Long Term Effects of Pair Programming

Max Smith, *Student Member, IEEE*, Andrew Giugliano, and Andrew DeOrio, *Senior Member, IEEE*

***Abstract*—CONTRIBUTION: This study provides evidence for the benefits of pair programming early in the curriculum on student performance later in the curriculum. It also confirms the short term benefits of partnerships at scale.**

BACKGROUND: Engineers often work in teams, both in industry and in academia. Previous work has shown that partnered programming yields higher student performance during the course in which students partner.

RESEARCH QUESTIONS: This study investigates the long term effects of early curriculum pair programming on student performance in a following course. Specifically, do student partnerships impact long-term student performance in a later course? Are previously observed effects of partner programming robust to a larger sample size?

METHODOLOGY: This quantitative analysis examines 2,468 students in an introductory computer science sequence at a large, public research institution. The data set comprises two academic years and includes partnership participation, project and exam scores, withdraw rates, time between courses, GPA, and gender.

FINDINGS: A positive relationship is observed between partnering in an introductory course, and higher project scores in a future course where all students worked alone (N=1,003). Students with the lowest GPAs experienced the greatest benefits. Additionally, results with a large population of students confirm the observations of previous research, showing that partnerships are associated with an overall positive grade impact during the course in which the partnership takes place (N=2,468).

***Index Terms*—Computer science education, observations, pair programming, quantitative, teamwork, undergraduate**

I. Introduction

Engineering is a collaborative activity, but students often work alone in introductory computing courses [1]. Specifically, engineering students in lower-level programming courses often complete programming assignments independently. Recent interest in pair programming bridges the gap between engineering practice and engineering education, introducing students to programming in groups early in their education. Pair programming is a technique where two people share a single workstation, collaboratively working on the same problem.

In previous studies in the literature, working in a partnership was associated with higher project scores [2]–[5] during the semester that the partnership took place. In particular, students with lower confidence and lower test scores benefited the most from working in a partnership [6], [7]. Several experiments showed increased retention rates [8]–[11], addressing a major problem afflicting introductory computing courses. Furthermore, a meta-analysis of additional literature suggests that, when implemented correctly, pair programming has a positive effect on exam scores and attitudes toward programming [4]. Partnerships also reduced the reliance on teaching staff [5].

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor {mxsmith@umich.edu, agiuglia@umich.edu, awdeorio@umich.edu}.

These previous studies examined performance in the first computing courses taken by students. The effects of partnerships on students' performance has also been studied in a second programming course in an introductory three-course sequence [3], [12], [13]. These three courses are commonly referred to as CS1, CS2 and CS3 in the literature; a usage adopted here. Multiple studies have found higher project scores [3], [12], [13]. Some studies have observed higher exam scores [12], [13], while others found lower exam scores [3]. One possible reason for these different results is that the course with lower exam scores [12] was structured with only three assignments, giving students less opportunity to pair program. Additionally, students working in pairs have more favorable experiences than those who worked individually [12], [13].

Recent work has begun to explore the effects of partnership on future performance. Two of the authors of this paper, Giugliano and DeOrio [3], examined the impact of partnerships in CS2 on student performance in CS3. While they were able to draw conclusions about specific GPA groups, they were not able to draw conclusions about the CS3 general population. The study presented here expands that work by looking at the effects over a period of time twice as long, resulting in over twice as many data points. After filtering, their data set contained 1,032 CS2 student records and 507 CS3 student records. This study after filtering, examines 2,468 CS2 student records and 1,003 CS3 student records, and employs stronger statistical methods, using multivariate ANOVA compared to the previous work using a student's t-test. Results from this new assessment data include new findings, and different analyses of data compared to the previous paper.

Williams *et al.* [14] and Cockburn *et al.* [15] studied pair programming in the software engineering industry. Paired programmers experienced more happiness and confidence in their work than their peers who worked alone. Software engineers produced higher quality code in less elapsed time, but in marginally more programming hours (15% increase) [16]. Hannay *et al.* [17] also found that production time was reduced when pair programming; however, they found that higher quality code was only present when the task complexity was high. The same study showed that lower quality code was produced for simpler tasks.

This paper evaluates concerns that pair programming may impact future student performance. It examines how pair programming in a CS2 course affects performance in both CS2 and CS3, accounting for factors such as GPA and gender. "Long term" is defined in this study as the next course in the computer science curriculum. Subsequent courses after CS3 no longer follow a standard curriculum, because the students are taking a selection of advanced computer science electives. Specifically, it evaluates the questions:

- Do student partnerships in CS2 impact long-term student

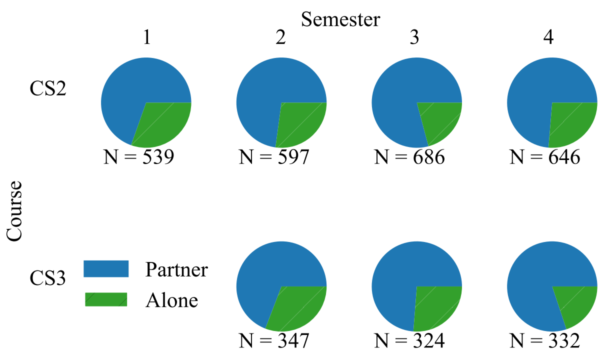

Fig. 1. **Data set overview: two courses over four semesters**, shown with the proportion of students who partnered in CS2. Approximately 75% of students worked in partnerships during CS2. All students worked alone in CS3, and the figure shows the proportion of CS3 students who partnered in their previous CS2 course. N represents the number of records after filtering (see Section II-A4), with a total of 2,468 records from CS2 and 1,003 records from CS3.

performance in a later CS3 course? Are gender and GPA demographics affected differently?
- Are previously-observed effects of partner programming robust to a larger sample size?

