ABSTRACT
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 designed to measure the learning gains of using Proof Blocks for students learning proof by induction. The study participants were 332 students recruited after completing the first month of their discrete mathematics course. Students in the study took a pretest and read lecture notes on proof by induction, completed 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 from reading lecture notes and using Proof Blocks as by reading lecture notes and writing proofs from scratch, but in far less time on task. This finding complements previous findings that Proof Blocks are useful exam questions and are viewed positively by students.

CCS CONCEPTS
• Mathematics of computing → Discrete mathematics; • Applied computing → Computer-assisted instruction.

KEYWORDS
discrete mathematics, CS education, automatic grading, proofs

1 INTRODUCTION AND BACKGROUND
There is little research on teaching interventions for writing mathematical proofs, despite it being one of the most difficult and important aspects of computing theory to learn [10, 26]. Even when students have all the content knowledge needed to construct a proof, they still struggle to write proofs, showing that students need scaffolding to help them through the process [28, 29]. To address this need, we’ve created Proof Blocks, a software tool that allows students to drag and drop prewritten proof lines into the correct order rather than having to write a proof completely on their own (see Figure 1 for an example Proof Blocks problem). In this paper, we detail a study designed to measure the learning gains of students practicing their proof-writing skills using Proof Blocks. The authors’ prior work reviews other software tools providing visual methods of constructing proofs [21]. While Proof Blocks have been shown to be useful in assessment [20], there has been...
no previous research on their utility for student learning. We hypothesize that Proof Blocks problems can also improve learning, because the prewritten lines provide a scaffold that helps students focus on the sub-task of constructing logical arguments. To test this hypothesis, 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.

1.1 Research on Teaching and Learning Proofs
Many threads of research seek to illuminate students’ understandings and misunderstandings about proofs [24–26]. One thread establishes that, as they learn, students go through different phases in how they are able to think about solving proof problems [30]. Another study demonstrated that even when students had all of the knowledge required to write a proof and were able to apply that knowledge in other types of questions, they were still unable to write a proof [29], thus highlighting that performing all aspects of proof writing at once is a separate challenge than learning the component skills of proof writing.

On the other hand, there is little research on concrete educational interventions for improving the proof-learning process [12, 26]. Based on their review of the literature on teaching and learning proofs, Sylwianides and Stylianides concluded that “more intervention-oriented studies in the area of proof are sorely needed [26].”

There have been a few experimental studies seeking to improve student understanding of proofs [16, 23, 31]. Hodds et al. [12] showed that training students to engage more with proofs by using self-explanation methods increased student comprehension of proofs in a lasting way. 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 [2, 13, 26, 27]. A few studies have taken a non-experimental approach, giving reflections on the development of novel instructional practices [11, 15]. To our knowledge, our study is the first experimental study on an intervention to improve students’ abilities to write (as opposed to understand) mathematical proofs.

1.2 Parsons Problems
Proof Blocks were inspired by Parsons problems, which similarly scaffold students’ learning of programming by scrambling prewritten lines of programs [18]. Parsons problems have been studied for their desirable properties both in assessment and learning [5, 6, 8, 9, 32]. Denny et al. [5] showed that Parsons problems are easier to grade than free-form code writing questions, and yet still offer rich information about student knowledge. The same is true of Proof Blocks problems relative to free-form proof-writing questions [20].

Ericson et al. [8, 9] performed a series of randomized controlled experiments to compare the learning gains of students using Parsons problems to the learning gains of students doing other learning activities. First, they compared the learning gains of students using Parsons problems to students 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 Parsons’ problems learned the most. Work is in progress to replicate this work across many universities [7]. Similar to the work of Ericson et al., the study which we report on is designed to confirm if Proof Blocks help accelerate the learning process of writing proofs.

1.3 Research Questions
One traditional proof learning activity is for students to attempt to write mathematical proofs on their own. Other research has been rated as one of the most important and difficult topics in a discrete math course [10]. Proof by induction is so important because it is a prevalent technique that is referred to in upper level courses and it employs recursive 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. Prior work shows that students lack key conceptual knowledge related to induction [1, 11, 22].

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). As soon as a student finished one section, they were allowed to proceed to the next. We also invited them all to participate in a posttest one week later. The pretest, practice test, and posttests were exactly identical, all containing the exact same two proof-writing problems: proving the correctness of closed-form formula for a finite sum, and proving...
Table 1: Design of the learning experiment. Students were free to move on as soon as they finish a particular portion of the 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 | Posttest | Posttest |

Efficiency of Learning from Proof Blocks Versus Writing Proofs | SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

2.1 Experiment Environment
All students completed their learning activities and tests in PrairieLearn, a problem-driven online learning system [33]. Since PrairieLearn automatically keeps track of when a user opens and closes each assessment and when they submit an answer to each problem, we were 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 PrairieLearn, so the students we recruited for the study are already familiar with the platform and its user interface. Students wrote their proofs in a text entry box that 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 our University’s computer based testing facility, which provides a locked down environment where student could only access the assessment that they were working on [35]. Students could choose to complete the learning activity at any point over the period of a few days.

