Improving Short and Long-term Learning in an Online Homework System

Ben Prystawski
University of Toronto
ben.prystawski@mail.utoronto.ca

Jacob Nogas
University of Toronto
jacob.nogas@mail.utoronto.ca

Andrew Petersen
University of Toronto
andrew.petersen@utoronto.ca

Joseph Jay Williams
University of Toronto
williams@cs.toronto.edu

ABSTRACT

Online homework systems are common in university courses. While scientific findings about learning could have bearing on how instructors design these systems, there is little guidance available for instructors on the problem of extrapolating scientific results in various contexts to make design decisions in specific settings. This paper leverages the value of online environments to conduct randomized experiments that directly test principles in a real-world introductory programming course. We investigate the relative benefit of giving students explanations of the correct solution to a problem and giving them an additional problem. We find suggestive evidence that students do better on subsequent problems in the same exercise when given an explanation, but they do better on a post-test two weeks later when given an additional practice problem. These results can inform instructors' decisions in designing online homework.

Keywords

education; programming; explanation; online homework; experiment; MOOC

1. INTRODUCTION AND RELATED WORK

Many university courses use online homework systems to give students practice material. These systems enable students to conveniently practice their skills and instructors to automatically grade homework and gather data on student performance. They typically consist of problems with either written or multiple-choice responses for students to complete.

Past research has investigated the effects of these supports on student learning. There is experimental evidence, for example, that explanations and practice problems can help student learning under certain circumstances [5, 6, 8]. However, instructors still must solve the considerable problem of how to translate this research to a real-world setting.

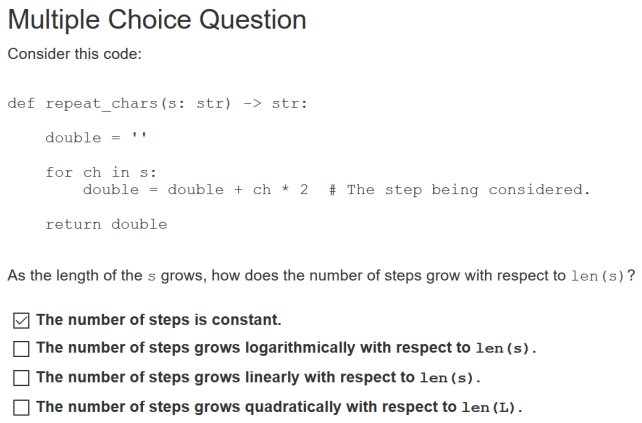

Figure 1: Screenshot of the problem on which the student supports were deployed in the PCRS homework system

While much work in educational data mining has focused on extracting and analyzing data students generate as they naturally interact with these systems, we diverge from that trend in this paper by deliberately embedding a randomized experiment in an online homework environment. We believe that randomized experiments can provide valuable insight for computing education researchers and practitioners because they can directly test the impact of different educational interventions.

The ICAP framework [2, 3] provides a theoretical basis for thinking about different educational methods by grouping them into levels based on the depth of students' engagement with the material. The levels are, from most to least engaging: interactive, constructive, active, and passive methods. Using this framework, one might expect the active learning approach of solving problems to be more effective for students' understanding than the passive approach of reading more explanations. Students must engage with practice problems at a deeper level then explanations, so practice problems might produce better learning outcomes. Comparing the effectiveness of these two methods enables us to test the active-passive boundary within the ICAP framework.

While the ICAP framework might lead one to predict that

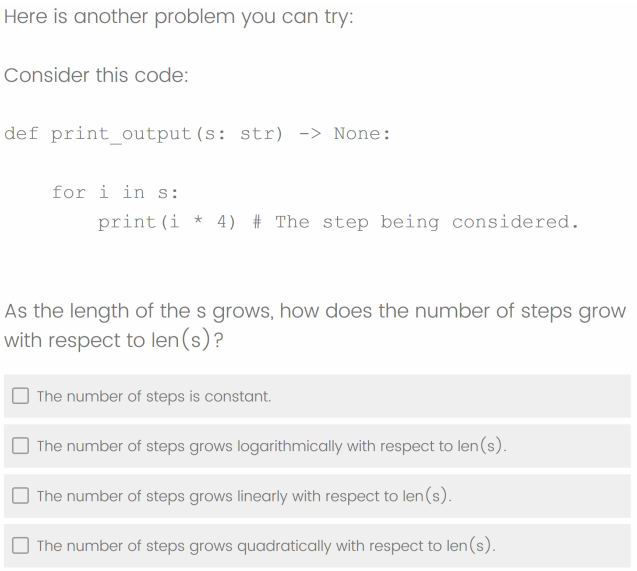

Figure 2: An additional problem given to some students as a follow-up exercise in the online homework system