The contributions of this work include an examination of the effect of partnerships early in the curriculum on student performance later in the curriculum. To the best of the authors' knowledge, this the largest single study of pair programming and the first to successfully evaluate its long-term impact on the general student population.

II. Methodology

This paper uses a quantitative analysis to compare the performance of students working alone to that of those working in a partnership. The data set comes from the second and third courses (CS2 and CS3) in a three-course sequence (CS1–3), offered on the second and third semesters. This section discusses the data set, the two courses, the variables and the statistical methods.

A. Data

A quantitative analysis was performed on student partnership data by analyzing its effects on student project and exam scores in both CS2 and CS3. These scores are used to measure the association with factors such as partnership, GPA, and gender. Fig. 1 shows a summary of the data set. This section describes the specific details of the CS1–3 sequence analyzed in this study.

1) Curriculum Overview: The introductory programming sequence at the large, highly ranked public research institution in this study consists of three courses. The first course (CS1) is offered in three different versions: for engineering majors, for accelerated students, and for non-engineering majors. CS1 introduces students to programming for the first time, and teaches the fundamentals: control flow, selection and iteration, and basic object-oriented programming. CS1 is taught in C++ and Python. Students from all versions of CS1 then take CS2 together, where they learn additional programming paradigms and begin studying data structures. Finally, students take CS3, where they study data structures and advanced algorithms in depth. The entire course sequence (CS1–CS3) is a requirement for computer science majors and minors, but also includes a diverse group of students from many majors, especially across engineering disciplines. Many engineering students complete a computer science minor, because it is helpful in completing their respective coursework.

The instructors in each course offering of CS1, CS2 and CS3 varied. Instructors coordinated between sections each semester, striving to provide a uniform experience for students in different sections of the same course. All courses were taught in the traditional classroom method, with an emphasis on in-class exercises.

2) CS2 Course Description: In CS2 "Programming and Introductory Data Structures", students complete five programming projects and two exams. The first project is completed alone, and is a student readiness check. For the remaining four projects, students have the option to select a partner. Project 1 was worth 4% of the students' course grade and the remaining projects are each worth 9%. Project 1 verified that students had the prerequisite knowledge for the course. The midterm and final are worth 25% and 29% respectively. The remaining 6% of the course grade constitutes lab exercises and course surveys. Students attend two 90-minute lectures and a two-hour lab weekly for approximately fourteen weeks. CS2 covers three topic categories: functional abstraction, data abstraction, and dynamic resource management. Projects 2-5 focus on: recursion, abstract data types, dynamic resource management, and polymorphism. Students complete assignments in C++ using the development tools of their choice. Lab exercises are completed using a combination of the student's preferred development environment, and the Labster interactive program visualization system [18].

Students who chose to partner in CS2 were required to use pair programming best practices, where both students work on the same problem on a single computer [4]. Pair programming guidelines were outlined in the course syllabus and encouraged in lecture [4]. Each partnership submitted one project solution for the pair and both partners received the same grade. In between projects, students had the option to change partner, or to resume working alone. Fig. 2 shows the distribution of CS2 project and exam scores.

3) CS3 Course Description: CS3 "Data Structures and Algorithms," focuses on composing foundational programming components into computational tools, such as algorithms and data structures, to solve more complex problems. The course textbooks is [19]. CS3 contains four projects and two exams. Each project contributes 10% to the final course grade, and all are equally challenging. The two exams each count for 25% of the final grade, with the remaining 10% coming from small problem sets. Fig. 3 shows the distribution of CS3 project and exam scores. Students complete all course content in CS3 independently. CS3 introduces students to a number of fundamental techniques to solve programming problems including: complexity analysis, container data structures, hash-

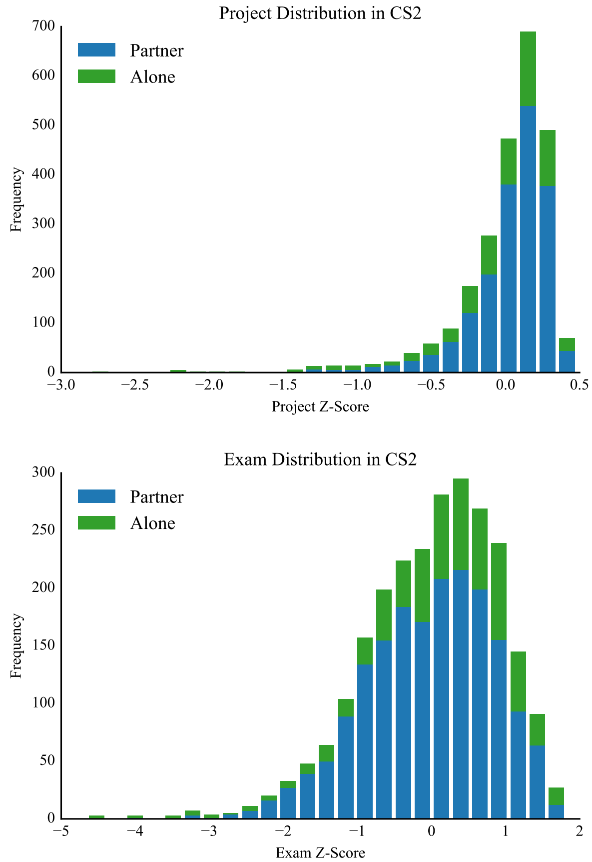

Fig. 2. **CS2 total project and exam score distributions**, represented as Z-scores. Stacked bars indicate students who worked alone and those who worked in partnerships. (N=2,468)