2.2 Experimental Subjects
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 proof by induction. Since all parts of the experiment were complete before the students covered proof by induction in class, the students had little to no motivation to study the material outside of the context of the study, helping with the validity of the experiment. The Discrete Mathematics course in our department is typically taken by first year students in the computer science and computer engineering majors or computer science minor. Introductory programming and calculus are prerequisites.

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 Spring 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 for the posttest). Due to constraints of PrairieLearn, it was much easier to pre-assign all eligible students to an experimental condition before they elected to participate. This resulted in a small variation in the population of sizes for each treatment. 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 of 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 Learning Activity Materials
We designed the experiment to maximize the similarity between learning activities, so that the only difference between the learning activities was the way students were constructing their proofs. The theorems that the students are proving are the exact same between the proof writing and Proof Blocks problems, and the example solutions that are shown to the students writing proofs from scratch are identical to the proofs in the Proof Blocks problems. Thus, all students are provided with all of the exact same information, the only difference is the way they are interacting with that information.

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 that are similar proofs to the first proof problem from the test, and two that are similar to the second problem. The students completing the Proof Blocks learning activity completed all five of these problems as Proof Blocks, while the students in written proofs activity completed all five 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.

We wanted to encourage students working on written proofs to make a good faith attempt at writing the proof instead of just clicking through the prompt. To do so, 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 shown them after their initial submission. The instruction to “compare” is intentionally vague. We wanted 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, including which line of their proof was the first incorrect line. This type of feedback is commonly used in Parsons problems, and it has been called relative line-based feedback [6].
Table 2: Rubric for grading written proofs. Each detail 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, achieving a Krippendorff’s alpha of 0.82.

| Proof Section | Proof Detail | Level of Familiarity | Experimental Condition |
|---------------|--------------|----------------------|------------------------|
| Base Case(s)  | (1) Identify Base Case(s) | Very Familiar | Proof Blocks 9 Hybrid 6 Written Proofs 7 |
|               | (2) Prove Base Case(s)     | Somewhat Familiar | 37 30 29 |
| Inductive Hypothesis | (3) Hypothesis is stated | Never Heard | 66 71 77 |
|               | (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 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.

For more details of the Proof Blocks autograder and feedback system, see [19, 21]. Students were given three tries to complete each Proof Blocks before being shown the example solution. 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 this first study for simplicity, but we plan to do so in future work.

3 DATA ANALYSIS

3.1 Rubric

There are no existing validated rubrics for assigning student grades for mathematical proofs. Thus, as part of our study, we designed and validated a rubric for grading student proofs, in addition to grading all student work by hand. While there has been some work to understand how mathematicians typically grade proofs [17], we have no knowledge of research attempting to find a standard rubric.

This use of a rubric can be considered a data transformation that converts our qualitative data (students’ 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: [3, 4]. We started out with a rubric that had been used for proof by induction problems in recent semesters in our department, then edited it to remove as much human judgment as possible in the interest of creating a reliable measure. We then graded the student proofs collected during our pilot study. To start, three members of the research team independently graded two proofs from one student. We met together with a fourth member of the research team to come to an agreement about which rubric points should be applied to each proof, revising our rubric as necessary. Then three members of the research team graded proofs by more students, 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, 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 [14], 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 is shown in Table 2.

Each point on the rubric is assigned 0 for not present, 1 for partially correct, or 2 for correct (2 questions × 7 rubric items × 2 points each = 28 points possible on the test), and then the test score as a whole is converted to be out of 100 for ease of reporting. As a further measure to ensure no bias entered the grading process, 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 × 4 = 1,328 proofs in the main study 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.

3.2 Comparability of Experimental Groups

Because students were randomly assigned to experimental conditions, we had strong reason to believe a priori that the populations of students 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 results 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 comparability of the experimental groups had similar baseline knowledge, we proceed with 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, with summary statistics in Table 4. 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 rubric points on average, or between 32% and 35%. Because of concerns
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.

Table 4: Mean, 20% and 80% quantiles for the pretest and posttest scores (out of 100), score improvement between the pretest and posttest, standardized score improvement (Wilcox’s $Q$), and time spent on the learning activity for each experimental group.

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 four times faster on average than students in the Written Proofs condition.

Within a given learning condition, there is a relatively weak relationship between time spent on task and posttest score. For the Proof Blocks learning activity, time spent on the activity and posttest score were essentially uncorrelated ($p = 0.32$, $r = 0.10$, 95% CI $[-0.09, 0.28]$). The time spent on the activity and posttest scores were moderately correlated for the hybrid activity ($p < 0.001$, $r = 0.64$, 95% CI $[0.52, 0.74]$) and for the written proofs activity.
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. Re-running all analyses on each of the two test questions separately also gave all the same results, showing that the learning was not different across the two test questions.

