Infusing Cooperative Learning into AP Computer Science Principles Courses to Promote Engagement and Diversity

Jeff Gray
University of Alabama
Tuscaloosa, AL
gray@cs.ua.edu

Kathy Haynie
Haynie Research & Evaluation
Skillman, NJ
kchaynie@alumni.stanford.edu

Fran Trees
Rutgers University
New Brunswick, NJ
fpt9@cs.rutgers.edu

Owen Astrachan
Duke University
Durham, NC
ola@cs.duke.edu

Chinma Uche
Academy of Aerospace &
Engineering
Windsor, CT
chinmauche@gmail.com

Siobhan Cooney
Cooney Collaborative
Narbeth, PA
siobhan@cooneycollaborative.com

Richard Kick
Newbury Park High School
Newbury Park, CA
kickrg@gmail.com

ABSTRACT

The Advanced Placement Computer Science Principles (AP CSP) course was the culmination of an eight-year NSF/College Board pilot project that exceeded all expectations in terms of enrollment in its first two official years. Four NSF-sponsored projects and six other projects have endorsed AP CSP curricula and professional development (PD), a first for an AP course. In this paper, we report on an NSF-sponsored multi-year effort to infuse cooperative learning (CL) structures into AP CSP classrooms to improve class participation and student learning. As we report, CL structures have been beneficial for both new and experienced teachers and across the curricula of the endorsed providers. Since AP CSP was designed to engage all learners, the CL structures used in our PD workshops and the CL resources designed by participating teachers in our project have the potential to positively impact all AP CSP classrooms.

Research was conducted on the extent to which use of CL structures impacted student efficacy and student achievement. Three cohorts of AP CSP teachers participated in PD that focused on AP CSP pedagogical content knowledge using CL structures. A cumulative 143 teachers attended one-week PD workshops spanning July 2015, 2016, and 2017. We studied the effect of CL structures on student learning using AP scores as an outcomes measure. Use of CL structures was a statistically significant and positive predictor of student AP scores for participating classes in cohorts 2 and 3. Additionally, the use of pair programming was a significant and positive predictor of AP scores.

ACM Reference format:

J. Gray, K. Haynie, F. Trees, O. Astrachan, C. Uche, S. Cooney, R. Kick. 2019. Infusing Cooperative Learning into AP Computer Science Principles

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SIGCSE '19, February 27–March 2, 2019, Minneapolis, MN, USA.

© 2019 Association of Computing Machinery.

ACM ISBN 978-1-4503-5890-3/19/02...\$15.00. <http://dx.doi.org/10.1145/3287324.3287421>

Courses to Promote Engagement and Diversity. In *Proceedings of 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19), February 27–March 2, 2019, Minneapolis, MN, USA*. ACM, New York, NY, USA, 7 pages. <http://dx.doi.org/10.1145/3287324.3287421>

1 Introduction

Broadening the accessibility of computing to include a diverse population of students has been a fundamental design goal of the AP CSP curriculum framework (CF). The “Big Ideas” of AP CSP provide more breadth of topic coverage than the current AP CSA exam in a manner that is appealing to students with varied interests and abilities. The recognition and celebration of the social impacts of computing, within a context that encourages students to explore their own creativity through the creation of computational artifacts, opens more opportunities for teachers to engage students.

However, *curriculum alone does not necessarily engender engagement*. The most interesting and innovative curriculum can be taught in a disengaged manner, leading to lost opportunities for broadening interest in computing across a diverse student population. The objectives of the project described in this paper are to demonstrate the benefits of infusing an AP CSP course with a pedagogical approach that has been demonstrated to engage diverse learners who are in the same classroom. The learning science literature on Cooperative Learning (CL) [1] has been shown to increase class participation and student learning using pedagogical structures and strategies that offer an environment where every student is encouraged to participate equally toward their own academic achievement, while simultaneously working toward a positive common goal with teammates from different backgrounds and mixed abilities. This paper introduces our investigation into infusing CL into AP CSP and provides the results of observed best practices of CL that can be applied across the AP CSP curriculum framework.

