming courses, where besides improvements in some students' outcomes, literature also suggests their issues. For example, on a positive note, in addition to learning, while studies appreciate the ability of live coding to enhance students' debugging skills [26], [10], there is evidence that pair programming could enhance students' peer skills and integration of feedback in pair programming [14] as well. However, it is noteworthy that studies have also noted situations where active learning approaches resulted in a more stressful environment and caused a decline in student outcomes [27].

Considering the advantages of live coding and pair programming approaches that give students learning autonomy, sequential live coding is a combination approach. With its effective design, sequential live coding, similar to effective active learning approaches [28], could impact students' cognitive and non-cognitive outcomes. Although prior literature has examined the impact of this newly developed approach on students' learning [18], it is important to examine its effect on students' non-cognitive factors.

Research Design and Methods

The study uses a quantitative approach and a correlational cross-sectional research design to answer the research question.

Site and Participants

The data were collected from 63 undergraduate non-CS major engineering students enrolled in one of the two introductory programming courses taught by the same instructor at the University of Florida (a large R1 southeastern university): C++ (39 students) and Python (24 students). Table 1 presents the information on students' demographics for both classes.

	C++ N=39		Python N=24	
	Number of students	Percentage	Number of students	Percentage
Gender				
Male	28	71.79%	12	50%
Female	10	25.64%	10	41.67%
Prefer not to disclose	1	2.56%	2	8.33%
Race/Ethnicity				
Hispanic or Latino, or Spanish Origin of any race	5	12.82%	3	12.50%
Asian	9	23.08%	5	20.83%
Black or African American	1	2.56%	0	0%
White	18	46.15%	10	41.67%
Two or more races	2	5.13%	5	20.83%
Nonresident Alien	1	2.56%	0	0%
Race and ethnicity unknown	1	2.56%	0	0%
Prefer not to disclose	2	5.13%	1	4.17%

Data collection occurred during the Fall of 2024 when students voluntarily participated in the sequential live coding activities throughout the semester. Table I. Student demographicsThe courses specifically covered the basics of programming languages, such as inputs, outputs, data types, flow of control (branching and looping), functions, classes, pointers, and introductory object-oriented programming.

Course Design for Sequential Live Coding

Both C++ and Python courses, with about 45 enrolled students, use a flipped class model, dividing each week into three stages of students' work. 1) Before the class – students engaged with the learning material (video lecture and other material) and attempted a weekly quiz before coming to the class. 2) During class, students worked on 2-3 in-class problems. During this time, the instructor asked students to engage in sequential live coding to solve the class problems. 3) After the class – students work on homework problems individually and submit them by the end of the week.

The sequential live coding activity is conducted during the in-class portion. Students already have some familiarity with the topic and its basic operations. During the 50-minute class session, students are given five minutes to familiarize themselves with the problem and create pseudocode for the solution. Following initial preparation, students engage in sequential live coding, where five to six students take turns to code the solution for about 30 minutes. All students can see each step and provide feedback through a shared screen, working together towards the common goal. The session is interactive, and students answer their peers' questions. Also, the rest of the class helps the students debug code and identify errors in case of coding errors.

Meanwhile, the instructor acts as a facilitator. The instructor notes down the issues and mistakes made by the students. In the end, the instructor conducts a backward-style lecture for about 15 minutes to highlight the key points and reinforce the important concepts of the week.

Data Collection and Measures

In this study, we focused on two aspects of data collection: 1) Participation metrics indicating whether a student participated in a group that performed sequential live coding. 2) Students' engagement – collected using a pre-existing instrument, validated for the context of programming courses, "The Math and Science Engagement Scales" [29]. The modified version comprises 24 questions, where the only modification was changing the context from Math and science courses to programming. Table II presents the sample survey items for each dimension. Additionally, the survey, administered via Qualtrics, asked students about their demographic information. The engagement instrument collects data on students' engagement in four dimensions: behavioral (6 items), emotional (6 items), social (6 items), and cognitive (6 items). The data were collected in a pre-post manner (Pre: at the beginning of the semester; post -around the end of the semester). The survey items asked students to indicate their agreement with the prompts based on a 6-point Likert scale: one indicated "strongly disagree," and six indicated "strongly agree." This paper only used the data from students who provided complete sets, i.e., completed both pre and post-survey.

Table II: Sample survey items

Dimension	**Sample Survey items**

Behavioral	I put effort into learning programming
Emotional	I feel good when I am in programming classes.
Social	I work with classmates to come up with ways to solve problems in programming classes.
Cognitive	I try to understand my mistakes when I get something wrong in programming classes.

