Learning gains of Proof Blocks versus Writing Proofs

Anonymous Author(s)

Abstract

TODO: also mention creating a validated rubric?

Proof Blocks is a software tool that provides students with a scaffolded proof-writing experience, allowing them to drag-and-drop prewritten proof lines into the correct order instead of starting from scratch. In this paper we describe a randomized controlled trial with designed to measure the learning gains of using Proof Blocks. The study participants were 332 students recruited after the first month of completing their discrete mathematics course. Students in the study completed a pretest on proof writing and a brief (less than 1 hour) learning activity and then returned one week later to complete the posttest. Depending on the experimental condition that each student was assigned to, they either completed only Proof Blocks problems, completed some Proof Blocks problems and some written proofs, or completed only written proofs for their learning activity. We find that students in the early phases of learning about proof by induction are able to learn just as much by using Proof Blocks as by writing proofs from scratch, but in far less time on task. This finding that Proof Blocks are an effective learning tool complements previous findings that Proof Blocks are useful exam questions and are viewed positively by students.

CCS Concepts

• Mathematics of computing → Discrete mathematics; **• Social and professional topics → Computing education**; **• Applied computing → Computer-assisted instruction**.

Keywords

discrete mathematics, CS education, automatic grading, proofs

ACM Reference Format:

Anonymous Author(s). 2018. Learning gains of Proof Blocks versus Writing Proofs. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction and Background

Despite being on of the most difficult and important aspects of computing theory to learn, there is very little research on the teaching and learning of writing mathematical proofs [15]. Even when students have all the content knowledge needed to construct a proof, they still struggle to write proofs themselves, showing that students need scaffolding to help them through the learning process [30, 31]. To address this need, we've created Proof Blocks, a software tool that allows students to drag-and-drop pre-written proof lines into the correct order rather than having to write a proof completely on their own. In this paper, we detail a study designed to measure the learning gains of students practicing their proof-writing skills using Proof Blocks.

An example Proof Blocks problem can be seen in Figure 1. To create a Proof Blocks problem, an instructor writes the lines of the proof, and specifies which lines of the proof logically depend on which other lines of the proof (see [5] for more details of question authoring). The lines are then randomly rearranged, and students are given the opportunity to drag-and-drop the lines back into the correct order. Each time a student submits and answer, they are able to receive immediate feedback on where their proof went wrong. Our research so far has shown that Proof Blocks problems are a useful tool for assessment: they can provide about as much information about student knowledge as written proofs, while being slightly easier. Prior research by Ericson et al. suggests that in the early stages of learning to write a certain kind of program, Parsons Problems can help students learn just as much as writing code from scratch, but in a shorter amount of time [13, 14]. We performed a randomized controlled trial to compare the learning gains of students who completed a Proof Blocks learning activity to students who completed a proof writing learning activity, and students who completed a learning activity with some proof writing and some Proof Blocks problems.

TODO: Where do we want this paragraph on why to choose proof by induction?

Proof by induction is also a perfect topic because it was rated as the most important topic in the course while also being the third most difficult of the 37 topics in the course [15]. Proof by induction is so important because it is a prevalent technique that is referred to in upper level courses and it employs recursive and reductive thinking as well as precise logical arguments. Therefore, helping students gain mastery of this topic is impactful not only for their success in the course but also for their later CS courses. Students struggle with this topic and so figuring out what kind of help they need is helpful. It is also an important topic so it is important to make sure students understand it. It is also a topic that is taught with a very clear outline for grading and has isomorphic problems that are easy to practice on and then generalize to the one at test. Prior work shows that students lack key conceptual knowledge related to induction [7, 16, 24]. Students often think they understand while reading the proofs but then cannot reproduce a proof for a new problem [31].

1.1 Research on Teaching and Learning Proofs

There are many threads of research in seeking to illuminate students' understandings and misunderstandings about proofs [26–28]. One thread establishes that, as they learn, students go through different phases in the complexity of ways they are able to think about solving proof problems [32]. Another study demonstrated that even when students had all of the knowledge required to write a proof