1.1 Cooperative Learning for Computer Science

The phrase “group learning” has been used to describe educational activities in which students work in clusters or groups of two or more. However, such group work [2] is not cooperative learning,

and the two are often misunderstood [3, 4]. Group work (or group discussion) is an unstructured activity in which the teacher asks a question or announces a topic to a class that has been physically organized in groups of 3-4 students. The students are then instructed to discuss the topic or solve the problem in their groups. In group learning, there are no specific guidelines on who speaks first and how long each student speaks. Characteristics of this type of discussion include unequal participation with the chance that one student dominates the discussion and others do not participate at all. This may foster a competitive, rather than cooperative, learning environment [5]. In contrast to group work, CL has built-in structures that naturally drive the equal participation of all students. Cooperative learning focuses on four key concepts (called “PIES” by Kagan [4]):

1. *Positive Interdependence*: Positive interdependence is realized when group members perceive that group success is based on every group member succeeding. Each member’s efforts benefit the entire group [6], gains for one imply benefits for the others, and cooperation replaces competition. CL activities are structured so that no one person can work alone, but all group members must work together to succeed. This interdependence fuels cooperation.
2. *Individual Accountability*: In addition to group accountability, individual accountability exists when an individual in the group is able to perform and succeed without assistance. The assessment of the individual is made public to the group. It is the responsibility of the group to provide assistance, support, and encouragement to individual members who need it [6].
3. *Equal Participation*: CL structures ensure that each group member participates in the learning activity, and no one can just be a passive listener. Student participation contributes to achieving expected outcomes [7].
4. *Simultaneous Interaction*: With CL structures, the amount of active engagement increases. Participation of many class members occurs simultaneously. In groups of four, 25% of the students in the class are speaking and the others are actively listening at any one time.

Although CL structures are content-independent, this paper describes our three-year project that has incorporated CS content (particularly AP CSP content) to provide activities that are engaging and that address AP CSP learning objectives. Our project invited teacher participants who have adopted all of the various endorsed AP CSP curricula – our project was not focused on any specific AP CSP curriculum provider. The pedagogical approach that drives our CL activities involves the equal participation of 100% of the students in the class. No student has the opportunity to “tune-out.” We have dozens of CL structures that we have applied to AP CSP. Below are just a few example CL structures and example activities we have used:

1. *Round Robin* is a CL structure that provides each student with an opportunity to speak exclusively for an equal amount of time, alternating between students at specified time intervals.

As an example of applying CL to AP CSP, we introduce Round Robin with an activity that asks students to share their thoughts on effects of computing in other fields, or the beneficial (or harmful) effects of social media. Active involvement (speaking) occurs with 25% - 50% of the students in class at any one time.

2. To access higher-order thinking skills, a teacher may use *The Debate Team Carousel* [8]. In this activity, the students are in teams of four. Each student is given a worksheet with a yes-no question on the top of the worksheet that will require them to analyze a problem prompted by the teacher. Think-time is provided. The remainder of the worksheet is divided into four quadrants. The upper-left quadrant provides the prompt, “Write your opinion and one reason why you think that way.” Students are given a specified amount of time to write their answer in the designated quadrant. Students pass their worksheet to the person on their right. A specified amount of time is given to respond to the prompt in the upper-right quadrant, “Read your partner’s response and add another reason that you would **support** what your partner wrote.” Again, the papers are passed to the right for the third prompt in the lower-left quadrant to be answered in a specified amount of time: “Write a reason that might be used to argue **against** what is written in the boxes above.” And finally, passing the papers one more time, the prompt in the last quadrant is, “Read what is written in all three boxes. Add YOUR opinion and YOUR reason here.”
3. To work on teambuilding skills, a teacher may announce a topic, set a time limit, give all students a short quiet (think) time to formulate ideas or answers to a question such as, “How does the Internet impact your privacy?” The teacher will then initiate CL by setting a timer. During a prescribed time (~1 min) students, in turn, will say an answer, write it on a slip of paper, and place the paper in the center of their group table with no papers overlapping. This structured activity (called *Jot Thoughts* by Kagan [4]) involves active listening, encouragement, praise, and a chance to build on the ideas of classmates.
4. Pair programming is an example of CL that has been used in CS classes for decades and encouraged for use in AP CSP. In pair programming, two students work together at one computer. One student is the driver and types the code; the other is the navigator and reviews and suggests as the program is being developed. The roles change after a specified time. This requires 100% active involvement.