Procedure and Data Analysis

For engagement, to bring the data on the same scale for analysis, we first reverse-coded all the negatively worded items (~11 items). Also, we examined the data for outliers, skewness, kurtosis, and multicollinearity and found no issues to report. As we only considered complete sets, there were no missing data items. For analysis purposes, an average was calculated for each dimension of engagement for both pre and post-surveys.

For the participation metric, we considered 15% participation as a threshold. If a student participated in 15% of the total sequential live coding session, we considered that they participated in sequential live coding (represented as 1). Students who participated in less than 15% of activities were marked as not participating (represented as 0). The 15% criteria were developed to ensure that students participated in sequential live coding of more than 1 activity.

To analyze the data, we utilized the statistical software SPSS V 30.0. To answer the research question, we conducted two mean comparisons. The first mean comparison was performed using an independent samples t-test, where we identified the difference in four dimensions of engagement between students who participated in sequential live coding and those who did not. In addition, we conducted a paired sample t-test to examine the difference between pre-and post-engagement for all four dimensions. Before running the statistics, we tested the data to determine the assumptions of the statistics. The data normality was observed using QQ plots. Levene's statistics were used to evaluate the homogeneity of the data, where the results indicated a homogenous sample.

Results

To answer the research question, we first conducted the independent samples t-test to identify the difference in four dimensions of engagement between students who participated in the sequential live coding and students who did not participate in at least 15% of the sequential live coding activities. Table III presents the results of the analysis.

Table III: Mean difference between students who participated in sequential live coding and those who did not

	Participated N= 26		**Not Participated N=37**				
Dimension	**Mean**	**SD**	**Mean**	**SD**	**t(61)**	**p**	**Cohen's d**
Pre-Engagement							
- Behavioral	5.121	.569	5.134	.649	-.079	.937	-.020
- Emotional	5.314	.542	5.136	.723	1.062	.292	.272
- Social	5.013	.926	4.946	.778	.310	.757	.079
- Cognitive	4.769	.403	4.766	.546	.027	.979	.007
Post Engagement							
- Behavioral	5.250	.624	5.004	.770	1.344	.184	.344
- Emotional	5.333	.587	4.919	.799	2.250*	.028	.576
- Social	5.051	.834	4.789	.800	1.500	.139	.384

- Cognitive	4.609	.577	4.590	.581	.127	.899	.032

*$p<.05$, **$p<.01$

For both kinds of students, the pre-engagement presents a non-significant difference. This result indicates that all students started with similar levels of engagement. However, in the post-engagement, the results indicate a significant difference between the students who participated vs. those who did not participate on the dimension of emotional engagement with $t(61) = 2.250$, $p=.028$, and effect size of .576, which is indicative of a moderate to large effect [30]. Students who participated indicated significantly better emotional engagement than those who did not.

Further, we conducted a paired sample t-test to examine the difference between pre-and post-engagement for all four dimensions. We conducted separate analyses for students who participated in sequential live coding and students who did not participate. The results of the analysis are presented in Table IV.

Table IV. Comparison between pre and post-engagement for students who participated and students who did not participate

	Participated			**Not Participated**		
Engagement	**t (25)**	**p**	**Cohen's d**	**t (36)**	**p**	**Cohen's d**
Behavioral	-1.293	.208	-.254	1.427	.162	.235
Emotional	-.245	.809	-.048	2.360*	.024	.388
Social	-.298	.768	-.058	1.664	.105	.274
Cognitive	1.266	.217	.248	1.783	.083	.293

*$p<.05$, **$p<.01$; *Negative results are indicative that pre-engagement is less than post-engagement*

Some noteworthy aspects of these results indicate that although non-significant, students who participated have improved behavioral, emotional, and social engagement from pre to post, while their cognitive engagement declined. However, engagement declined in all dimensions for students who didn't participate. Also, for students who didn't participate, the decline in emotional engagement was significant with $t(36) = 2.360$, $p=.024$, and Cohen's d of .388, which indicates a low to moderate effect size [30].

Discussion and Conclusion

The study measures the impact of a new instructional approach called sequential live coding on students' academic engagement using the principles of self-system motivation theory [23] and academic engagement principles [24]. This paper describes the process of integrating sequential live coding into two programming courses, i.e., C++ and Python, where students voluntarily participated in class activities designed for sequential live coding. Also, the study examines the difference between students who participated and those who didn't participate in their pre- and post-engagement.

The study's results highlighted that sequential live coding could be an effective instructional approach to promote student engagement. Although all students started the course with similar engagement (non-significant difference in pre-engagement), the students who did not participate in sequential live coding showed lower emotional engagement in the course at the end. Also, we found that the mean engagement for all dimensions declined for students who didn't participate (significant decline in emotional engagement). Meanwhile, mean engagement for students who participated showed increased behavioral, emotional, and social dimensions.