4 DISCUSSION AND LIMITATIONS

Students who read lecture notes completed the Proof Blocks and hybrid activities learned as much as students who read lecture notes and completed the Written Proofs activity, but in a shorter amount of time. This result gives a foundation for future work about student learning using Proof Blocks.

One limitation of our study is that it has limited ecological validity. For example, it would be useful to explicitly study ways to situate Proof Blocks problems within course content, as Weinmann et al. did with faded Parsons problems [32]. 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, and not from the learning activities. 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 on an unrelated topic. We are also not able to comment on learning saturation—how many Proof Blocks or written proof problems must a student complete before they aren’t learning any more from each additional problem? And what is the most effective way to help students continue to improve after they have completed a few Proof Blocks problems? 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 measured 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 learned just as much reading lecture notes and using Proof Blocks as reading lecture notes and 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 in various contexts and how it can be extended to further scaffold and improve learning.

ACKNOWLEDGMENTS

We would like to give a huge thanks to Dave Mussulman and the rest of the staff at the computer based testing facility for helping us use their facility to run our experiment. Seth Poulsen was supported by an NSF Graduate Research Fellowship.
REFERENCES

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

[2] 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

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

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

[5] 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.

[6] 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

[7] 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.

[8] 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.

[9] 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.

[10] 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 9th SIGCSE technical symposium on Computer science education. 256–260.

[11] Guershon Harel. 2001. The development of mathematical induction as a proof scheme: A model for DNR-based instruction. In In. Citeseer.

[12] 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.

[13] Hans Niels Jahnke and Ralf Wambach. 2013. Understanding what a proof is: a human communication research approach. In Proceedings of the 10th Conference for Research on Undergraduate Mathematics Education. 157–168.

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

[15] 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

[16] Alxra 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.

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

[18] Dale Parsons and Patricia Haden. 2006. Parson’s Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses. In Proceedings of the 6th Australasian Conference on Computing Education - Volume 52 (Hobart, Australia) (ACE’06). Australian Computer Society, Inc., AUS, 157–163.

[19] Seth Poulsen, Shubhang Kulkarni, Geoffrey Herman, and Matthew West. 2022. Efficient Partial Credit Grading of Proof Blocks Problems. https://doi.org/10.48550/ARXIV.2204.04196

[20] Seth Poulsen, Mahesh Viswanathan, Geoffrey L. Herman, and Matthew West. 2021. Evaluating Proof Blocks Problems as Exam Questions. In Proceedings of the 17th ACM Conference on International Computing Education Research. 157–168.

[21] Seth Poulsen, Mahesh Viswanathan, Geoffrey L. Herman, and Matthew West. 2022. Proof Blocks: Autogradable Scaffolding Activities for Learning to Write Proofs. In Proceedings of the 2022 ACM Conference on Innovation and Technology in Computer Science Education.

[22] Gila Ron and Tommy Dreyfus. 2004. The Use of Models in Teaching Proof by Mathematical Induction. International Group for the Psychology of Mathematics Education (2004).

[23] Somnai Roy. 2014. Evaluating novel pedagogy in higher education: a case study of e-proofs. thesis. Loughborough University. /articles/thesis/Evaluating_new_pedagogy_in_higher_education_a_case_study_of_e-proofs/9574297/1

[24] 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/UPO9780883885975.009

[25] Andrew J Stylianides, Kristen N Birda, 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.

[26] 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, Jufa Cui (Ed.). National Council of Teachers of Mathematics. Chapter 10, 237–246.

[27] 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.

[28] Lev Semonovich Vygotsky. 1978. Mind in society: The development of higher psychological processes. Harvard university press.

[29] 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

[30] 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

[31] 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, Citeeseer, 480–495.

[32] Nathaniel Weinman, Armando Fox, and Marti A Hearst. 2021. Improving Instruction of Programming Patterns with Faded Parsons Problems. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. 1–4.

[33] Matthew West, Geoffrey L. Herman, and Craig Zilles. 2015. PrairieLearn: Mastery-based Online Problem Solving with Adaptive Scoring and Recommendations Driven by Machine Learning. In 2015 ASEE Annual Conference & Exposition. ASEE Conferences, Seattle, Washington, 26.1238.1–26.1238.14. https://peer.asee.org/24575.

[34] Rand Wilcox. 2019. A robust nonparametric measure of effect size based on an analog of Cohen’s d, plus inferences about the median of the typical difference. Journal of Modern Applied Statistical Methods 17, 2 (2019), 1.

[35] Craig Zilles, Matthew West, Geoffrey L. Herman, and Timothy Breel. 2019. Every University Should Have a Computer-Based Testing Facility. In CSEDU (1). 414–420.