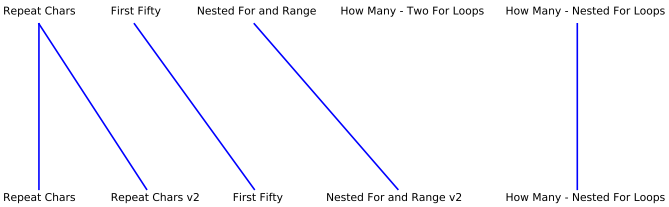

Figure 3: Names of problems in week 10 (top) with corresponding problems in week 12 (bottom). Blue lines indicate correspondence between problems. Problems with the same name are identical, while problems ending in "v2" are very similar but with minor differences such as different variable names. Learning supports were all deployed on the problem "Repeat Chars" in week 10.

an additional problem should be more helpful to learning than an explanation, this could be confounded by the fact that the additional problem is optional. If students spend enough time thinking about and attempting the problem, it should improve their understanding beyond the improvement they would see from reading an explanation of the solution. However, it is also possible that students will dedicate less time and attention to the additional problem than they would to the explanation as trying to solve a problem is a more daunting task than reading an explanation. Furthermore, there is the variable of time to improvement. Perhaps students will not see any immediate benefit from trying an additional problem, but doing it will help their learning in the longer term by cementing their understanding of the concept the problem tests. Will students benefit from additional homework immediately, or will the improvement affect how well the student remembers that week's material later? Both hypotheses appear plausible. Likewise, one might expect that additional explanations will not have a significant effect on student learning. They fit into the passive category of the ICAP framework, which is the lowest level of engagement. The explanation is also optional to read, so students might ignore it entirely. However, one might also expect that reading a well-written explanation of a concept will deepen a student's understanding of the concept they are being tested on. Furthermore, it could be the case that students forget the explanation as soon as they finish working on the exercise, but it could also improve their understanding over a longer period of time, similar to how they remember what they learned from lectures when completing homework.

There is evidence that providing students with instructional explanations when they are solving problems can benefit learning – which is intuitive. However, these explanations are not always effective, especially when they merely give away the answer rather than help students come to see how to solve a problem [8, 11]. For instance, when learners already have some knowledge about a subject, providing additional explanations instead of other knowledge-reinforcing activities can be detrimental to learning [12]. There has also been considerable research on prompting students to write their own explanations in a laboratory setting, finding that having students write their own explanations of key course concepts can help learning [10, 4].

Similarly, the effects of solving problems on learning can be varied and complex under different circumstances. For instance, there is a large body of research on the design of intelligent tutoring systems which automatically determine which practice problems to show learners and in what order to improve their understanding most effectively [1]. However, mathematics and computer science education research point to the challenges in assuming additional practice of problems is always helpful, as sometimes it is a poor use of students' time, or leads them them to focus on procedural knowledge, instead of understanding the underlying principles [7, 9].

These studies have shown that even in a controlled laboratory environment, the effects of these intuitively-helpful interventions vary significantly. Instructors seeking to apply these findings to their courses face the problem of translating findings in laboratory experiments to design decisions about what kind of support to provide in online problems and other educational environments.

Many counterintuitive effects have been found in prior education research, so it is essential to empirically study the effects of interventions before recommending them to instructors. In this paper, we extend upon past literature about the role of reading explanations and solving problems in learning and provide empirical evidence on how these forms of student support affect learning in a real-world setting. Ultimately, we hypothesize that students will perform better on subsequent tests of their understanding when they are shown an additional problem compared to when they are shown an explanation. This hypothesis is motivated by the active-passive boundary from the ICAP framework.