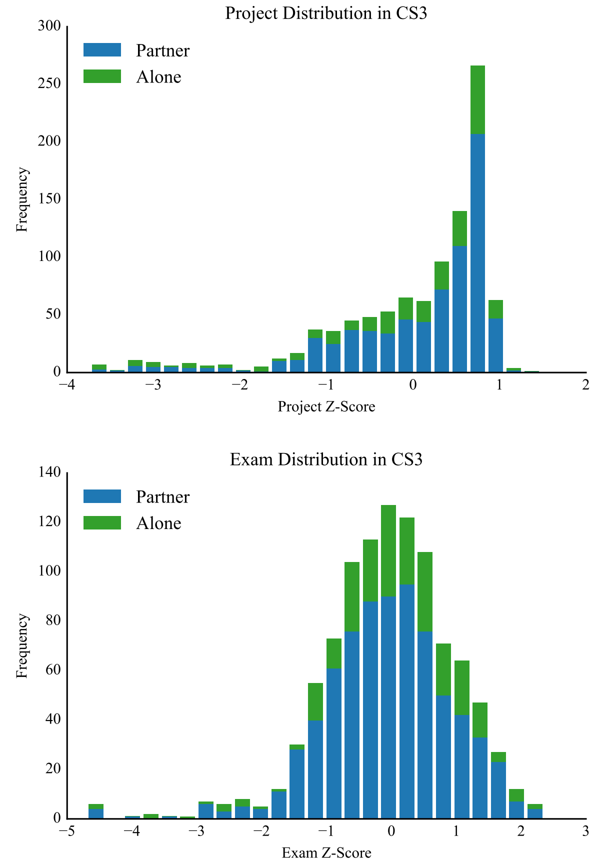

Fig. 3. **CS3 total project and exam score distributions**, represented as Z-scores. All students worked alone in CS3, and stacked bars indicate students who worked alone or in partnerships in their previous CS2 course. (N=1,003)

ing, backtracking, greedy programming, and dynamic programming. Students attend two 90-minute lectures and a one-hour discussion each week. Project topics include searching and storing data, priority queues, graph search algorithms, and complexity analysis. Students completed projects in C++ with the development environment of their choice.

4) Cleaning Data: Students who withdrew from either CS2 or CS3 represent incomplete data, as they may not have finished their coursework prior to withdrawing, so were removed from the dataset. Concerns that partnerships may influence withdrawal in CS3 are addressed in Section III-C, where no statistically significant association was observed.

Students must complete CS2 with a passing grade to enroll in CS3. CS3 grade data was collected from three semesters over two academic years - the same two years for which CS2 grade data was obtained. In CS2, students had the option to partner. Students are not allowed to partner in CS3 for any of their coursework. Fig. 1 shows an overview of the data set.

Duplicate records, from students who took a course more than once, were handled by dropping the previous records, and keeping data from a student's most recent semester. The two data sets were joined using anonymized, unique identifiers. Demographic information came from the university records, and was joined into the data set.

Students who audited the course were removed, as were those reported for cheating, or those who took an incomplete from either course. Removing these data points reduces noise from the students not having the same incentive to perform on coursework. Students who took CS3 without taking CS2 were removed. This ensured that all students being considered had the option to pursue a partnership. The raw data consisted of 2,696 CS2 student records and 1,880 CS3 student records. After cleaning, the data set contained 2,468 CS2 records and 1,003 CS3 records.

B. Statistical Methods

Students were identified by unique, anonymous identifiers. These identifiers were used to follow individual students from CS2 through CS3, and collect all of their score data. Score data was then converted into Z-score data per semester per assignment. Z-scores standardize the distributions of data across semesters, and account for differences such as exams unique to each semester. Z-scores are a way of relating scores to a mean, with a Z-score of 1 meaning a standard deviation above the norm. Before computing Z-scores, each assignment's scores were verified to be normally distributed.

ANOVA was used to analyze the variance between and within the factors (independent variables) in the study. Statistical tests were performed with a 95% confidence interval, and $\alpha = 0.05$.

	Men	Women	Total
Partner (in past CS2)	593	152	745
Alone (in past CS2)	212	46	258
Total	805	198	1003

TABLE I
CS3 student gender distribution, BY PAST CS2 PARTNERSHIP.

C. Factors

First, the dependent variables are discussed, namely, the factors used to measure changes in student performance. The effects of partnerships in CS2 on students is observed through their scores on exams and projects in both CS2 and CS3. The project score metric is calculated as the weighted sum, per the syllabus, of individual project scores, and then converted to a Z-score per project per semester. CS2 exam scores measure if project partnerships affect student performance in their independent work. The CS2 exam score is weighted according to the syllabus and converted into a Z-score by semester.

Long-term effects of CS2 partnerships are measured by CS3 score dependent variables. Again, both project and exam scores are considered. CS3 total project score is composed of the four projects weighted according to the syllabus and then Z-scored. The CS3 exam score is the Z-score of the average of the midterm and final from the course.

A summary of the factors (independent variables) examined includes:

- **Partnership in CS2:** if a student had at least one partner for a minimum of one full project while in CS2. This is represented as a discrete variable with two values.
- **Gender:** retrieved from university records; represents the student's self-reported gender as man or woman.
- **GPA:** the students' cumulative GPA previous to the semester that they took CS2. It is on a 4.0 scale.
- **Time Between CS Courses:** the integer number of academic year semesters between when the student took CS2 and when they took CS3.

