

# Providing the Option to Skip Feedback – A Reproducibility Study

Amruth N. Kumar<sup>[0000-0002-1951-3995]</sup>

Ramapo College of New Jersey, Mahwah NJ 07430, USA  
amruth@ramapo.edu

**Abstract.** Would providing choice lead to improved learning with a tutor? We had conducted and reported a controlled study earlier, wherein, introductory programming students were given the choice of skipping the line-by-line feedback provided after each incorrect answer in a tutor on if/if-else statements. Contrary to expectations, the study found that the choice to skip feedback did not lead to greater learning. We tried to reproduce these results using two tutors on if/if-else and switch statements, and with a larger subject pool. We found that whereas choice did not lead to greater learning on if/if-else tutor in this reproducibility study either, it resulted in decreased learning on switch tutor. We hypothesize that skipping feedback is indeed detrimental to learning. But, inter-relationships among the concepts covered by a tutor and the transfer of learning facilitated by these relationships compensate for the negative effect of skipping line-by-line feedback. We also found contradictory results between the two studies which highlight the need for reproducibility studies in empirical research.

**Keywords:** Skipping feedback, Worked example tutor, Reproducibility study

## 1 Introduction

Findings are mixed on the effect of choice on learning (e.g., [4,5]). We had conducted a study on providing students of introductory programming courses the choice to skip feedback in a tutor on code-tracing [2]. We had noted two factors that could affect the outcome of the study:

- The programming concepts covered by the tutor were inter-related. The tutor provided line-by-line explanation of the correct solution in the style of worked examples [6] as feedback, and this explanation has been shown to improve learning [3]. Reading the feedback on one concept had the potential to help students also learn about other inter-related concepts. So, we expected students to be able to skip reading feedback on some problems without hampering their learning.
- The study was conducted while the tutor was being used for after-class assignment in introductory programming courses. In this unsupervised setting, some students may be motivated to maximize learning while others may be motivated to complete the assignment as quickly as possible. So, students may exercise the option to skip feedback for varying reasons - some related to learning, while others are not. Those

who skip feedback for expediency may hamper their learning by skipping feedback.

In the study, we found that providing choice did not lead to greater learning [2]. Students who had the choice to skip feedback needed marginally more problems to learn each concept, and their pre-post improvement was marginally less than that of those who did not have the choice. We tried to reproduce the results of this study.

Reproducibility is a core principle of scientific research. Reproducibility refers to the ability to draw the same results using different instruments, methods, protocols and/or participants [1]. The parameters of our reproducibility study were as follows:

- **Instrument:** Whereas the earlier study had used a single tutor on `if/if-else` statements, in this study, we used that tutor as well as a tutor on `switch` statements. Whereas `if/if-else` tutor presented only code-tracing problems, `switch` tutor presented problems on code-tracing as well as code-debugging. Both the tutors were adaptive, and presented feedback consisting of line-by-line explanation of the correct solution when a student's solution was incorrect.
- **Subjects:** In both the studies, the subjects were students in introductory programming courses. The earlier study was conducted in Fall 2015. The reproducibility study was conducted using `if/if-else` tutor in three subsequent semesters: Spring 2016-Spring 2017, and `switch` tutor in four semesters: Fall 2015 – Spring 2017.

Both the studies were controlled, and used the same pretest-practice-post-test protocol. Experimental group subjects were given the choice to skip the line-by-line explanation feedback whereas control group students were not. Both the studies were conducted *in-natura*, i.e., under unsupervised conditions in real-life introductory programming courses where the tutors were used for after-class assignments.

## 2 The Reproducibility Study

**Participants:** The tutors on `if/if-else` and `switch` statements were used by students in introductory programming courses from multiple institutions that were randomly assigned to control or experimental group each semester. Table 1 lists the number of students in control group (no choice to skip feedback) and experimental group (choice to skip feedback) for the two tutors who granted IRB permission.

**Table 1.** Number of participants in the study under each treatment.

| Tutor                   | Control Group | Experimental Group |
|-------------------------|---------------|--------------------|
| <code>if/if-else</code> | 528           | 322                |
| <code>Switch</code>     | 142           | 221                |

**Instruments:** The tutor on `if/if-else` statement presents code-tracing problems. In each problem, the student is asked to identify the output of a program containing one or more `if/if-else` statements, one output at a time, along with the line in the program that produced that output. If the student's answer is incorrect, the tutor provides

line-by-line explanation of the correct answer [3]. The tutor covers 12 concepts on one-way (`if`) and two-way (`if-else`) selection statements.

The tutor on `switch` statement presents both code-tracing and code-debugging problems. In a code-debugging problem, a program containing a `switch` statement is presented and the student is asked to identify the line, code object and the specific syntax/semantic error applicable to the code object on the line. If the student's answer is incorrect, the tutor explains the genesis of the error contained in the program. The tutor covers 12 concepts.