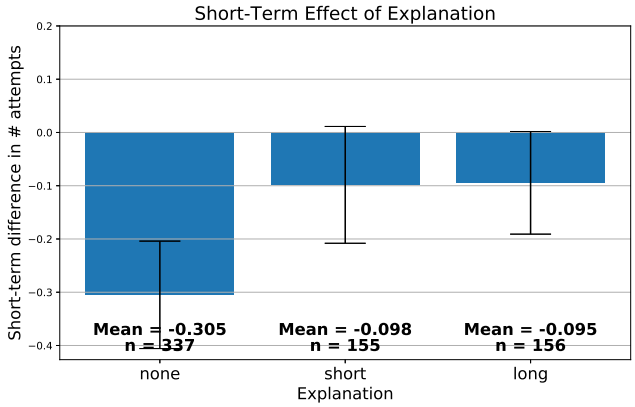

Figure 4: Effect of giving an explanation on number of attempts to get the right answer on subsequent problems in the same exercise. T-test results: none vs. short: (t(490)=-1.24, p=0.215), none vs. long: (t(491)=-1.29, p=0.195), short vs. long: (t(309)=-0.0263, p=0.979)

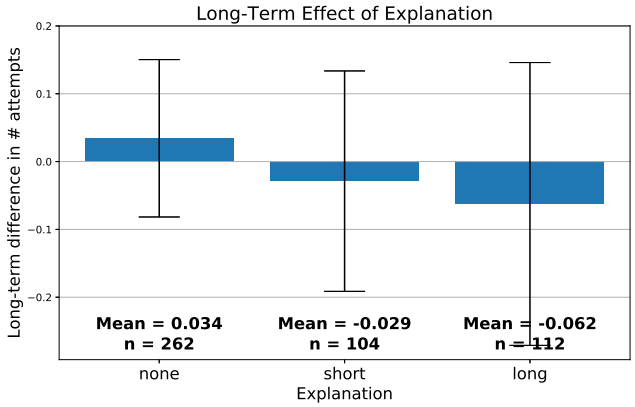

Figure 5: Effect of giving an explanation on number of attempts to get the right answer on the same problem given two weeks later. T-test results: none vs. short: (t(364)=0.300, p=0.764), none vs. long: (t(372)=0.433, p=0.665), short vs. long: (t(214)=0.126, p=0.900)

2. METHODS

The context for the experiment on explanations and additional practice problems was the Programming Course Resource System (PCRS) online homework system for the introductory computer programming course at the University of Toronto. This course spans one twelve-week semester and students are given for-credit online homework exercises each week. The problem we deployed the supports on is shown in Figure 1. It asks students to analyze the runtime of a for loop in Python.

A total of 648 students completed the homework in week 10 of the course. 478 of these students also completed the optional follow-up exercise in week 12. There were 5 problems in each week. This choice of weeks ensures that there is considerable delay between the initial intervention and subsequent measurement, enabling us to measure long-term learning.

In the experiment, after students attempted a homework problem in week 10 of the course, we used a factorial design to independently vary two factors: whether an explanation was provided and whether an additional problem was provided. The experiment was performed in the context of a multiple-choice problem pertaining to run-time analysis, shown in Figure 1. Students were given course credit for completing the main problem, but did not have any direct incentive to read the explanation or attempt the additional problem.

To measure the impact on learning over a longer time frame, we designed a post-test with problems that were either identical to or variants of the problems asked in week 10. We gave these problems to students two weeks after the experiment (week 12). Some follow-up problems were identical to the corresponding week 10 problems and others had features of the problem changed, such as having a loop executing 50 times rather than 30. Students were not at ceiling performance in the post-test, suggesting that they did not remember the exact answers by week 12, so these were non-trivial measures of learning. Figure 3 shows the names of problems in week 10 and the corresponding problems in the week 12 follow-up activity. All of these problems were focused on analyzing the runtime of Python programs.

2.1 Experimental Factors

We experimentally varied two variables in a factorial experiment. Each time the student submitted an answer to the first problem of the exercise, they were randomly assigned to a condition for the Explanation factor and the Additional Problem factor.

The Explanation factor had three levels: absent (none), short, or long. The short explanation states, "The third answer is correct because the code inside the for loop takes constant steps regardless of len(s) and it will be executed len(s) times." The long explanation states "Suppose s = 'cat'. Then, double = double + ch * 2 will be executed 3 times because the for loop iterate through each character of s (i.e. 'c', 'a' and 't'). Now, suppose s = 'google'. Then double = double + ch * 2 will be executed 6 times. As you can see, if len(s) doubles, the number of steps also doubles. So, the third answer is correct."