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

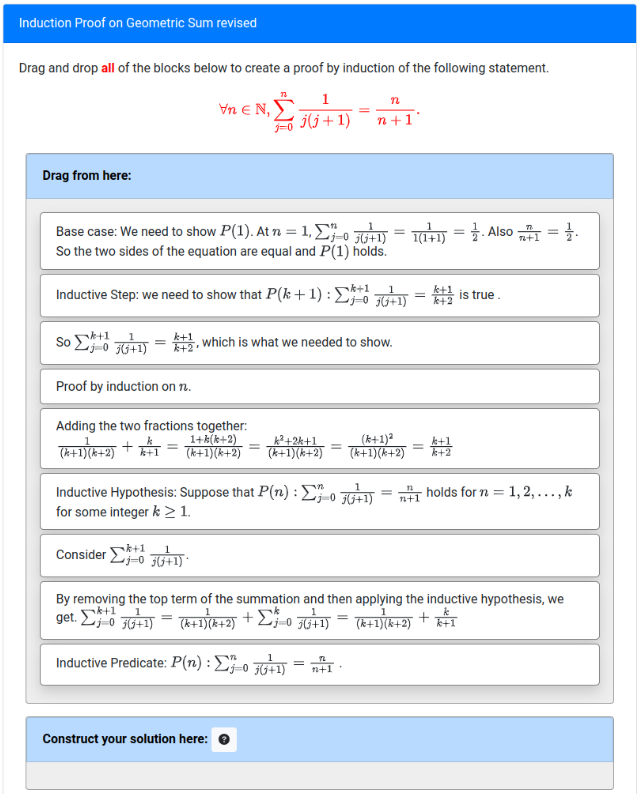

Figure 1: Example Proof Blocks problem from the Proof Blocks learning activity. Students are provided with the lines of the proof in scrambled order, and they must drag and drop them back into the correct order. They are able to receive instant feedback from the computer about the correctness of their solution.

and were able to apply that knowledge in other types of questions, they were still unable to write a proof [31], thus highlighting the need to scaffold students through the proof writing process.

On the other hand, there is little research on concrete educational interventions for improving the proof learning process [17, 28]. Stylianides and Stylianides [28] conducted an extensive review of the literature on teaching and learning proofs. They gave an overview of the emerging literature, and then concluded that: "more intervention-oriented studies in the area of proof are sorely needed" [28].

There have been a few experimental studies seeking to improve student understanding of proofs [21, 25, 33]. Hodds et al. [17] showed that training students to engage more with proofs by using self-explanation methods increased student comprehension of proofs in a lasting way. Proof Blocks problems similarly force deliberate engagement with proof content, as close reading is necessary to determine the correct arrangement of lines. There have also been interventions that focus on helping students understand the need for proofs, as students often believe empirical arguments without seeing the need for proof [6, 18, 28, 29]. There have been a few other studies which taken a non-experimental approach to proof education interventions, give a rich reflection on the development of novel instructional practices [16, 20]. To our knowledge, our study is the first experimental study on an intervention to improve student's abilities to write mathematical proofs.

1.2 Parsons Problems

The use of scrambled code problems was first documented by Parsons & Haden. [23]. They have since been studied for their desirable properties both in assessment and learning [10, 11, 14]. The desirable properties of Parsons problems were a major inspiration for the creation of Proof Blocks. Denny et al. [10] showed that Parsons problems are easier to grade than free-form code writing questions, and yet still offer rich information about student knowledge, and we have shown the same to be true of Proof Blocks problems relation to free-form proof-writing questions [3].

Ericson et al. [13, 14] performed a series of randomized controlled trials to compare the learning gains of students using Parsons problems to the learning gains of students. First, they compared the learning gains of Parsons problems to fixing code and writing code from scratch in a posttest measurement of both Parsons problems and code writing questions, they found that students learned equal amounts across all conditions. In their next experiment, they compared adaptive Parsons problems, Parsons problems, code writing, and a control condition where students solved Parsons problems on a topic unrelated to the posttest topic. They found that students who completed the Parsons problems and adaptive Parson's problems learned the most. Work is in progress to replicate this work across many universities [12]. Similar to the work of Ericson et al, the study which we report on is designed to confirm if Proof Blocks also help accelerate the learning process writing proofs.

Many variants of Parsons problems have appeared in classrooms and in the research literature. The Proof Blocks problems used in our study most closely resemble Parsons problems with relative line-based feedback and no distractors [11]. While Proof Blocks does support the use of distractors, and many instructors use them this way in the classroom, we decided to use no distractors in the study for simplicity.