Both the tutors cover C++, Java, and C#. Both are accessible over the web. They are part of a suite of problem-solving tutors for introductory programming topics called problets ([www.problets.org](http://www.problets.org)). Typically, students use the tutors as after-class assignments, often multiple times till they have mastered all the concepts in the topic.

**Protocol:** Every time a software tutor is used, it administers pretest-practice-post-test protocol as follows:

- **Pretest:** During pretest, the tutor presents one problem per concept to prime the student model. If a student solves a problem correctly, no feedback is provided to the student. On the other hand, if the student solves a problem partially correctly, incorrectly, or opts to skip the problem without solving it, line-by-line explanation is presented to the student.
- **Adaptive Practice:** Once the student has solved all the pretest problems, practice problems are presented on only the concepts on which the student skipped solving the problem or solved the problem partially/incorrectly during pretest. On each such concept, the student is presented multiple problems until the student has mastered the concept, i.e., solved a minimum percentage (e.g., 60%) of the problems correctly. After each incorrectly solved problem, the tutor presents line-by-line explanation of the correct answer.
- **Adaptive Post-test:** During this stage, which is interleaved with practice, the student is presented a test problem on each concept already mastered by the student during practice.

Pretest, practice and post-test are administered back-to-back without interruptions, entirely over the web by the tutor. The entire protocol is limited to 30 minutes. Since this was a controlled study, experimental group had the option to skip the line-by-line explanation provided after the student had either skipped solving a problem or solved the problem incorrectly/partially, whereas control group did not. Students who skip solving the pretest problem on a concept or solve it partially/incorrectly, solve enough problems during practice to master the concept, and solve the post-test problem on the concept correctly are said to have **learned** the concept.

The grade on each code-tracing problem was normalized to  $0 \rightarrow 1.0$ . Code-debugging problems were graded as correct or incorrect (no partial grade). If a student used a tutor multiple times, we considered data from the session when the student had learned the most number of concepts. If the student did not learn any concepts, we considered data from the first session when the student had solved the most number of problems.

In order to account for the 30-minute limit placed on each session, the variables of the study were designed to be insensitive to the number of problems solved. They were:

- Pretest score per problem to verify that the control and experimental groups were comparable;
- The time spent per pretest problem - to assess the impact of treatment on the pace of solving problems during pretest;
- The number of concepts learned as a measure of the amount of learning;
- The number of practice problems solved per learned concept, as a measure of the pace of learning. It was calculated by dividing the number of practice problems solved on all the learned concepts by the number of concepts learned;
- Pre-post change in grade per learned concept as a measure of improvement in learning.

The fixed factor was treatment: whether students did or did not have the option to skip line-by-line explanation.

### 3 Results and Discussion

**if/if-else Tutor Results:** One-way ANOVA analysis of the pretest score per problem and the time spent per pretest problem yielded no significant main effect for treatment. *So, the two groups were comparable.* Analysis of the number of concepts learned yielded no significant main effect for treatment. *So, the treatment did not lead to greater learning.* The number of practice problems solved per learned concept was not significantly different between the two groups. *So, the treatment did not affect the pace of learning.*

But, a significant difference was observed on the **pre-post change in score** on the learned concepts between control and experimental subjects [ $F(1,348) = 5.797$ ,  $p = 0.017$ ]; pre-post improvement was  $0.844 \pm 0.03$  for control subjects as compared to  $0.902 \pm 0.038$  for experimental subjects. *So, treatment led to greater improvement in score on learned concepts.*

**switch Tutor Results:** One-way ANOVA analysis of the pretest score per problem and the time spent per pretest problem yielded no significant main effect for treatment when only those who learned at least one concept were considered. *So, the two groups were comparable.*

Analysis of the **number of concepts learned** yielded significant main effect for treatment [ $F(1,177) = 4.816$ ,  $p = 0.03$ ]; among those who learned at least one concept, control subjects ( $N = 65$ ) learned  $2.877 \pm 0.33$  concepts whereas experimental subjects ( $N = 133$ ) learned  $2.416 \pm 0.25$  concepts. *So, overall, the option to skip feedback led to significantly less learning.*

The number of practice problems solved per learned concept was not significantly different between the two groups. *So, the treatment did not affect the pace of learning.* No significant difference was observed in the pre-post change in score on the learned concepts between control and experimental subjects.

Table 2 compares the results obtained in the earlier study with those from this reproducibility study. In the table, empty cells correspond to no significant difference found between treatments. Parenthesized results are only marginally significant.