This paper is organized as follows. Section 2 overviews the specific project details concerning teacher recruitment and professional development (PD) during our multi-year project. Section 3 provides details of our research methods, followed by a summary of our project evaluation results in Section 4. We offer our project observations and lessons learned in Section 5 and concluding remarks in Section 6.

2 Project Background

Students with less computing experience are likely to have greater anxiety when using a computer. Such anxiety is related to lower achievement in computation-based activities and leads to lower competence/self-efficacy. Female students and underrepresented minority (URM) students have historically had fewer computational experiences (e.g., [9, 10]), and therefore greater anxiety and lower self-efficacy regarding computational topics. Experiential anxiety is also related to experience and familiarity with computational thinking. Busch [11] found that students with lower computational self-efficacy (i.e., URM students), when given the choice, selected cooperative learning techniques. In addition, cooperative groups were found to score significantly better on achievement tests, self-concept, and sense of control-mastery (i.e., self-efficacy) than groups receiving direct instruction [12]. A study by Ericson and Guzdial [13] performed an aggregated analysis of gender and ethnicity. It is critically important to consider if CL structures within AP CSP offer benefits to diverse groups of students, leading to outcomes such as greater self-efficacy, higher achievement, and broadened participation in CS. Given these research findings, we expect:

Hypothesis 1: The impact of using CL structures will be cumulative; use of CL structures will increase students' computing self-efficacy, particularly for URM and female students.

Hypothesis 2: More opportunities to learn collaboratively will support stronger learning outcomes (as measured by AP scores).

Hypothesis 3: Students' achievement in AP CSP classes that used CL structures will exceed those of the national AP sample, particularly for females and URM students.

The implementation of teacher PD, teachers' use of CL structures, and student outcomes were externally evaluated using a goals-based approach. Research was conducted on the extent to which use of CL structures impacted student efficacy and achievement.

2.1 Teacher Professional Development

Three cohorts of AP CSP teachers were offered PD in using CL structures as classroom pedagogy, with activities to support student learning while focusing on strategies that promote positive interdependence, individual accountability, equal participation, and simultaneous interaction. From our grant support, we offered week-long summer PD (with travel support, stipend, and academic year followup) to three separate cohorts spanning July 2015 (at the CSTA conference in Dallas, Texas), 2016, and 2017 (both at CSPdWeek in Golden, Colorado), with a total of 143 teachers participating. Figure 1 shows the location of teachers selected across the US from the three cohort groups.

Teachers' quality ratings for the workshop on a scale of 1 to 5 averaged 4.4 in year 1, 4.5 in year 2 and 4.7 in year 3, indicating that in all three years teachers found the workshop to be of high quality. Moreover, teachers made significant pre- to post-PD gains in understanding CL structures, confidence, and perceived preparedness to teach the AP CSP course in all three years.

Figure 1. Location of Project Teacher Participants Cohort1 (Blue), Cohort2 (Red), Cohort3 (Yellow)

The PD training evolved in effectiveness from years 1 to 3, and workshop content increasingly integrated CL structures within the course content of AP CSP; those teachers in Cohort 1 (July 2015) were offering CSP in their schools a year before the official AP CSP exam was available. In open-ended survey responses to items asking about the best aspects of the PD, teachers often discussed 1) the energy, experience and knowledge of the instructors; 2) having CL modeled for them, thereby giving them opportunities to practice and participate in CL themselves; and 3) networking with other CS teachers and learning alongside them.