1.3 Research Questions

Since one traditional learning activity for students learning to write mathematical proofs is to attempt to write proofs on their own from scratch and then viewing an example solution to assess their progress, which we call a *proof-writing* activity, we feel that this makes the most appropriate baseline for comparing the learning gains of Proof Blocks. Because Proof Blocks are a scaffolded version of proof writing, we felt that completing some Proof Blocks problems and then some proof-writing problems would be an effective learning activity. We call this a *hybrid* learning activity. In this study, we seek to quantify the difference in learning gains between students who complete a learning activity containing only Proof Blocks problems, a learning activity containing only proof-writing problems, and students who complete a hybrid learning activity.

We go in to this study with the following research questions:

> **RQ1**: What are the differences in learning gains between students who complete a Proof Blocks activity, proof-writing activity, or hybrid activity?

Group A	Group B	Group C
Pretest	Pretest	Pretest
Lecture Notes	Lecture Notes	Lecture Notes
Proof Blocks	Hybrid Activity	Proof Writing
Practice Test	Practice Test	Practice Test
	One Week	
Posttest	Posttest	Posttest

Table 1: Design of the learning experiment. Students were free to move on as soon as they finish a particular portion of the activity

RQ2: How does time on task vary between the experimental conditions?

Though it is possible that student in the Proof Blocks and hybrid learning condition will learn less or more than the students who practice by writing proofs from scratch, we hypothesized that the learning gains for students in the hybrid activity will be the most time efficient, meaning that students in the hybrid activity will learn the most, relative to the amount of time spent on task. Maybe connect this to the CHI paper on faded Parsons problems? i.e., more scaffolding is better...

2 Experimental Design

We used a between-subjects experimental design. To control for confounding variables, we ran our study as a controlled lab study rather than as part of a course. We recruited students from the Discrete Mathematics course in our department who already had knowledge of some kinds of proofs, and used Proof Blocks to teach them a new kind of proof. All students who participated took a pretest, read through some lecture notes about proof by induction, completed a learning activity (proof-writing, Proof Blocks, or hybrid), and then took a practice test (see Table 1). We also invited them all to participate in a posttest one week later. The pretest, practice test, and posttests are exactly identical, all containing the exact same two proof-writing problems. While the students were shown the example solutions for all exercises they complete during the learning activities, they were not shown the example solutions for any of the test questions.

2.1 Experiment Environment

All students completed their learning activities and tests in an Open source Homework and Exam Platform, which we will call OHEP for anonymization purposes [1]. Since OHEP automatically keeps track of when a user opens and closes each assessment, and when they submit an answer to each problem, we are able to track and analyze the amount of time that students spend on each portion of the study. Since Fall 2021, all sections of Discrete Mathematics have submitted their homework and completed their exams through OHEP, so the students we recruited for the study are already familiar with the platform and its user interface. Students wrote their proofs in text entry box which supported markdown and LaTeX, but were told that using plain text (for example, spelling out "sum from i=0 to n" instead of using $\sum_{i=0}^{n}$) was acceptable as they were not expected to learn LaTeX for the course. To control the student learning environment for our study, we used the University's Computer Based Testing Facility (CBTF) [2], which provides a proctored and locked down environment where student work on lab computers where they can only access the assessment which they are supposed to be working on. The CBTF allows flexible scheduling so that students can choose to complete the learning activity at any point over the period of a few days.

2.2 Experimental Subjects

Proof by induction was a perfect topic for this purpose, because students in the Discrete Mathematics course in our department learn proof by induction sometime in the middle of the semester. Thus, we could recruit students for our study roughly a month into the semester, after they have learned the basics of writing proofs, but before their course has covered anything about proof by induction. The Discrete Mathematics course course in our department is typically taken by freshmen students in the computer science and computer engineering majors or computer science minor who have already completed introductory programming and calculus classes.

In the study we had students complete proofs by induction on two different topics: proving the correctness of closed-form formulas for finite sums, which we expected to be easier for students to complete, and proving the correctness of a closed-form solution to a recursively defined function, which we expected to be more difficult.

We ran a pilot study during Fall 2021 with 5 students to test our materials, and students were given a gift card as compensation for their participation. For the main study during the Winter of 2022, we offered students one homework assignment of extra credit for participating in each day of the study (one for the learning activity, another one for the posttest a week later). Because students completed their learning activities at various times, we assigned all eligible students to a treatment condition upon announcing the study. We had 451 students participate in the learning activity. Of these, 353 students showed up the following week to complete the posttest. As allowed by the terms of our research protocol, 13 subject in this pool opted to not have their data used for purposes of the research project. Another 8 participants did not complete all the questions. After removing these, we were left with a final data set of 332 students. Broken down by experimental condition, 107 of 138 (77.5%) of students who started in the Proof Blocks condition, 112 of 160 (70%) of students who started in the Hybrid learning condition, and 113 of 153 (73.8%) of students who started in the Written Proofs condition were included in the final data set.