The Additional Problem factor had two levels: absent (none) or present (one additional problem that was very similar to the problem students had attempted in asking them to trace through a for loop and determine its time complexity). A screenshot of this problem is shown in Figure 2.[1]

To measure how well a student performs on a problem, we used the number of attempts until the first correct answer. This is simply the number of submissions made before the

[1]These factors were varied in the context of a larger experiment with more factors that will not be described in this paper in the interest of space. We used weighted randomization in favour of not showing students additional activities to avoid overwhelming them with too many activities

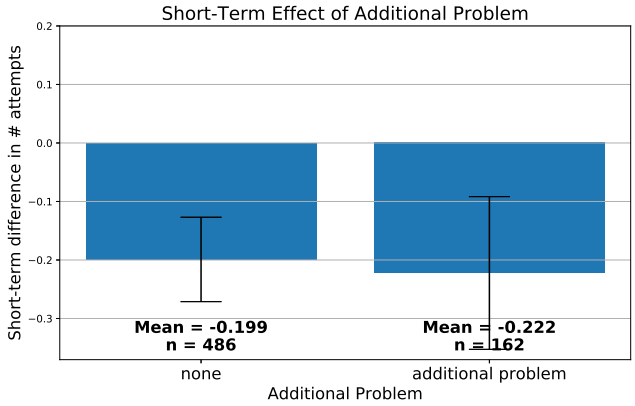

Figure 6: Effect of giving an additional problem on number of attempts to get the right answer on subsequent problems in the same exercise. T-test result: (t(646)=0.158, p=0.874)

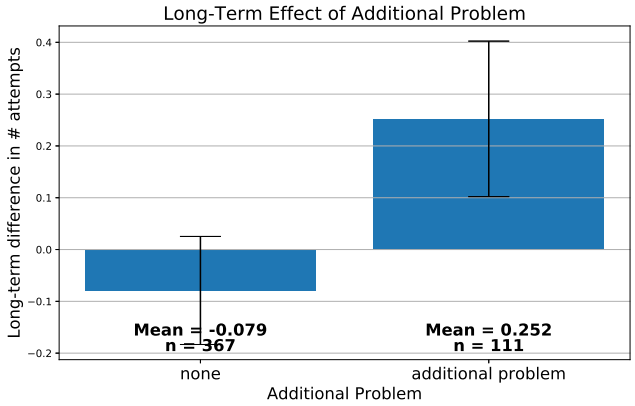

Figure 7: Effect of giving an additional problem on number of attempts to get the right answer on the same problem given two weeks later. T-test result: (t(476)=-1.602, p=0.110)

student selects all of the correct options and none of the incorrect options on the multiple-choice problem. For example, if a student gets the problem correct on their first try, their number of attempts is 1. If they get the first attempt wrong but the second attempt right, their number of attempts is 2.

To measure short-term improvement, we took the difference between the number of attempts on problem 1 of the week 10 exercise and the average number of attempts for the remaining four problems in that exercise. This number can be negative if students did worse on the remaining problems than they did on the problem we deployed the supports on, and the higher the number, the greater the improvement.

To measure improvement on the delayed exercise, we took the difference between the number of attempts on problem 1 of the week 10 exercise and the number of attempts on the exact same problem when presented in the week 12 follow-up.

3. RESULTS AND DISCUSSION

In this section, we first show a lack of evidence for an improvement in performance between the problem we added student support to in week 10 and the same problem given in a follow-up exercise in week 12. Next, we analyze the effects of the Explanation and Additional Problem factors. Our results did not reach the significance threshold of $p < 0.05$, and as such they should be interpreted with caution. We present suggestive evidence that the explanations were helpful on the same homework exercise (t(490)=-1.24, p=0.215), but not on the follow-up test two weeks later (t(364)=0.300, p=0.764). Finally, we show the reverse trend with the Additional Problem: it was not helpful in the same homework exercise (t(646)=0.158, p=0.874) but might have been in the follow-up exercise two weeks later (t(476)=-1.602, p=0.110). We interpret how these results can inform instructors' design choices and address possible limitations of the work.

3.1 Minor improvement on the same problem

Students took only slightly fewer attempts to get the problem correct in week 12 compared to week 10. While they took 2.07 attempts on average to get the answer right in week 10, they took 2.02 attempts to get the answer right on week 12, even though they had completed the same problem just two weeks before. We found little evidence that students improved between solving a problem in week 10 and solving the same problem in week 12 (t(1136)=-0.529, p=0.597). This suggests that students might not have remembered the answer to the problem even when they already solved it two weeks earlier, meaning testing them on the same problem in week 12 appears to be a non-trivial measure of their understanding of the same concepts.