3 Research Methods Overview

To address the research questions, we utilized data collected via teacher reports of CL practices used during course implementations, student-reported pre/post-course computing self-efficacy surveys, teacher-reported student enrollment data and demographics, and teacher-reported student AP scores provided by the College Board. Measures included:

- A teacher pre-PD background information survey
- Teacher surveys of the frequency and average length of time each CL structure was used in class
- A 15-item measure (5-point Likert scale) of student computing self-efficacy adapted from [14]
- Classroom-level aggregate student enrollment and grade results, including gender and race/ethnicity
- Individual student AP scores (scale of 1-5)
- National AP data available publicly

3.1 Study Sites and Data Collection

Our research study aimed to include classrooms led by 30 cohort 2 and 3 teachers who completed the PD the preceding summer. The AP CSP exam was first given in the 2016-17 school year; therefore, cohort 1 teachers and their students are not included in the present study. With each cohort, in the spring prior to the PD, we contacted the teachers and asked them to participate in our research and to facilitate their students' participation by distributing and collecting consent/assent forms, administering the pre- and post-course student surveys, and submitting student enrollment data, course grades, and AP CSP exam scores. Teachers who agreed to participate and facilitate the participation of their students were then asked to solicit and document school-level or

district-level permission for the research. We included in the study those teachers for whom student participation in the research was approved by an administrator. Ultimately, our study sample was comprised of 14 Cohort 2 and 13 Cohort 3 teachers from 27 high schools across 14 states. The study's 27 schools include 22 public schools, 3 charter or magnet schools, a technology center, and a homeschool network. The analysis sample is 613 students, the majority of whom are male (58%) and/or non-URM (61%). All students assented to the research and if less than 18 years old, had a parent/guardian give permission.

All 27 teachers in this study completed a pre-PD survey of background information in June-July 2016 (Cohort 2) or June-July 2017 (Cohort 3), a mid-course classroom implementation survey in January 2017 (Cohort 2) or January 2018 (Cohort 3) and a final implementation survey the following May. Students completed a pre-course survey in September 2016 (Cohort 2) or September 2017 (Cohort 3) and a post-course survey the following May. Of the 253 Cohort 2 students, 154 (61%) completed the post-course survey; 223 (62%) of the 361 Cohort 3 students completed the post-course survey. AP scores were reported by teachers for 230 (91%) Cohort 2 students and for 240 (67%) Cohort 3 students. National AP data from each May exam administration were drawn from publically available AP data the following November.

3.2 Analytical Procedures

We explored our first hypothesis related to whether students' gains in computing self-efficacy were associated with the use of CL structures. The first step was to calculate opportunity to learn collaboratively (OTLC) scores for each participating classroom for three time periods based on teacher survey data: (1) the first half of the course, named OTLC1, (2) the second half of the course, named OTLC2, and (3) the entire course, named OTLCTot. Scores for use of pair programming for the same three time periods were also calculated. Total pre- and post-course computing self-efficacy scores were determined based on student survey data. Finally, student gender and URM status were variables drawn from student surveys.

We measured the extent to which a student had opportunities to engage with CL structures (of which pair programming was one) by summing across CL structures the product of how often a structure was used (0=not at all, 1=once or twice during this time period, 2=once a month, 3=a few times a month, 4=once a week, 5= every day or a few times a week) with the average length of time per use (0=not at all, 1=1-5 minutes, 2=6-10 minutes, 3=11-15 minutes, 4=more than 30 minutes). We use the term "opportunity to learn cooperatively" so as to make no assumptions about the extent to which a student engages in a particular cooperative learning activity, but to indicate the degree to which the opportunities to engage were available. Therefore:

$$\text{OTLC} = \Sigma (\text{frequency of use}) \times (\text{avg length of use})$$

To control for the number of CLs queried in a given time period (e.g., surveys did not ask about the *exact same set* of CLs each time), we divided each sum by the number of CLs surveyed.