The results of the study are novel and intriguing for various reasons. First, the study results align with existing studies that suggest that actively engaging students in learning activities could promote better student outcomes. Second, in prior literature, it is observed that students often decline in conceptually hard courses like programming, and students' engagement declines from pre- to post. Although the study had insignificant differences in mean observation, there was an increase in the number of students who participated in the sequential live coding. Third, these results highlight the importance of stress-free learning environments in conceptually hard courses such as programming.

The study has several limitations, which can help in understanding the results. First, the study uses a correlation research design and does not have control group data. Future studies can be designed by comparing traditional (without sequential live coding) and experiment group (with sequential live coding) designs for conclusive impact analysis. Second, the sampling strategy does not adhere to randomized control samples and thus has limitations and less generalizability to the interpretation of results. Future studies can consider overcoming this issue. Third, the study had a small sample size, specifically the students participating in sequential live coding (N<30). Future studies can consider larger implementation, and one strategy could be to consider making minimum participation in sequential live coding a mandatory aspect of the course. Fourth, the study was based on self-reported student engagement. Future studies could consider other modes and process data to measure engagement. Fifth, in this study, only one non-cognitive factor, i.e., engagement, is considered. Other constructs of students' cognitive and non-cognitive outcomes could be considered in the future to describe the effectiveness of this approach. Lastly, although it reported students' demographics, this study didn't consider them for analysis. Future studies could look into student variations based on their demographics.

The results of this study are novel and provide insight into how their participation in sequential live coding impacted students' academic engagement. Also, the results highlight the importance of pedagogical approaches in programming courses.

References

[1] T. Jenkins, "On the difficulty of learning to program," in *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, Aug. 2002, pp. 53–58.

[2] B. Hanks, C. McDowell, D. Draper, and M. Krnjajic, "Program quality with pair programming in CS1," in *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*, Leeds United Kingdom: ACM, Jun. 2004, pp. 176–180. doi: 10.1145/1007996.1008043.

[3] T. W. Goodman, "Identifying students who may experience difficulty in an introductory computer science course," in *Proceedings of the 28th annual Southeast regional conference on - ACM-SE 28*, Not Known: ACM Press, 1990, p. 182. doi: 10.1145/98949.98988.

[4] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Comput. Sci. Educ.*, vol. 13, no. 2, pp. 137–172, Jun. 2003, doi: 10.1076/csed.13.2.137.14200.

[5] B. Li, F. Ning, L. Zhang, B. Yang, and L. Zhang, "Evaluation of a practice system Supporting distributed practice for novice programming students," *J. Pac. Rim Psychol.*, vol. 15, p. 18344909211008264, Jan. 2021, doi: 10.1177/18344909211008264.

[6] R. Scherer, F. Siddiq, and B. Sánchez Viveros, "A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions," *Comput. Hum. Behav.*, vol. 109, p. 106349, Aug. 2020, doi: 10.1016/j.chb.2020.106349.

[7] M. Hristova, A. Misra, M. Rutter, and R. Mercuri, "Identifying and correcting Java programming errors for introductory computer science students," in *Proceedings of the 34th SIGCSE technical symposium on Computer science education*, Reno Navada USA: ACM, Jan. 2003, pp. 153–156. doi: 10.1145/611892.611956.

[8] D. McCall and M. Kolling, "Meaningful categorisation of novice programmer errors," in *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, Madrid, Spain: IEEE, Oct. 2014, pp. 1–8. doi: 10.1109/FIE.2014.7044420.

[9] A. Gaspar and S. Langevin, "Active learning in introductory programming courses through student-led 'live coding' and test-driven pair programming," presented at the International Conference on Education and Information Systems, Technologies and Applications, Orlando, FL, Jul. 2007.

[10] A. G. S. Raj, J. M. Patel, R. Halverson, and E. R. Halverson, "Role of live-coding in learning introductory programming," in *Proceedings of the 18th Koli Calling International Conference on Computing Education Research*, Koli Finland: ACM, Nov. 2018, pp. 1–8. doi: 10.1145/3279720.3279725.

[11] N. Salleh, E. Mendes, and J. Grundy, "Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review," *IEEE Trans. Softw. Eng.*, vol. 37, no. 4, pp. 509–525, Jul. 2011, doi: 10.1109/TSE.2010.59.

[12] V. Handur *et al.*, "Integrating class and laboratory with hands-on programming: Its benefits and challenges," in *2016 IEEE 4th International Conference on MOOCs, Innovation and Technology in Education (MITE)*, Madurai, India: IEEE, Dec. 2016, pp. 163–168. doi: 10.1109/MITE.2016.041.