3.2 Explanations might have helped in the short term

We found no statistically significant difference between students who were given explanations and those who were not. However, the results suggest that when students were given explanations, they took slightly fewer attempts to get the right answer in subsequent problems than those who did not, regardless of whether they saw a short (t(490)=-1.24, p=0.215) or long explanation (t(491)=-1.29, p=0.195), as shown in Figure 4. However, the effect of seeing explanations was much smaller in the long term, as the sample means were similar in all three conditions. This is shown in Figure 5 and suggests that an explanation in a homework context might be useful only during that homework session. This could have happened because the problems tested a procedural skill, namely runtime analysis. While reading an explanation gives students a clear formula they can apply in subsequent runtime analysis, they might forget that formula when they stop working on their homework and lose the benefit of the explanation.

3.3 Additional Problems might have helped in the long term

Similarly to the Explanation factor, we did not find statistically significant evidence for a difference in means for the

Additional Problem factor. However, we still found suggestive evidence that giving an additional problem has an effect in the long term. We did not find evidence for a difference between the performance of students who were shown an additional problem and those who were not on subsequent problems in the same homework exercise, but students who received the additional problem took fewer attempts in the post-test (t(476)=-1.602, p=0.110). These results are shown in Figures 6 and 7 respectively.[2] This difference might suggest that the value of the additional practice problem was primarily as a memory aid. Doing more problems could have helped students remember the skill they learned better when writing the post-test. If this knowledge is already in their minds when they are doing the exercise, it makes sense that they did not benefit immediately from more practice. However, they might remember more when writing the post-test, which would explain the improvement in performance there.

3.4 Limitations

A notable limitation of this work was the lack of statistical signficance. However, the results are consistent with each other and align with ideas from the ICAP framework. As such, they suggest a trend that could inform future research. In the interest of open and replicable science, it can be valuable to publish suggestive and negative results that do not meet the threshold for statistical significance. Real-world data is often messy and suggestive results can reveal crucial new directions for analysis.

Another limitation of this work is that the problems in the week 12 follow-up were not all identical to those in the week 10 homework. They tested the same concepts and some were exact copies, but others were slight modifications of problems on the original homework. Therefore, the observed results might be due to the supports having different degrees of relevance to the problems in week 10 and week 12 rather than the duration between support and post-test. We have mitigated this by using the differences between number of attempts on the problem we applied the explanations and additional problem to and the relevant subsequent problems as dependent variables, so if one intervention improves students' score on problem 1 both in week 10 and week 12, that would be reflected in that the changes to both scores cancel out when the difference is computed.

One might also raise the concern that we had different sample sizes in different conditions. More students were assigned to the "none" condition than other conditions for both the Explanation and Additional Problem factors. We intentionally weighted the randomization in this way to minimize the burden on students from having too many additional activities, a strategy used in randomized clinical trials in the medical field.

Considering that the effects of reading explanations and solving problems might vary widely with context, such as the week of a course in which supports were given, it is unclear how broadly the trends we identify in our data apply. While it appears possible that giving students more practice problems helps them develop lasting procedural knowledge of how to analyze the runtime of an algorithm, it is not clear that we can conclude the same about different tasks in computer science education like learning the syntax of a programming language or how to design an algorithm.

Finally, the week 12 post-test was optional, so dropout is a concern. While 648 students completed the exercise in week 10, only 478 completed the follow-up post-test in week 12. Therefore, the conclusions we draw about the effects of educational supports in the long term might reflect only the population of students who choose to complete the post-test. Though this was the majority of students, the reported effect could be different if, for instance, the students who are unlikely to do optional homework problems in week 12 are also unlikely to attempt an optional problem given to them in their week 10 homework exercise.

4. CONCLUSION AND FUTURE WORK

Our experiment investigated the effects of explanations and additional problems on performance both on a post-test and subsequent problems on the same test. We found intriguing but not definitive insight into the effects of explanations. The mean number of attempts for students who saw either a short or long explanation was lower than those who saw no explanation, but this difference was not statistically significant (t(490)=-1.24, p=0.215). It is possible that the explanations we showed students simply did not have an effect on their learning in either the short or long term. It could be that the explanations used in this experiment did not benefit students as much as they could have and effort should be directed to designing better explanations. Alternatively, it is possible that the explanations helped students somewhat on the remaining problems in the homework exercise. If this result were replicated in a larger study, it would be interesting because it could guide instructors in deciding how to effectively incorporate instructional explanation into their courses.

In exploring the effect of additional problems, we found that the mean number of attempts on the equivalent post-test problem was lower for students who were shown an additional problem than those who were not. This difference was not statistically significant, though we found stronger evidence for it than we found for explanations (t(476)=-1.602, p=0.110). Like with the explanations, it is possible that the additional problem we gave students was truly not effective and future work should focus on how to design more effective practice problems. However, if the long-term improvement as a result of the additional problem is replicated in subsequent large-scale experiments, it could provide guidance for instructors in deciding how to incorporate practice problems into their courses effectively.