We calculated total self-efficacy as the sum of the 15 computing self-efficacy items given on the pre- and post-course student surveys. Student URM status was coded as '1' where a student indicated race as African American or American Indian/Native American, or ethnicity as Hispanic/Latino. If race/ethnicity was indicated but did not include these categories, URM status was coded as '0'.

To test the first hypothesis, two linear regression models were constructed with class-level OTLCs (and pair programming only) as the independent variables: model 1 with gain in computer self-efficacy as the dependent variable and model 2 with post-course self-efficacy as the dependent variable.

We explored our second hypothesis related to whether students' CS learning (as measured by AP scores) was associated with OTLC. A mediating variable was each teacher's number of years teaching high school computer science. Our belief was that teachers newer to teaching high school CS could benefit more from CL structures since their pedagogical skills were earlier on the developmental curve. The dependent variable was student AP score; independent variables (at the class level) included OTLC1, OTLC2, and OTLCTot. The mediating variable, number of years teaching high school (HS) CS, was used to divide the sample into different analysis groups.

To test this hypothesis, three linear regression models were run with class-level OTLCs as independent variables. The models conditioned AP scores on independent variable combinations for three different samples: (1) students of all teachers, (2) students of teachers with 0-2 years of experience teaching HS CS, and (3) students of teachers with 3+ years of experience teaching HS CS.

Our third hypothesis explored the impact of CL structures in comparison with national AP CSP exam results, disaggregated by gender and by race/ethnicity. We aggregated the separate race/ethnicity distributions into a URM and a non-URM distribution (see above definitions). To test whether each CL distribution was significantly different than the corresponding national distribution, chi-square tests were run for: total group, female students, male students, URM students, and non-URM students. For distributions that were significantly different, group passing rates (a score of 3 or more on the AP exam) were compared and percentage differences were calculated.

4 Results from our Project Evaluation

4.1 Computing Self-Efficacy and OTLC

The 15-item computing self-efficacy measure exhibited excellent post-course reliabilities for both cohorts (Cohort 2 $\alpha=.92$; Cohort 3 $\alpha=.83$). Computing self-efficacy scores were initially higher for non-URM students (Cohort 2 $p<.05$; Cohort 3, $p<.02$) compared with URM students, and for males (Cohort 2, $p<.05$; Cohort 3, $p=.001$) compared with females. Pre-post gains in self-efficacy were not significant for either cohort. In fact, Cohort 2 students' average pre/post-course self-efficacy scores were identical (3.1 on a 5-point scale). The gain in self-efficacy score was non-significant for Cohort 3 students; however, gains for

Cohort 3 non-URM students ($n=116$) were significantly higher than those of URM students ($n=31$; $p<.01$, with a small effect size of .07). Gains in computing self-efficacy did not differ by gender. Linear regressions of self-efficacy gains and post-course self-efficacy scores were conditioned on OTLC scores. For both cohorts, OTLC was not a significant predictor of gains in computing self-efficacy scores or post-course self-efficacy scores.

4.2 Student Learning and OTLC

For the combined student sample (cohorts 2 and 3), passing rates were higher than expected, based on total sample, for non-URM students ($X^2_1, p<.05$), lower than expected for URM students ($X^2_1, p<.01$), and lower than expected for female students ($X^2_1, p<.05$). However, there were no group-level passing rate differences for Cohort 2 alone, and no differences in passing rates by gender for Cohort 3 alone. AP scores were significantly and positively associated with students' initial self-efficacy scores ($p<.01$) and with gains in computing self-efficacy ($p=.000$).