Table II-C describes the gender distribution of CS3 students, divided by choice of partnership in their prerequisite CS2 course. Approximately three quarters of students elected to partner, among both men and women. All students worked alone in CS3.

III. Results

This section quantifies the relationship between working in a partnership, and student project and exam scores. It also examines the interaction with GPA, gender, and time between CS courses. First, the long term effects of CS2 partnerships on CS3 is examined. Then, results within the same CS2 semester are analyzed. A graphical summary of results is shown in Fig. 4, which illustrates that students who partnered had, on average, higher net course grades in both CS2 and CS3. More specifically, in CS3 those who partnered in their earlier CS2 course averaged a 0.14 higher CS3 project Z-score and no statistically significant difference in exam Z-score (N=1,003). In CS2, those who partnered averaged a 0.12 lower CS2 exam Z-score and 0.21 higher CS2 project Z-score (N=2,468). The net impact of partnerships on CS2 Z-scores was positive.

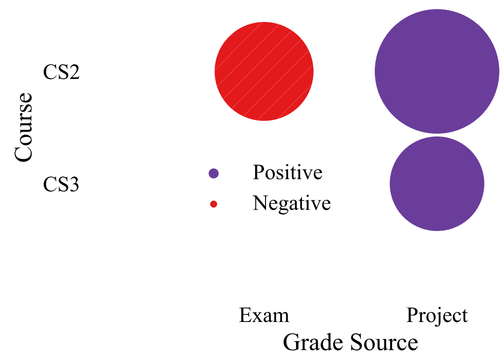

Fig. 4. **Impact overview of partnering on CS2 and CS3** project and exam Z-scores. More specifically, those who partnered in CS2 were associated with higher project Z-scores in both CS2 (while working with a partner) and CS3 (while working alone). Partnerships in CS2 were associated with lower exam Z-scores in CS2, and were not associated with exam Z-scores in CS3. The net impact of partnerships was positive in both courses.

A. CS3 and Partnerships From CS2

This experiment analyzes data from CS3, where the dependent variables are exam and project Z-scores in CS3. The effects of partnerships in CS2 on future student performance in CS3 where students worked alone are examined. These results help answer the first research question: does CS2 partnership effect CS3 performance? First the relationship between partnership and CS3 performance while controlling for GPA is shown. Next, the association of partnerships with gender is examined. Finally, the interaction between the time between CS courses impacts student performance in CS3 while controlling for GPA is analyzed.

First, the relationship between CS2 partnership and CS3 exam Z-scores, while controlling for GPA is examined. This relationship was analyzed through a two-way ANOVA test, and the summary table is shown in Table II. There was no significant difference in exam Z-score for partnership, independent of student GPA. Additionally, it was found that the interaction between partnership and GPA has a significant association with differences in CS3 exam Z-score, with a small effect size.

Next, the same analysis was repeated with a CS3 project Z-score as the dependent variable. In the general population, there was a statistically significant difference in CS3 project Z-score for those who partnered in their past CS2 course, after controlling for GPA. It is noted that the effect size is small, and therefore the strength of these observations should not be overstated.

1) CS3, Gender, and Partnerships From CS2: Next, the effect of CS2 partnerships on gender groups in CS3 is analyzed. Table III shows a summary of an ANOVA test with independent variables gender, CS2 partnership, and GPA. Again, dependent variables are CS3 exam and project Z-scores. This section of the analysis is interested in any associations with the interaction of gender and partnerships. The results did not show any statistically significant associations. However, a

TABLE II

CS3 ANOVA WITH INDEPENDENT VARIABLES PAST CS2 PARTNERSHIP AND GPA. DEPENDENT VARIABLES WERE CS3 AVERAGE EXAM Z-SCORE AND AVERAGE PROJECT Z-SCORE. PARTNERING IN CS2 HAD A STATISTICALLY SIGNIFICANT RELATIONSHIP WITH PROJECT PERFORMANCE IN CS3, AFTER CONTROLLING FOR GPA. (N=1,003) (†: STATISTICALLY SIGNIFICANT DIFFERENCES IN PROJECT Z-SCORES; *: STATISTICALLY SIGNIFICANT DIFFERENCES IN EXAM Z-SCORES)

	CS3 Exams						CS3 Projects					
	df	Sum Sq.	Mean Sq.	F	P(>F)	η_p^2	df	Sum Sq.	Mean Sq.	F	P(>F)	η_p^2
Partner (in past CS2)†	1	0.9057	0.91	1.10	2.937e-01	1.0e-3	1	6.9936	6.9936	8.45	3.74e-03	9.0e-3
GPA*†	1	200.9843	200.98	244.97	2.562e-49	2.0e-1	1	125.2907	125.2907	151.37	2.15e-32	1.4e-1
Partner x GPA*	1	8.0845	0.08	9.85	1.747e-03	1.0e-2	1	2.0534	2.0534	2.48	1.16e-01	3.0e-3
Residual	954	782.7133	0.8205				954	789.6320	0.8277			