**Table 2.** Comparison of the results from the two studies

| Variable             | if/if-else Tutor |                | switch Tutor   |
|----------------------|------------------|----------------|----------------|
|                      | Earlier Study    | Reprod. Study  | Reprod. Study  |
| Pretest Time         | Control > Exp.   |                |                |
| Concepts Learned     |                  |                | Control > Exp. |
| Practice per Concept | (Exp. > Control) |                |                |
| Pre-post change      | (Control > Exp.) | Exp. > Control |                |

The **time spent per pretest problem** was found to be significantly greater for control than experimental group in the earlier study. In the reproducibility study, even when the same if/if-else tutor was used, no significant difference was found between treatments. One explanation for this might be that experimental subjects in the reproducibility study skipped feedback on a small percentage of the solved problems: 9.09% (342 problems out of 3761 solved) in if/if-else tutor and 11.67% (501 problems out of 4295 solved) in switch tutor. So, even though they saved time when they skipped feedback, the time saved skipping feedback was small compared to the time the subjects spent solving problems.

No significant difference was found between treatments on the **number of concepts learned** in both the earlier study and the reproducibility study that used if/if-else tutor. But, in the reproducibility study that used switch tutor, experimental subjects learned significantly less than control subjects. One explanation might be that the concepts on switch statements are less inter-related than those on if/if-else statements: whereas if/if-else tutor presented only code-tracing problems, switch tutor presented both code-tracing and debugging problems, with no conceptual overlap between the two types of problems. So, there was less transfer of learning among concepts on switch than on if/if-else. Therefore, when students skipped feedback and did not benefit as much from transfer of learning, they ended up learning less. If this explanation is true, skipping feedback is indeed detrimental to learning. But, *interrelationships among the concepts covered by a tutor and the transfer of learning facilitated by these relationships compensate for the negative effect of skipping feedback*. This is a hypothesis worth testing in the future.

In the earlier study, experimental subjects solved marginally more **practice problems per learned concept**. But, in this reproducibility study, we did not find any difference between treatments with either tutor. Since the result of the earlier study was only marginally significant, it may have been an artifact of the student population that should have been ignored in the previous study.

The final difference in Table 2 is on **pre-post change in score on learned concepts**. Whereas pre-post increase in score was marginally greater for control group subjects in the earlier study, it was significantly greater for experimental group subjects in the reproducibility study *that used the same if/if-else tutor*. The two results are clearly

contradictory. One possible explanation for the contradictory results is that data selection criteria differed between the two studies. In the previous study, when a student used the tutor multiple times, data was considered from “only the first time when the student had solved all the pretest problems.” In the current study, the session in which the student had learned the most concepts was chosen. If a student skips solving a pretest problem, it is marked as not attempted and would make the session less likely to be selected according to the criterion used in the previous study. The student could still go on to solve practice problems and learn several concepts, which would make the session more likely to be selected in the current study. Yet, we do not expect this difference in criteria to have affected more than a handful of students: usually, the more problems a student solved, the more concepts the student learned.

While the reason behind the contradictory results remains to be investigated further, the contradictory results themselves highlight the need for reproducibility studies in empirical research. Reproducibility studies might open up new research questions, expose nuances that the original work may not have considered, or buttress earlier results with additional empirical support.

Based on the earlier study and this reproducibility study, we conclude that providing the option to skip feedback does not increase learning. On the other hand, if the concepts covered by the tutor are not inter-related, the option to skip feedback will result in decreased learning.

## Acknowledgments

Partial support for this work was provided by the National Science Foundation under grant DUE-1432190.

## References

1. Drummond, C.: Replicability is not Reproducibility: Nor is it Good Science. In Proc. Evaluation Methods for Machine Learning Workshop, 26<sup>th</sup> ICML, Montreal, Canada. (2009)
2. Kumar, A.N.: Providing the Option to Skip Feedback in a Worked Example Tutor. In Proc. Intelligent Tutoring Systems (ITS 2016), LNCS 9684, pp. 101-110. Springer, Heidelberg. (2016)
3. Kumar, A.N.: Explanation of step-by-step execution as feedback for problems on program analysis, and its generation in model-based problem-solving tutors. Technology, Instruction, Cognition and Learning (TICL). Special Issue on Problem Solving Support in Intelligent Tutoring Systems, vol. 4(1) (2006)
4. Meyer, B.J.F., Wijekumar, K., Middlemiss, W., Higley, K., Lei, P.W., Meier, C. and Spielvogel, J.: Web-based Tutoring of the Structure Strategy with or without Elaborated Feedback or Choice for Fifth- and Seventh-Grade Readers. Reading Research Quarterly, vol. 45(1), pp. 62-92. (2010)
5. Ostrow, K.S., Heffernan, N.T.: The role of student choice within adaptive tutoring. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) Artificial Intelligence in Education. LNCS 9112, pp. 752–755. Springer, Heidelberg. (2015)
6. Sweller, J., and Cooper, G.A.: The use of worked examples as a substitute for problem solving in learning algebra. Cognition and Instruction, vol. 2, pp. 59–89. (1985)