We studied the effect of CLs on student learning (measured by AP scores) for the same combined cohort sample with promising results. Use of CLs positively predicted AP scores. OTLC1 was a significant and positive predictor of AP scores ($n=467, p=.000, R^2=.04$); OTLC2 also positively predicted AP scores ($n=400, p=.03, R^2=.01$), as well as OTLCTot ($n=400, p=.02, R^2=.01$).

Student AP scores are positively predicted by teachers' years of teaching HS CS ($n=23$ teachers, $n=469$ students, $p=.000, R^2=.10$). Interestingly, when the teaching sample was restricted to teachers with less than three-years experience teaching high school CS ($n=11$ teachers), use of CLs was a stronger positive predictor: OTLC1 ($n=234, p=.000, R^2=.13$), OTLC2 ($n=188, p=.000, R^2=.09$) and OTLCTot ($n=188, p=.000, R^2=.09$).

We also considered the impact on AP scores when using just the pair programming CL structure. The use of pair programming in the first half of the course significantly (and positively) predicted AP scores ($n=467, p=.000, R^2=.05$), particularly for students of teachers with less than three years of experience teaching high school CS ($n=234, p=.000, R^2=.09$). The relatively stronger effect of OTLC (and particularly pair programming early in the course) in the hands of newer teachers is particularly promising, given the immediate need for successful pedagogical approaches in the early years of teaching CS.

4.3 Comparison with National Sample

AP scores of students in CL classes were compared with the AP National (AP-N) samples for 2017 and 2018. The combined analytic sample for Cooperative Learning ($n=469$) is 34.5% female ($n=162$), 57.4% male ($n=269$), 36.0% URM ($n=169$), and 55.9% non-URM ($n=262$). The AP-N sample ($n=114,644$) is 31.0% female ($n=35,492$), 69.0% male ($n=79,152$), 31.0% URM ($n=35,561$), and 66.5% non-URM ($n=76,222$).

Cohorts 2 and 3 AP score results were compared with the combined AP National samples from 2017 and 2018. Figure 2 shows the percentages of students who received each AP score for the CL and

AP-N samples by and across student groups. The passing rate (score of 3 or above) for all students in the CL sample, 76.8%, was significantly higher than the AP-N sample, 72.3% ($X^2_1, p<.05$). The passing rate for URM students in the CL sample, 66.3%, was significantly higher than the AP-N sample, 55.0% ($X^2_1, p<.01$) – a difference of over 11 percentage points! The passing rate for CL male students, 79.9%, was significantly higher than male students in the AP-N sample, 73.7% ($X^2_1, p<.05$) – a difference of over 6 percentage points. Finally, the passing rates for female students and for non-URM students were not different for the two samples (female CL= 69.1% vs. 69.2% for AP-N; non-URM CL=82.4% vs. 80.6% for AP-N).

Figure 2. AP Scores for Cohorts 2+3 Cooperative Learning and 2017+2018 AP National Samples

5 Discussion and Lessons Learned

Drawing on previous research, we set out to prepare three cohorts of AP CSP teachers to infuse CL techniques into their courses and then examine the results of our efforts. Specifically, we hypothesized that opportunity to learn collaboratively would have a positive and cumulative impact on students' computing self-efficacy, particularly for URM and female students. Additionally, we anticipated that opportunities to learn collaboratively would positively impact students' AP scores, and that our sample of students learning with CL would have higher AP CSP exam scores than those of the national sample, particularly for females and URM students.

As anticipated, baseline assessments of students' computing self-efficacy indicated lower self-efficacy for URM than non-URM students, and for female than male students. Unexpectedly, gains from pre-post gains were non-significant in both cohorts 2 and 3, and opportunities to learn collaboratively predicted neither pre-post

gains in self-efficacy nor post-course self-efficacy. In Cohort 3, non-URM students gained more than URM students between pre- and post-measures of self-efficacy. The lack of apparent association between OTLC and computing self-efficacy was surprising and may be an area of future research in CS courses.