If the results reported above reflect a real effect, they suggest that explanations are helpful in the short term, but not in the long term. Conversely, additional problems are helpful in the long term, but not in the short term. This aligns with what one might expect based on the ICAP framework, as solving a problem qualifies as deeper engagement with the learning material than reading an explanation. Instructors likely care more about whether their students retain information in the long term rather than whether they un-

[2]After this analysis, we noticed that the control and experimental groups had different variances, which violates the assumption of the standard t-test. We then ran Welch's t-test and found a p-value of 0.07. (t(476)=-1.813, p=0.0712)

derstand concepts immediately, so focusing on the long-term learning measure makes sense.

The possible difference between the effects of these interventions is interesting and motivates further research into how immediate and delayed effects of reading explanations and solving problems might differ. This might help guide instructors in thinking about the trade-offs involved in deciding when to give explanations to students and when to give them more problems.

Future work should investigate how generally this pattern holds. The part of the course we deployed these supports on focused on the procedural skill of reading an algorithm and analyzing its time complexity. Would giving an additional problem still be effective in teaching a different concept in the course, such as the difference between for and while loops? Perhaps additional problems are more helpful in developing procedural knowledge while good explanations might be more effective in building propositional knowledge. By running similar experiments at different points in the introductory computer science course, we hope to learn more about which types of student support are helpful in developing different skills.

Additionally, we are interested in investigating whether the effects of these interventions differ across subgroups of students. One reason why we might not see a large average effect is that the effectiveness of different forms of support could vary significantly across students. Perhaps, for example, students who take a programming course out of intrinsic interest are more likely to benefit from an additional practice problem than those who take it to satisfy a breadth requirement. By analyzing this experimental data jointly with contextual variables derived from surveys and data mining, we hope to provide a richer picture of which forms of support work for which students and how instructors can tailor interventions more precisely to individual students' needs.

5. REFERENCES

[1] C. J. Butz, S. Hua, and R. B. Maguire. A web-based bayesian intelligent tutoring system for computer programming. *Web Intelligence and Agent Systems: An International Journal*, 4(1):77–97, 2006.

[2] M. T. Chi. Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in cognitive science*, 1(1):73–105, 2009.

[3] M. T. Chi and R. Wylie. The icap framework: Linking cognitive engagement to active learning outcomes. *Educational psychologist*, 49(4):219–243, 2014.

[4] J. L. Chiu and M. T. Chi. Supporting self-explanation in the classroom. *Applying science of learning in education: Infusing psychological science into the curriculum*, pages 91–103, 2014.

[5] M. Feng, N. T. Heffernan, and J. E. Beck. Using learning decomposition to analyze instructional effectiveness in the assistment system. In *AIED*, 2009.

[6] R. Hosseini, T. Sirkiä, J. Guerra, P. Brusilovsky, and L. Malmi. Animated examples as practice content in a java programming course. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, SIGCSE '16, page 540–545, New York, NY, USA, 2016. Association for Computing Machinery.

[7] J. Kay, M. Barg, A. Fekete, T. Greening, O. Hollands, J. H. Kingston, and K. Crawford. Problem-based learning for foundation computer science courses. *Computer Science Education*, 10(2):109–128, 2000.

[8] D. S. McNamara, T. O'Riley, and R. S. Taylor. Classroom based reading strategy training: Self-explanation vs. a reading control. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 28, 2006.

[9] N. Rummel, M. Mavrikis, M. Wiedmann, K. Loibl, C. Mazziotti, W. Holmes, and A. Hansen. Combining exploratory learning with structured practice to foster conceptual and procedural fractions knowledge. Singapore: International Society of the Learning Sciences, 2016.

[10] J. J. Williams and T. Lombrozo. The role of explanation in discovery and generalization: Evidence from category learning. *Cognitive Science*, 34(5):776–806, 2010.

[11] J. Wittwer, M. Nückles, and A. Renkl. Improving human tutoring by improving tutor-generated explanations. In *Avoiding Simplicity, Confronting Complexity*, pages 359–368. Brill Sense, 2006.

[12] J. Wittwer and A. Renkl. Why instructional explanations often do not work: A framework for understanding the effectiveness of instructional explanations. *Educational Psychologist*, 43(1):49–64, 2008.