2.3 Experimental Materials

We designed the experiment so that as much as possible would be the same between groups except for the way in which they are constructing their proofs during the learning activity. The theorems which the students are proving are the exact same in between the proof writing and Proof Blocks problems, and the example solutions which are shown to the students writing proofs from scratch to help them self-grade are identical to the proofs in the Proof Blocks problems. This way all students are provided with all of the exact same information, the only difference is the way in which they are interacting with that information.

Table 1 gives a breakdown of the parts of the study completed by the subjects in each experimental condition If student finish one part of the activity early, they are allowed to immediately move on to the next portion.

As an extra check for the comparability of experimental groups, at the beginning of the learning activity students were asked a single question to gauge their level of familiarity with proof by induction: "What was your level of familiarity with proof by induction before today?" with answer choices (a) I was very familiar with proof by induction, (b) I was somewhat familiar with proof by induction, and (c) I had never heard of proof by induction.

Each of the learning activities consists of five problems: three which are isomorphic proofs to the first proof problem from the test, and 2 which are isomorphic to the more difficult problem. The students completing the Proof Blocks learning activity completed all 5 of these problems as Proof Blocks, while the students in written proofs activity completed all 5 of these problems as written proofs. In the hybrid activity, students are given two Proof Blocks problems, then one written proof, followed by another Proof Blocks problem and then another written proof.

To encourage students working on written proofs to make a good faith attempt at writing the proof instead of just clicking through the prompts in addition to the text entry box for them to write their proof in, we added another text entry box to the problem, with a prompt for students to compare their proof to the example proof. The instruction here to "compare" is intentionally vague. We wanted to say enough to encourage the students to make a good faith attempt at each problem, but we did not want to give students extra scaffolding for meta-cognition above and beyond what they were given in the Proof Blocks problems, to ensure that we are fairly comparing between the learning conditions.

Students working on Proof Blocks were given instant feedback on their work. They were told if their submission was correct or not. If they made a mistake, they were told which line of their proof was the first incorrect line, based on the logical structure of the proof. This type of feedback is commonly used in Parsons problems, and in the Parsons Problems literature it has been called relative line-based feedback [11]. For more details of the Proof Blocks autograder and feedback system, see [4, 5]. Students were given 3 tries to complete each Proof Blocks before being shown the correct answer.

3 Data Analysis

3.1 Rubric

There are also no existing validated rubrics for assigning student grades for mathematical proofs. Thus, as part of our study, we needed to design and validate a rubric for grading student proofs, in addition to grading all student work by hand. While there has been some work to understand how mathematician's typically grade proofs [22], we have no knowledge of research attempting to find a standard rubric.

This use of a rubric can be considered a data transformation which converts our qualitative data (student's written proofs) to quantitative data (a numerical score). A discussion of the methods and validity of such transformations, and more examples, can be seen in standard books on mixed methods research: [8, 9]. We started out by examining rubrics that had been used for proof by induction problems in recent semesters in our department. One of the issues we found was that many portions of the rubric relied on human judgment, such as assigning a wide range of points for "remaining algebra details". While this may be acceptable for use in a classroom, we wanted to remove as much human judgment out of our rubric as possible so that the points could be reliably applied by anyone, thus giving us a more robust answer to our research questions.

Proof Section	Proof Detail
Base Case(s)	(1) Identify Base Case(s) (2) Prove Base Case(s)
Inductive Hypothesis	(3) Hypothesis is stated (4) Hypothesis is given some bound
Inductive Step	(5) Goal is Stated (6) Expression of Size $k+1$ is decomposed into expression of size k (7) Inductive Hypothesis is applied

Table 2: Rubric for qualitative analysis of written proofs. For each proof, each point on the rubric is assigned the following points: 0 for not present, 1 for partially correct, or 2 for correct. We validated our rubric by having multiple authors grade the same proofs and iteratively refining it. In the final round we achieved a Krippendorff's alpha of 0.82.