TABLE III

CS3 ANOVA WITH INDEPENDENT VARIABLES PAST CS2 PARTNERSHIP, GPA, AND GENDER. DEPENDENT VARIABLES WERE CS3 AVERAGE EXAM Z-SCORE AND AVERAGE PROJECT Z-SCORE. IN PARTICULAR, THE INTERACTION OF PARTNERSHIPS AND GENDER IS EXAMINED. THE INTERACTION OF PARTNERSHIPS IN CS2 AND GENDER WAS NOT ASSOCIATED WITH A STATISTICALLY SIGNIFICANT DIFFERENCE IN CS3 PERFORMANCE. (N=1,003) (†: STATISTICALLY SIGNIFICANT DIFFERENCES IN PROJECT Z-SCORES; *: STATISTICALLY SIGNIFICANT DIFFERENCES IN EXAM Z-SCORES)

	CS3 Exams						CS3 Projects					
	df	Sum Sq.	Mean Sq.	F	P(>F)	η_p^2	df	Sum Sq.	Mean Sq.	F	P(>F)	η_p^2
Partner (in past CS2)†	1	0.9057	0.9057	1.1e+0	2.9e-01	1.2e-3	1	6.99	6.99	8.5	3.6e-03	8.9e-3
Gender*	1	17.9640	17.9640	2.3e+1	2.4e-06	2.3e-2	1	2.21	2.21	2.7	1.0e-01	2.8e-3
GPA*†	1	208.0989	208.0989	2.6e+2	4.7e-52	2.2e-1	1	126.78	126.78	154.0	7.0e-33	1.4e-1
Partner x GPA*	1	7.6064	7.6064	9.5e+0	2.1e-03	9.9e-3	1	1.97	1.97	2.4	1.2e-01	2.5e-3
Partner x Gender	1	0.1332	0.1332	1.7e-1	6.8e-01	1.8e-4	1	1.59	1.59	1.9	1.7e-01	2.0e-3
Gender x GPA	1	0.4645	0.4645	5.8e-1	4.5e-01	6.1e-4	1	1.92	1.92	2.3	1.3e-01	2.4e-3
Partner x GPA x Gender	1	0.0001	0.0001	1.0e-4	9.9e-01	1.3e-7	1	0.45	0.45	0.6	4.6e-01	5.8e-4
Residual	950	757.5250	0.7973				950	782.06	0.82			

TABLE IV

CS3 ANOVA WITH INDEPENDENT VARIABLES TIME BETWEEN CS COURSES AND GPA. DEPENDENT VARIABLES WERE CS3 AVERAGE EXAM Z-SCORE AND AVERAGE PROJECT Z-SCORE. AN ASSOCIATION BETWEEN DECREASED GAP BETWEEN COURSES AND INCREASED CS3 PERFORMANCE WAS OBSERVED. (N=1,003) (†: STATISTICALLY SIGNIFICANT DIFFERENCES IN PROJECT SCORES; *: STATISTICALLY SIGNIFICANT DIFFERENCES IN EXAM Z-SCORES)

	CS3 Exams			CS3 Projects		
	F	P(>F)	η_p^2	F	P(>F)	η_p^2
Time†	2.38	9.33e-02	0.005	8.97	1.39e-04	0.186
GPA*†	252.10	1.66e-50	0.210	148.26	8.66e-32	0.135
Time x GPA*†	7.73	4.67e-04	0.016	4.41	1.24e-02	0.009

gender difference on exams, independent of partnership status or GPA, was observed.

2) Time Between CS2 and CS3: The next analysis examines calendar time between when a student takes CS2 and CS3 as an independent variable. Calendar time is measured by integer semesters between CS2 and CS3. Again, a two-way ANOVA test is used to discern any association between the time-between-courses and performance in CS3 (Table IV), while controlling for GPA. Decreased time between CS courses was associated with increased CS3 project Z-scores, after controlling for GPA.

B. CS2 Partnerships

In this experiment data from CS2 is analyzed. The dependent variables are CS2 project Z-scores and CS2 exam Z-scores. The first analysis tests for a relationship between the independent variables partnerships and CS2 Z-scores while controlling for GPA. Then, the analysis turns to gender groups. These results answer the second research question: are the previously discovered trends on the effects of partnership robust to larger sample sizes?

First, the relationship between working in a partnership and student performance while controlling for GPA is explored. ANOVA tests were performed with the independent variables being partnership status and GPA (Table V). Dependent variables were CS2 exam Z-score and CS2 project Z-score. A statistically significant relationship was observed between partnering in CS2 and both project and exam Z-scores. Additionally, there was an interaction effect of partnerships with GPA. It is noted that while several results were statistically significant, their respective effect sizes were very small.

1) CS2, Gender and Partnerships: Next, the analysis turns to gender differences in CS2 partnerships. The results of an ANOVA test are shown in Table VI. Independent variables were partnership status in CS2, gender and GPA. Dependent variables were CS2 exam Z-score and CS2 project Z-score. In particular, this analysis is interested in the interaction of partnerships with gender, while controlling for GPA. There was a small, statistically significant association with exam Z-scores, and none with project Z-scores.

C. CS3 Course Withdrawal Rate

This section examines CS3 withdrawal rates for any relationship to partnerships in CS2. In the raw data, 76 students completed CS2, started CS3, and then withdrew from CS3

TABLE V
CS2 ANOVA WITH INDEPENDENT VARIABLES CS2 PARTNERSHIP AND GPA. DEPENDENT VARIABLES WERE CS2 AVERAGE EXAM Z-SCORE AND AVERAGE PROJECT Z-SCORE. PARTNERSHIPS WERE ASSOCIATED WITH STUDENT PERFORMANCE ON BOTH PROJECTS (HIGHER) AND EXAMS (LOWER), AFTER CONTROLLING FOR GPA. (N=2,468) (†: STATISTICALLY SIGNIFICANT DIFFERENCES IN PROJECT Z-SCORES; *: STATISTICALLY SIGNIFICANT DIFFERENCES IN EXAM Z-SCORES)