When examining AP CSP exam scores, we found that students' scores, like scores on our measure of computing self-efficacy, were lower for URM than non-URM students, and for female than male students. Furthermore, these two measures were correlated: computing self-efficacy, and gains in self-efficacy from pre- to post-course, positively predicted student AP exam scores. As expected, students with more opportunities to learn collaboratively, both overall and using pair programming in particular, tended to have higher AP CSP scores. The positive impact of CL appeared most substantial for teachers with fewer than three years of CS teaching experience, a promising and important finding given that newer teachers do not yet have the same pedagogical experience and expertise than veteran teachers.

When we compared students in our research Cohorts 2 and 3 to the national cohort of AP CSP exam takers in 2017 and 2018, we see that the CL study students had passing rates (i.e., scoring 3 or higher) above or equal to the national sample in *all* groups of interest (female, male, URM, non-URM, and total). This is a promising finding, made more so when looking at particular passing rates for these groups: while 55% of URM students in the national sample passed, 66% did in our study, and while 74% of the male students in the national sample passed, 80% did in our study. Furthermore, all students in our combined cohort ($n=469$) outperformed the extremely large national AP sample ($n=144,644$) by 5 percentage points. Broadly, these findings may suggest that CL can particularly be advantageous to URM students and helpful to male students, while having a potentially positive impact on female students (the CL sample had a larger % of female students, when compared with the AP-N sample; this larger female sample did as well as the national sample).

5.1 Limitations and Threats to Validity

We note some limitations of the present study. In this research, we engaged in numerous and varied data collection activities with the participating teachers, and to a lesser extent, their students, in order to understand the effects of the PD on teachers; whether, how, and to what extent the teachers implemented CL in their AP CSP classrooms; and whether and to what extent the CL activities had an impact on student self-efficacy and AP exam scores. In this study, however, we know little about the students in the national sample and their AP CSP teachers - whose outcomes constitute the counterfactual in this research. For example, we are unaware of the extent to which the nation's AP CSP teachers are trained in CL techniques, and the extent to which they may also use CL in their CSP classrooms - an important point when we are examining their average exam scores compared to the CL study sample. We also must consider that in addition to CL, the summer workshops included content on the AP CSP exam more generally. In this study, we did not examine in a national sample the extent to which AP CSP teachers are formally trained on and understand, for example, the

AP CSP curriculum framework and how the exam is structured and scored - all of which may impact their students' exam scores. Finally, we acknowledge that the teachers who self-selected into our PD opportunity are likely different to some extent from their peers across the country, and that these differences may drive some or even all of the findings comparing the CL study sample to the national sample - in essence, a threat to internal validity. These limitations could be addressed in future research with an experimental or quasi-experimental design in which both study groups (research and comparison/control) are fully engaged with data collection activities.

6 Conclusion

Cooperative learning has been studied deeply in many areas, including STEM subjects, but little is known about its application in the context of CS courses. This paper described our investigation into the best practices that lead to the creation and evaluation of PD that infuses the ideas of CL structures within the AP CSP context. Although student opportunities to learn collaboratively (OTLC) was generally not a significant predictor of gains in computing self-efficacy scores or final (post-course) self-efficacy scores, student learning in CL classrooms (as measured by AP CSP scores) demonstrated positive results when compared to national scores. This is particularly notable for female students and URM students. The use of pair programming in the early part of the course significantly and positively predicted AP scores, particularly for students of teachers with less than three years of experience teaching high school. This is encouraging, given the immediate need for successful pedagogical approaches in the early years of teaching CS.

One of the goals of this project is to create a publicly available collection of resources (e.g., workbook and accompanying videos) that document the best practices of applying cooperative learning structures within the AP CSP context. To that end, a subset of 22 teachers from Cohorts 2 and 3 met for two days in July 2018 to share AP CSP lessons that integrate CL structures designed by the teachers for use in their own AP CSP classes. Each of the participating teachers had the responsibility of documenting lessons using a predetermined template that specifies the AP CSP learning objectives targeted in the lesson, as well as the CL structures used. The lessons are documented such that they could be reproduced in any AP CSP class. In addition, video clips of current AP CSP teachers demonstrating selected CL structures in their AP CSP classes are available (<http://bit.ly/apcsp-cl>).