After making some preliminary revisions, we started to grade the student proofs collected during our pilot study in order to establish our rubric. To start, three members of the research team independently graded two proofs written by one student. We met together with a fourth member of the research team to come to an agreement about which rubric points should have been applied to each proof, revising our rubric and adding clarifications and changes as necessary. After this, three members of the research team coded proofs by more experimental subjects, and we met together to attempt to agree on the meaning of the rubric points and clarify their wording as much as possible. After the final round of grading the pilot study data according to our rubric, we used Krippendorff's alpha to calculate an inter-rater reliability of 0.82 over $n = 56$ rubric points (8 proofs). Since this was above the generally accepted threshold of 0.8 [19], we decided that our rubric was reliable enough to have only one member of the research team grade each student proof moving forward. The final rubric can be seen in Table 2. When grading the data from the main study, the graders were blinded to whether the proof they were grading came from a pretest or posttest, and were blinded as to what experimental condition the subject they were grading was under. Of the $332 \times 4 = 1,328$ proofs in the final data set, $1,170$ were graded by author the first author, 49 were graded by the second author, and 109 were graded by the third author.

The proof by induction familiarity survey question was also converted to a numerical score, with 'very familiar' being assigned 2, 'somewhat familiar' being assigned 1, and 'never heard of' being assigned 0.

Level of Familiarity with Induction	Experimental Condition		
	Proof Blocks	Hybrid	Written Proofs
Very Familiar	9	6	7
Somewhat Familiar	37	30	29
Never Heard	66	71	77

Table 3: Breakdown of prior students knowledge by experimental group. This data, along with the pretest scores, confirms that each of the experimental groups started out roughly equal in knowledge of proof by induction.

3.2 Comparability of Experimental Groups

Because students were randomly assigned to experimental conditions by a computer, we had strong reason to believe a priori that the populations of student in each experimental group were comparable, but we still ran statistical checks to be certain. First, we compared the student responses to the survey on the familiarity with proof by induction. A Shapiro-Wilk test showed the data to be non-normal ($p < .001$ for all three experimental groups), so we used a Kruskal-Wallis test to confirm that the familiarity level is similar across groups. We fail to reject the null hypothesis that the distribution of familiarity scores between groups are the same ($\chi^2 = 2.34$, $p = 0.31$). Details of the familiarity survey can be seen in Table 3. We also compare the groups pretest scores to one another in a similar manner. A Shapiro-Wilk test also showed that the pretest scores were non-normal ($p < .001$ for all three experimental groups), so we again use a Kruskal-Wallis test for this comparison. We fail to reject the null hypothesis that the distribution of pretest scores between groups are the same ($\chi^2 = 0.97$, $p = 0.62$).

After verifying the validity of the experiment by confirming that experimental groups had similar baseline knowledge, we proceed with the portion of the analysis that directly answers the research questions.

3.3 Learning Gains

To measure learning gains and answer **RQ1**, we compare the pretest scores of each group to the posttest scores of the same group. These score distributions can be seen in Figure 2. A Shapiro-Wilk test also showed that the posttest scores were non-normal ($p < .001$ for all three experimental groups), so we use a paired Mann Whitney U test to test for learning gains within each group, and found that all three groups performed significantly better on the posttest than on the pretest ($p < .001$ for all three groups). All three groups improved by between 9 and 10 points on average, or between 32% and 35% (2 questions $\times$ 7 rubric items $\times$ 2 points each = 28 points possible in total). See Table 4 for full details, and Figure 2 to see plots of the score distributions. Because of concerns over the use of Cohen's d for effect sizes between non-normal distributions, we use Wilcox's Q as our standardized effect size for learning gains, and find that students improved by a standardized effect size between 0.67 and 0.74. We also performed a Kruskal-Wallis on the posttest scores between groups, and we do not reject the null hypothesis that students from each experimental group performed equally well on the posttest ($\chi^2 = 0.54$, $p = 0.76$).

Thus, we conclude that students in all three experimental groups learned more about proof by induction by reading the lecture notes and completing their respective learning activities, and that no one group learned significantly more than the others.

3.4 Time Spent on Learning Activity

Next, to answer **RQ2**, we examine the amount of time that students from each group spent on the learning activity (see Figure 3). Pairwise t-tests show that students in the Written Proof activity took significantly longer on their learning activity that students in the Hybrid and Proof Blocks conditions, and that students in the Hybrid condition took significantly longer than students in the Proof Blocks condition ($p < 0.001$ in all cases). Students in the Proof Blocks condition completed their activity about 4 times faster on average than students in the Written Proofs condition. Full details of the time differences and statistical tests can be seen in Table 5. Using this together with the above result that students in all conditions learned equally, we conclude that students in the Proof Blocks condition learned more per unit time than students in the Written Proofs condition.