	CS2 Exams						CS2 Projects					
	df	Sum Sq.	Mean Sq.	F	P(>F)	η_p^2	df	Sum Sq.	Mean Sq.	F	P(>F)	η_p^2
Partner*†	1	13.3227	13.3227	22.5	2.3e-006	9.46e-3	1	6.9022	6.9022	83.9	1.1e-019	3.44e-1
GPA*†	1	506.1199	506.12	853.0	2.9e-160	2.66e-1	1	56.06283	56.06	681.5	3.8e-132	2.25e-1
Partner x GPA†	1	1.1483	1.15	1.9	1.6e-001	8.22e-4	1	8.9185	8.92	108.4	7.5e-025	4.41e-2
Residual	2352	1395.4947	0.5933				2352	193.4845	0.0823			

TABLE VI
CS2 ANOVA WITH INDEPENDENT VARIABLES PARTNERSHIP, GPA AND GENDER. DEPENDENT VARIABLES WERE CS2 AVERAGE EXAM Z-SCORE AND AVERAGE PROJECT Z-SCORE. IN PARTICULAR, THIS ANALYSIS IS INTERESTED IN THE INTERACTION OF PARTNERSHIPS AND GENDER. THE INTERACTION OF PARTNERSHIPS AND GENDER WAS ASSOCIATED WITH CS2 EXAM Z-SCORES, BUT NOT PROJECT Z-SCORES. IT IS NOTED THAT THE EFFECT SIZE WAS SMALL. (N=2,468) (†: STATISTICALLY SIGNIFICANT DIFFERENCES IN PROJECT Z-SCORES; *: STATISTICALLY SIGNIFICANT DIFFERENCES IN EXAM Z-SCORES)

	CS2 Exams						CS2 Projects					
	df	Sum Sq.	Mean Sq.	F	P(>F)	η_p^2	df	Sum Sq.	Mean Sq.	F	P(>F)	η_p^2
Partner*†	1	13.32	13.32	23.0	1.8e-006	9.7e-3	1	6.90	6.90	83.9	1.1e-019	3.4e-2
Gender*	1	20.81	20.81	35.9	2.4e-009	1.5e-2	1	0.10	0.10	1.2	2.7e-001	5.2e-4
GPA*†	1	515.22	515.22	888.2	8.2e-166	2.7e-1	1	55.89	55.89	679.3	9.4e-132	2.2e-1
Partner x GPA†	1	1.25	1.25	2.2	1.4e-001	9.2e-4	1	8.96	8.96	108.9	5.8e-025	4.4e-2
Partner x Gender*	1	2.57	2.57	4.4	3.5e-002	1.9e-3	1	0.08	0.08	0.9	3.4e-001	4.1e-4
Gender x GPA	1	0.81	0.81	1.4	2.4e-001	5.9e-4	1	0.02	0.02	0.2	6.4e-001	1.0e-4
Partner x GPA x Gender	1	0.12	0.12	0.2	6.5e-001	8.8e-5	1	0.25	0.25	3.1	7.9e-002	1.3e-3
Residual	2348	1361.98	0.58				2348	193.17	0.08			

TABLE VII
CS3 WITHDRAW RATE χ^2 TEST. THE INDEPENDENT VARIABLE WAS PAST CS2 PARTNERSHIP STATUS, AND THE DEPENDENT VARIABLE HAD TWO LEVELS INDICATING WHETHER A STUDENT COMPLETED CS3 OR WITHDREW. NO STATISTICALLY SIGNIFICANT RELATIONSHIP BETWEEN CS2 PARTNERSHIP STATUS AND CS3 WITHDRAWAL RATES WERE OBSERVED.

	Partnered in CS2	No partnership in CS2
Withdrew from CS3	57	19
Completed CS3	747	273

before the end of the semester. In the previous analyses, these records were omitted. CS3 withdrawal rates were approximately 7%, both among those who partnered in their past CS2 course and those who worked alone. A χ^2-test was performed with past CS2 partnership as the independent variable, and a two-level dependent variable indicating whether a student completed CS3 or withdrew. A χ^2-test was used because it determines whether there is a significant association between two categorical variables, with at least five entries in each cell in the contingency table. The results of the test are shown in Table VII. The χ^2-statistic was 0.0405 with a p-value of 0.8405; therefore, no significant relation between partnerships in CS2 and withdrawal from CS3 was found. The results of this test support the earlier decision to remove students who withdrew from other analyses.

IV. DISCUSSION

Through the analysis of partnerships and student performance, a number of statistically significant relationships were observed. First, the impact on CS3 is discussed, then that on CS2, before the discussion turns to the role of gender in partnerships. Finally, limitations of this study are presented.

A. *CS3 and Partnerships From CS2*

In CS3, there was an association between partnering in a past CS2 course, and project Z-scores while working alone in CS3. Also, there was an interaction between partnerships and GPA on exam Z-scores.

In the CS3 general population, students who partnered in their previous CS2 course were associated with higher average CS3 project Z-scores, after controlling for GPA. Specifically, students who had partnered had a 0.14 higher average project Z-score. To put the Z-score in perspective, this translates to an approximately 2.1% higher final course grade in CS3 (out of 100%). This result also indicates students experienced additional learning opportunities from their partnership, leading to higher performance.