As future work, we plan to investigate the impact of CL structures in other curricula, such as AP CSA. Additionally, CL has been investigated in the context of college courses [15], but we are not aware of how CL can be extended or modified for use in college CS courses in large lecture groups (particularly when the seating arrangement is fixed and does not support the arrangement of students into groups). Other interesting areas for further investigation include online tools for supporting distance learning of CS courses that are taught with CL structures, where the participants may be in different geographical locations.

ACKNOWLEDGMENTS

We thank Lien Diaz for her support in encouraging us to consider the application of cooperative learning to CS Principles. We also thank the CSTA and our colleagues at CSPdWeek in Golden, Colorado for helping to host our summer workshops. This project was funded in part by NSF STEM-C award #1441045.

REFERENCES

- [1] Gillies, R. M. (2016). Cooperative Learning: Review of Research and Practice. *Australian Journal of Teacher Education*, 41(3).
- [2] Dillenbourg, P. *Collaborative Learning: Cognitive and Computational Approaches*. Elsevier Science / Pergamon, 1999.
- [3] Slavin, R. E. *Cooperative Learning: Student Teams, What Research Says to the Teacher*. Second Edition. NEA publishing, West Haven Connecticut 1987.
- [4] Kagan, S. Kagan, M., *Kagan Cooperative Learning*. Kagan Publishing, 2015.
- [5] Garvin-Doxas, K., Barker, L. Communication in computer science classrooms: understanding defensive climates as a means of creating supportive behaviors. *J. Educ. Resour. Comput.* 4, 1, Article 2 (March 2004).
- [6] Johnson, D., Johnson, R., Holubec, E. *The New Circles of Learning: Cooperation in the Classroom and School*. ACSD, Alexandria, VA. 1994.
- [7] Frey, N. Fisher, D., Everlove, S. *Productive Group Work: How to Engage Students, Build TeamWork, and Promote Understanding*. ACSD, Alexandria, VA. 2009.
- [8] Himmele, W., Himmele, P., How to Know What Students Know: Total participation techniques get all students engaged with content, giving teachers feedback about what they know. Educational Leadership. ASCD. 2012.
- [9] Margolis J, Estrella, R., Goode, J., Holme, J., Nao, K. *Stuck in the shallow end: education race and computing*. MIT press, Cambridge, MA, 2008.
- [10] Margolis, J. & Fisher, A. (2003). *Unlocking the clubhouse: Women in computing*. Cambridge, MA: MIT Press.
- [11] Busch, T. (1996). Gender, group composition, cooperation, and self-efficacy in computer studies. In *Journal of Educational Computing Research*. Vol. 15, No.2. 125-135.
- [12] McInerney, V., McInerney, D., & Marsh, H. (1997). Effects of metacognitive strategy training within a cooperative group learning context on computer achievement and anxiety: An aptitude-treatment interaction study. *Journal of Educational Psychology*, Vol 89(4), Dec 1997, 686-695.
- [13] Ericson, B., Guzdial M., Measuring Demographics and Performance in Computer Science Education at a Nationwide Scale Using AP CS Data, In *Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14)*. 2014.
- [14] Compeau, D.R & Higgins, C.A. Computer Self-Efficacy: Development of a Measure and Initial Test. *MIS Quarterly*, Vol. 19 (2): 189-211, 1995.
- [15] Premo, J., Cavagnetto, A., Davis, W., Promoting Collaborative Classrooms: The Impacts of Interdependent Cooperative Learning on Undergraduate Interactions and Achievement. May 2018. *CBE life sciences education* 17(2).