Many students took much longer than expected on the 'Written Proofs' learning activity, resulting in some students possibly not having time to complete the practice test. As a robustness check, we re-ran the analysis, dropping all students who may not have had time to finish the practice test, which was all those who took longer than 55 minutes, or 0.56 standard deviations above the mean, which turned out to be 37 out of 113 students in the Written proofs condition. To obtain a fair comparison, we likewise dropped all students who were 0.56 standard deviations above mean time on the hybrid activity, which was 24 of the 112 students who did the hybrid learning activity. We found all results of the analysis to be the same. Since our robustness check shows the results for the subset of the data would be identical to that of the entire data set, for simplicity we present the results over the whole data set.

3.5 Conditional Analyses

As a further robustness check, we compared only students who responded similarly on the familiarity survey. We re-ran all analyses, considering only students who had never heard of proof by induction ($n = 96$), and considering only students who were somewhat familiar with proof by induction ($n = 214$), and found the same results in both cases. We did not re-run the analysis for students who were very familiar with proof by induction, because there are too few of them to have meaningful results ($n = 22$).

We were also interested if students from the different experimental groups learned different parts of the proof by induction at different amounts. To test for this, we re-ran all analysis on each section of the rubric individually: the base case, the inductive hypothesis, and inductive step. All analyses showed exactly the same results, indicating that the learning happened roughly equally across all parts of the rubric, across all three experimental groups. Rerunning all analyses on each of the two test questions separately also gave all the same results, showing that the learning was not differential across the two test questions.

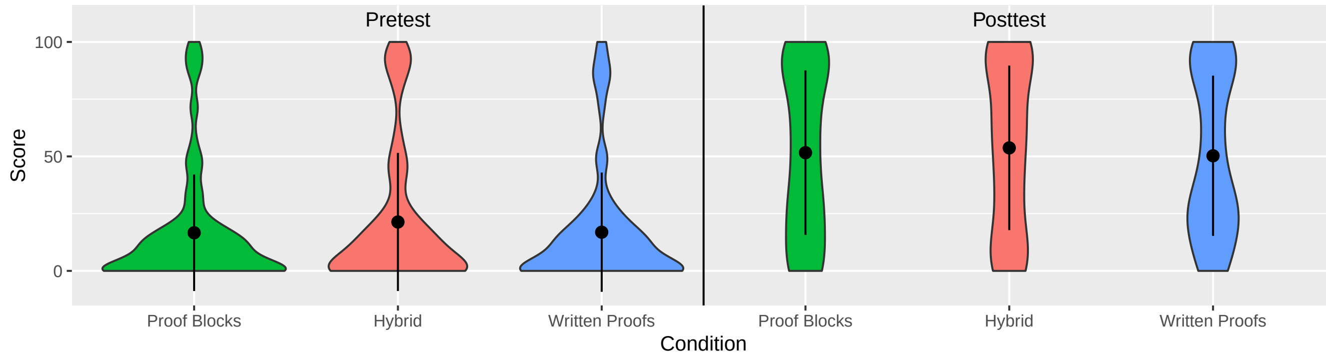

Figure 2: Comparison of pretest/posttest performance across conditions. Students in all 3 experimental groups increased their scores by between 30 and 40 percent between the pretest and the posttest. See Table 4 for full score details

	Pretest Score	Posttest Score	Score Improvement	Wilcox's Q	Activity Time (minutes)
Proof Blocks (n = 107)	16.66% (25.45%)	51.64% (35.93%)	34.98%	0.72	11.09 (6.35)
Hybrid (n = 112)	21.36% (30.16%)	53.73% (35.95%)	32.37%	0.76	32.13 (16.01)
Written (n = 113)	16.91% (26.05%)	50.28% (34.98%)	33.38%	0.66	43.36 (20.61)

Table 4: Mean and and standard deviation for the pretest and posttest scores, score improvement between the pre and post-test, standardized score improvement (Wilcox's Q), and time spent on the learning activity for each experimental group.

	Mean Time Difference (minutes)	Cohen's d
Hybrid – Proof Blocks	21.04	1.71
Written – Proof Blocks	32.27	2.09
Written – Hybrid	11.23	0.61

Table 5: Difference in time spent on the learning activity between each pair of experimental groups. The students who completed the written proofs learning activity took the longest by far, but did not learn any more than the other groups.