Furthermore, there was an interaction effect between partnership and GPA (independent variables) and CS3 exam Z-scores (dependent variable). To interpret this interaction, GPA was divided into quartiles, and the exam Z-score mean within each GPA quartile was computed. Fig. 5 provides a visualization of project Z-scores by GPA quartile. Students in the lowest

GPA quartile were associated with the greatest gains from their past CS2 partnership. The remaining quartiles experienced very small differences in CS3 exam Z-score. This came out to approximately a 3-4% difference in final grades (out of 100%), for the lowest GPA quartile. This result provides evidence that the students closer to the bottom of the class stand to benefit the most from the learning opportunities created by partnerships.

These results from CS3 provide evidence to support a long term, positive relationship between partnerships and student performance.

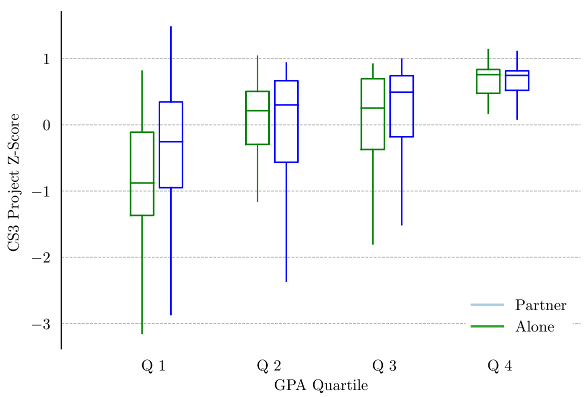

Fig. 5. **CS3 project Z-scores**, divided by CS2 partnership status and GPA quartile. Students in the lowest GPA quartile were associated with the greatest long term gains from past CS2 partnerships. (N=1,003)

B. CS2 and Partnerships

In CS2, partnerships were associated with higher project Z-scores and lower exam Z-scores with a net positive effect on final grades. Furthermore, students in the lowest GPA quartile were associated with the greatest overall gains from partnering.

As a whole, CS2 students in working partnerships had a 0.21 higher average project Z-score and a 0.12 lower average exam Z-score than those who worked alone. At the end of the course, this translated to a 1.2% higher final course grade (out of 100%) for those who partnered. A possible explanation for higher project Z-scores is the fact that partnered students had two people working and thinking on the same problem, which meant they had additional learning opportunities when presented with errors in the design of their solution. However, lower exam Z-scores for partnerships may point to some disadvantages to learning in a partnership. It is also possible that students working alone were a self-selected group that also performed better on exams.

A significant interaction was observed between partnerships and GPA on project Z-scores but not exam Z-scores. The bottom GPA quartile has the largest difference in average exam Z-score, where paired students averaged -0.27 and lone students averaged -0.70. This difference meant partnered students in the bottom GPA quartile had a 2-3% higher final course grade (out of 100%) than those in the same quartile who worked alone. This impact is equivalent to partnered students receiving a final letter grade of B- and lone students a C+.

Why do students in the lowest GPA quartile receive such a large benefit from partnerships? Is the benefit from a strong partner, or additional learning opportunities of the partnership experience? CS3 results, showing a positive association between CS2 partnerships and CS3 project Z-scores, provide evidence of the long term benefits of pair programming.

These results from CS2 confirm the previously observed effects of pair programming are robust to a large sample size.

C. Time Between CS2 and CS3

Students who took CS3 immediately after CS2 were associated with higher project and exam Z-scores. A possible intuitive understanding of this result is that the CS2 material necessary to succeed in CS3 will be fresh in students' minds. A significant interaction was observed between the number of semesters between when the student took CS2 and CS3, and their partnership status in CS2 (N=1,003). Those who had a partnership had an average 3% higher CS3 course grade (out of 100%) when they took it immediately than if they waited a semester. On average, partnered students experienced a statistically significant 1% lower course grade after waiting a semester (out of 100%), compared to peers who worked alone.

D. Gender

The discussion now turns to results on the interaction of gender and partnerships while controlling for GPA.

When examining CS3 project and exam Z-scores, no statistically significant interactions involving gender were observed (N=1,003). Gender was independently associated with exam Z-scores. Similarly, associations with CS2 project Z-scores were not statistically significant.

CS2 exam Z-scores showed a statistically significant larger difference between women who worked alone and men who worked alone (N=2,468). Specifically, women working alone averaged a 0.32 higher exam Z-score than partnered women. For comparison, men working alone averaged a 0.14 higher Z-score than partnered men. It is noted that despite gender-partnership interaction being associated with differing exam Z-scores, the strength is small. Its impact is weaker than GPA, and the tests all featured a large residual.

E. Limitations

This study was observational in design, rather than experimental. This led to a number of uncontrolled variables.

First, because students had the choice to partner in CS2 there is no way to ensure that the population of partnered and lone students is equal. For example, stronger students may have tended to work alone so they could complete their work free from the overhead of communication with a partner. Alternatively, stronger students may have chosen to work in partnerships. Furthermore, students selected their partners.

Second, there was no data describing the day-to-day practices of a partnership. While students were encouraged to use pair programming best practices in lecture and in the syllabus, partnership dynamics were not controlled.

The data set is compiled from multiple semesters of the same course. The course syllabus remained stable over the duration of the study, but there were some differences from semester to semester. Not all semesters shared the same set of instructors. Exams covered the same material, but the exact questions were unique to each semester.

Finally, while several observations were statistically significant, many residuals were also large in the statistical analysis. This limits the strength of the conclusions that can be drawn from these analyses.

V. Conclusions and Future Work

This paper has examined two research questions involving pair programming using a quantitative approach. First, it investigated the effect of CS2 partnerships on CS3 performance, finding that students who participated in partnerships during their CS2 course were associated with higher CS3 project grades. Furthermore, evidence was presented that students with lower GPAs stood to benefit the most from the learning opportunities created by partnerships.