[13] M. J. Rubin, "The effectiveness of live-coding to teach introductory programming," in *Proceeding of the 44th ACM technical symposium on Computer science education*, Denver Colorado USA: ACM, Mar. 2013, pp. 651–656. doi: 10.1145/2445196.2445388.

[14] L. A. Williams and R. R. Kessler, "Experiments with industry's 'pair-programming' model in the computer science classroom," *Comput. Sci. Educ.*, vol. 11, no. 1, pp. 7–20, Jan. 2001, doi: 10.1076/csed.11.1.7.3846.

[15] H.-T. Wu, P.-C. Hsu, C.-Y. Lee, H.-J. Wang, and C.-K. Sun, "The impact of supplementary hands-on practice on learning in introductory computer science course for freshmen," *Comput. Educ.*, vol. 70, pp. 1–8, Jan. 2014, doi: 10.1016/j.compedu.2013.08.002.

[16] S. L. Canfield and M. Abdelrahman, "Enhancing the programming experience for first-year engineering students through hands-on integrated computer experiences," *J. STEM Educ.*, vol. 13, no. 4, Jun. 2012.

[17] A. Selvaraj, E. Zhang, L. Porter, and A. G. Soosai Raj, "Live coding: A review of the literature," in *Proceedings of the 26th ACM Conference on Innovation and Technology in*

Computer Science Education V. 1, Virtual Event Germany: ACM, Jun. 2021, pp. 164–170. doi: 10.1145/3430665.3456382.
[18] R. E. Bruhn and P. J. Burton, "An approach to teaching Java using computers," *ACM SIGCSE Bull.*, vol. 35, no. 4, pp. 94–99, Dec. 2003, doi: 10.1145/960492.960537.
[19] B. Stephenson, "Coding demonstration videos for CS1," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, Minneapolis MN USA: ACM, Feb. 2019, pp. 105–111. doi: 10.1145/3287324.3287445.
[20] K. Cho, S. F. Ali, S. Bang, and S. Anwar, "Shaking The Silos: Impact of Sequential Live Coding on Students' Performance and Perceptions," in *2024 ASEE Annual Conference & Exposition Proceedings*, Portland, Oregon: ASEE Conferences, Jun. 2024, p. 47966. doi: 10.18260/1-2--47966.
[21] C. Fincher, "Learning theory and research," in *Higher education: handbook of theory and research*, New York, NY: Agathon Press.
[22] J. Dewey, *Democracy and education: An introduction to the philosophy of education*. Macmillan New York, 1930.
[23] J. P. Connell, "Context, self, and action: A motivational analysis of self-system processes across the life span," in *The self in transition: Infancy to childhood*, D. Cicchetti and M. Beeghly, Eds., University of Chicago Press, 1990, pp. 61–97.
[24] J. D. Finn and K. S. Zimmer, "Student engagement: What is it? Why does it matter?," in *Handbook of Research on Student Engagement*, S. L. Christenson, A. L. Reschly, and C. Wylie, Eds., Boston, MA: Springer US, 2012, pp. 97–131. doi: 10.1007/978-1-4614-2018-7_5.
[25] A. K. Mbiada, B. Isong, F. Lugayizi, and A. Abu-Mahfouz, "Introductory computer programming teaching and learning approaches: Review," in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, Prague, Czech Republic: IEEE, Jul. 2022, pp. 1–8. doi: 10.1109/ICECET55527.2022.9873427.
[26] J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming," *ACM SIGCSE Bull.*, vol. 39, no. 2, pp. 32–36, Jun. 2007, doi: 10.1145/1272848.1272879.
[27] M. Morgan, M. Butler, J. Sinclair, C. Gonsalvez, and N. Thota, "Contrasting CS student and academic perspectives and experiences of student engagement," in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, Larnaca Cyprus: ACM, Jul. 2018, pp. 1–35. doi: 10.1145/3293881.3295777.
[28] M. Prince, "Does active learning work? A review of the research," *J. Eng. Educ.*, vol. 93, no. 3, pp. 223–231, Jul. 2004, doi: 10.1002/j.2168-9830.2004.tb00809.x.
[29] M.-T. Wang, J. A. Fredricks, F. Ye, T. L. Hofkens, and J. S. Linn, "The math and science engagement scales: Scale development, validation, and psychometric properties," *Learn. Instr.*, vol. 43, pp. 16–26, Jun. 2016, doi: 10.1016/j.learninstruc.2016.01.008.
[30] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2 ed. Routledge, 2013. doi: 10.4324/9780203771587.