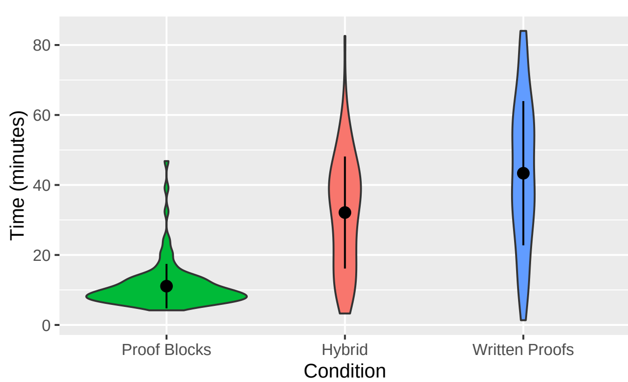

Figure 3: Distribution of time spent on each learning activity by each of the experimental groups. Students who completed the Proof Blocks activity were able to complete their activity much more quickly.

4 Discussion and Limitations

The statistical evidence is clear that students who completed the Proof Blocks and hybrid activities learned as much as students who completed the Written Proofs activity, but in a shorter amount of time.

One limitation of our study is that it has limited ecological validity. For example, it would be useful to study the ideal way to have Proof Blocks problems situated within course content. Another limitation is that because the learning of all three experimental groups was the same, there is possibility that the student learning happened entirely from reading the lecture notes. We will address this concern in a follow-up study in which one of the experimental groups completes either no learning activity, or a learning activity in an unrelated topic. Due to the conditions of our IRB protocol, we do not have any demographic information about our study participants.

5 Conclusions

In this work, we created a validated rubric for measuring student performance on proof by induction problems. We used this rubric to measure student learning gains across different learning activities in a randomized controlled trial. Our experiment showed that students in the early phases of learning a new type of proof can learn just as much using Proof Blocks as writing proofs on their own, but in far less time. Future work should continue to investigate the merits of Proof Blocks as a learning tool.

References

[1] Anonymous. 2015. Paper Describing an Online Learning System.
[2] Anonymous. 2019. Paper About Computer Based Testing Facility.
[3] Anonymous. 2021. Paper About Proof Blocks Problems as Exam Questions.
[4] Anonymous. 2022. Paper about Partial Credit Grading of Proof Blocks Problems.
[5] Anonymous. 2022. Tool paper about Proof Blocks.

[6] Stacy A. Brown. 2014. On skepticism and its role in the development of proof in the classroom. *Educational Studies in Mathematics* 86, 3 (July 2014), 311–335. https://doi.org/10.1007/s10649-014-9544-4

[7] Stacy A Brown et al. 2008. Exploring epistemological obstacles to the development of mathematics induction. In *The 11th Conference for Research on Undergraduate Mathematics Education*. 1–19.

[8] Elizabeth G Creamer. 2017. *An introduction to fully integrated mixed methods research*. sage publications.

[9] John W. Creswell. 2007. *Designing and conducting mixed methods research*. SAGE Publications, Thousand Oaks, Calif.

[10] Paul Denny, Andrew Luxton-Reilly, and Beth Simon. 2008. Evaluating a new exam question: Parsons problems. In *Proceedings of the fourth international workshop on computing education research*. 113–124.

[11] Yuemeng Du, Andrew Luxton-Reilly, and Paul Denny. 2020. A Review of Research on Parsons Problems. In *Proceedings of the Twenty-Second Australasian Computing Education Conference (ACE'20)*. Association for Computing Machinery, New York, NY, USA, 195–202. https://doi.org/10.1145/3373165.3373187

[12] Barbara J Ericson, Paul Denny, James Prather, Rodrigo Duran, Arto Hellas, Juho Leinonen, Craig S Miller, Briana Morrison, Janice L Pearce, and Susan H Rodger. 2022. Planning a Multi-institutional and Multi-national Study of the Effectiveness of Parsons Problems. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 2*. 576–577.

[13] Barbara J Ericson, James D Foley, and Jochen Rick. 2018. Evaluating the efficiency and effectiveness of adaptive parsons problems. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*. 60–68.

[14] Barbara J Ericson, Lauren E Margulieux, and Jochen Rick. 2017. Solving parsons problems versus fixing and writing code. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research*. 20–29.

[15] Ken Goldman, Paul Gross, Cinda Heeren, Geoffrey Herman, Lisa Kaczmarczyk, Michael C Loui, and Craig Zilles. 2008. Identifying important and difficult concepts in introductory computing courses using a delphi process. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*. 256–260.