The results confirmed the observations of previous studies of pair programming. Similar to prior work, this study found that partnered students in CS2 were associated with higher project Z-scores. These results agree with some prior work and disagree with others in its finding that partnerships in CS2 were also associated with lower CS2 exam Z-scores. In the end, the benefit from project Z-scores outweighed exam Z-scores, resulting in a net positive effect on students' final grades in CS2.

When investigating the effect of partnerships on different demographics, there were some discrepancies. Women students who partnered scored slightly lower on exams in CS2. Additionally, partnered students with the lowest GPAs had higher CS2 and CS3 project Z-scores compared to peers in the same GPA quartile who did not partner.

The results in this study suggest a number of future directions. A study should explore any partnership selection bias in the performance of students. A follow-up qualitative study would investigate how students benefit from pair programming, and why there is a negative association with CS2 exam Z-scores. A study of the gender dynamics within pair programming and its impact on long term student performance would help explain the gender differences among student partnerships. Additionally, future work might explore the factors that contribute to a successful programming partnership.

Practitioners of computer science education can leverage these results to include partnerships in introductory courses. The presented evidence can alleviate concerns that partnerships may have a negative long term effect on students. Overall, the results provide evidence to support a long term, positive relationship between partnerships and student performance.

References

[1] C. D. Hundhausen, A. S. Carter, and O. Adesope, "Supporting programming assignments with activity streams: An empirical study," *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 2015.

[2] C. McDowell, L. Werner, H. Bullock, and J. Fernald, "The effects of pair-programming on performance in an introductory programming course," *Proc. SIGCSE*, pp. 38–42, 2002.

[3] A. Giugliano and A. DeOrio, "Long term effects of partner programming in an introductory computer science sequence," *American Society of Engineering Education*, 2016.

[4] K. Umapathy and A. D. Ritzhaupt, "A meta-analysis of pair-programming in computer programming courses: Implications for educational practice," *ACM Trans. on Computing Education*, 2016.

[5] L. Williams, E. Wiebe, K. Yang, M. Ferzli, and C. Miller, "In Support of Pair Programming in the Introductory Computer Science Course," *Computer Science Education*, vol. 12, no. 3, pp. 197–212, Aug. 2010.

[6] K. Wood, D. Parsons, J. Gasson, and P. Haden, "It's never too early: Pair programming in cs1," *Proc. Australasian Computing Education Conference*, pp. 13–21, 2013.

[7] G. Braught, L. M. Eby, and T. Wahls, "The effects of pair-programming on individual programming skill," *Proc. SIGCSE*, pp. 200–204, 2008.

[8] C. McDowell, L. Werner, H. E. Bullock, and J. Fernald, "The impact of pair programming on student performance, perception, and persistence," *Proc. ICSE*, pp. 602–607, 2003.

[9] ——, "Pair programming improves student retention, confidence, and program quality," *Communications of the ACM*, vol. 49, no. 8, pp. 90–95, 2006.

[10] N. Nagappan, L. Williams, E. Ferzli, Miriam Wiebe, K. Yang, C. Miller, and S. Balik, "Improving the cs1 experience with pair programming," *Proc. SIGCSE*, pp. 359–362, 2003.

[11] L. L. Werner, B. Hanks, and C. McDowell, "Pair-programming helps female computer science students," *Journal on Educational Resources in Computing (JERIC)*, vol. 4, no. 1, 2004.

[12] E. Mendes, L. B. Al-Fakhri, and A. Luxton-Reilly, "Investigating pair-programming in a 2nd-year software development and design computer science course," *Proc. SIGCSE Conference on Innovation and Technology in Computer Science Education*, pp. 296–300, 2005.

[13] E. Mendes, L. Al-Fakhri, and A. Luxton-Reilly, "A replicated experiment of pair-programming in a 2nd-year software development and design computer science course," *Proc. SIGCSE Conference on Innovation and Technology in Computer Science Education*, pp. 108–112, 2006.

[14] L. Williams, R. R. Kessler, W. Cunningham, and R. Jeffries, "Strengthening the case for pair programming," *IEEE Software*, vol. 17, no. 4, pp. 19–25, 2000.

[15] A. Cockburn and L. Williams, *Extreme programming examined*. Addison-Wesley, 2001.

[16] ——, "The costs and benefits of pair programming," *Extreme programming examined*, 2000.

[17] J. E. Hannay, T. Dybå, E. Arisholm, and D. I. Sjøberg, "The effectiveness of pair programming: A meta-analysis," *Information and Software Technology*, 2009.

[18] J. Juett, "Using program visualization to illuminate the notional machine," Ph.D. dissertation, University of Michigan, 2016.

[19] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

Max Smith (S'17) received the B.S.E. degree in computer science from the University of Michigan, Ann Arbor, in 2016. He is currently a Ph.D. student at the University of Michigan, Ann Arbor. His research interests are in machine learning, computer vision, and STEM education.

Andrew Giugliano received the B.S.E. degree in computer science from the University of Michigan, Ann Arbor, in 2016. He is currently a software engineer working in private industry. His research interests are in Computer Science pedagogy. He is an experienced student instructor.

Andrew DeOrio (S'07–M'12–SM'17) received the B.S.E. and M.S.E. degrees in electrical engineering, and the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor, in 2006, 2008, and 2012, respectively. He is currently teaching faculty at the University of Michigan. His research interests are in engineering education, and ensuring the correctness of correctness of computer systems.