[16] Guershon Harel. 2001. The development of mathematical induction as a proof scheme: A model for DNR-based instruction. In *In*. Kluwer Academic, 185–212.

[17] Mark Hodds, Lara Alcock, and Matthew Inglis. 2014. Self-explanation training improves proof comprehension. *Journal for Research in Mathematics Education* 45, 1 (2014), 62–101.

[18] Hans Niels Jahnke and Ralf Wambach. 2013. Understanding what a proof is: a classroom-based approach. *ZDM* 45, 3 (May 2013), 469–482. https://doi.org/10.1007/s11858-013-0502-x

[19] Klaus Krippendorff. 2004. Reliability in content analysis: Some common misconceptions and recommendations. *Human communication research* 30, 3 (2004), 411–433.

[20] Sean Larsen and Michelle Zandieh. 2008. Proofs and refutations in the undergraduate mathematics classroom. *Educational Studies in Mathematics* 67, 3 (March 2008), 205–216. https://doi.org/10.1007/s10649-007-9106-0

[21] Aliza Malek and Nitsa Movshovitz-Hadar. 2011. The effect of using transparent pseudo-proofs in linear algebra. *Research in Mathematics Education* 13, 1 (2011), 33–58.

[22] David Miller, Nicole Infante, and Keith Weber. 2018. How mathematicians assign points to student proofs. *The Journal of Mathematical Behavior* 49 (2018), 24–34.

[23] Dale Parsons and Patricia Haden. 2006. Parson's Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses. In *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52* (Hobart, Australia) *(ACE '06)*. Australian Computer Society, Inc., AUS, 157–163.

[24] Gila Ron and Tommy Dreyfus. 2004. *The Use of Models in Teaching Proof by Mathematical Induction*. International Group for the Psychology of Mathematics Education, 35 Aandwind Street, Kirstenhof, Cape Town, 7945, South Africa. https://eric.ed.gov/?id=ED489630 Publication Title: International Group for the Psychology of Mathematics Education.

[25] Somali Roy. 2014. *Evaluating novel pedagogy in higher education: a case study of e-proofs*. thesis. Loughborough University. /articles/thesis/Evaluating_novel_pedagogy_in_higher_education_a_case_study_of_e-proofs/9374297/1

[26] Annie Selden and John Selden. 2008. Overcoming Students' Difficulties in Learning to Understand and Construct Proofs. In *Making the Connection*, Marilyn P. Carlson and Chris Rasmussen (Eds.). The Mathematical Association of America, Washington DC, 95–110. https://doi.org/10.5948/UPO9780883859759.009

[27] Andreas J Stylianides, Kristen N Bieda, and Francesca Morselli. 2016. Proof and argumentation in mathematics education research. In *The second handbook of research on the psychology of mathematics education*. Brill Sense, 315–351.

[28] GJ Stylianides, AJ Stylianides, and K Weber. 2017. Research on the teaching and learning of proof: Taking stock and moving forward. In *Compendium for Research in Mathematics Education*, Jinfa Cai (Ed.). National Council of Teachers of Mathematics, Chapter 10, 237–266.

[29] Gabriel J. Stylianides and Andreas J. Stylianides. 2009. Facilitating the Transition from Empirical Arguments to Proof. *Journal for Research in Mathematics Education* 40, 3 (2009), 314–352. https://www.jstor.org/stable/40539339 Publisher: National Council of Teachers of Mathematics.

[30] Lev Semenovich Vygotsky. 1978. *Mind in society: The development of higher psychological processes*. Harvard university press.

[31] Keith Weber. 2001. Student difficulty in constructing proofs: The need for strategic knowledge. *Educational Studies in Mathematics* 48, 1 (Oct. 2001), 101–119. https://doi.org/10.1023/A:1015535614355

[32] Keith Weber and Lara Alcock. 2004. Semantic and Syntactic Proof Productions. *Educational Studies in Mathematics* 56, 2 (July 2004), 209–234. https://doi.org/10.1023/B:EDUC.0000040410.57253.a1

[33] Keith Weber, E Fuller, JP Mejia-Ramos, Kristen Lew, Philip Benjamin, and Aron Samkoff. 2012. Do generic proofs improve proof comprehension. In *Proceedings of the 15th Annual Conference on Research In Undergraduate Mathematics Education*. Citeseer, 480–495.