Comparing the Impacts of Visually Grouped and Jumbled Distractors on Parsons Problems in CS1 Assessments

David H. Smith IV
University of Illinois
Urbana, IL, USA
dhsmith2@illinois.edu

Seth Poulsen
Utah State University
Logan, UT, USA
seth.poulsen@usu.edu

Max Fowler
University of Illinois
Urbana, IL, USA
mfowler5@illinois.edu

Craig Zilles
University of Illinois
Urbana, IL, USA
zilles@illinois.edu

ABSTRACT

Parsons problems are a commonly used problem type typically used in introductory computer science courses. They involve organizing blocks containing segments of code to form a program. These questions often use "distractors" which are plausible, but incorrect, blocks of code. In Parsons problems distractors are often included either by jumbling them in alongside the correct response options or visually grouping them with their correct alternatives. In this study, we investigate the impact of both jumbled and visually grouped distractors on: 1) student performance, 2) the amount of time students spent on the question, and 3) the item's quality in exams and quizzes in a CS1 Python course. Our findings indicate that the inclusion of distractors, both visually grouped and jumbled, have a marginal impact on item quality while increasing the amount of time needed to solve the problem and reducing students' performance. Though visually grouped distractors appear to mediate the amount of time students spend relative to their jumbled counterparts, their effect is not so large as to fully alleviate concerns related to the increase in duration.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

Parsons Problems, CS1, Distractors, Classical Test Theory

ACM Reference Format:
David H. Smith IV, Seth Poulsen, Max Fowler, and Craig Zilles. 2023. Comparing the Impacts of Visually Grouped and Jumbled Distractors on Parsons Problems in CS1 Assessments. In *Proceedings of the ACM Conference on Global Computing Education Vol 1 (CompEd 2023), December 5–9, 2023, Hyderabad, India.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3576882.3617927

CompEd 2023, December 5–9, 2023, Hyderabad, India

ACM ISBN 979-8-4007-0048-4/23/12…$15.00
https://doi.org/10.1145/3576882.3617927

1 INTRODUCTION

Parsons problems were first introduced by Parsons and Haden [30] as "Parsons programming puzzles" in order to give students a more engaging form of programming drills. They have been found to be effective for teaching common topics such as common programming patterns more efficiently than code writing drills and with less cognitive load [13, 14]. Subsequent work has also found them to be effective as exam items, measuring skills similar to traditional programming problems but allowing for faster, more consistent grading [5]. Given these affordances, Parsons problems have become a popular tool in introductory computer science courses, gaining a great deal of traction in recent years [11, 16].

Since their introduction, Parsons problems have included distractors designed to reflect common errors and misconceptions students might encounter while learning to write code [30]. Distractors are utilized in a variety of assessment items, most commonly multiple-choice questions where the effectivness of a question's distractors strongly influences both the difficulty of the item and the item's ability to discriminate between high and low performing students [28, 35]. Parsons problems are similar in that distractors are incorrect blocks of code presented alongside correct blocks with the intention of adding an additional dimension of difficulty to the problem [6, 21]. However, unlike multiple-choice questions, Parsons distractors have another presentation consideration: whether they are jumbled randomly with the correct blocks of code (Figure 1b) or presented such that distractors are always explicitly grouped with their associated correct block (Figure 1c). To date, there has been little research seeking to identify differences between visually grouped and jumbled distractors in formative and summative contexts to best make use of them [11, 16].

Given these gaps in the literature, we seek to answer the following research questions:

- RQ1. What is the impact of including visually grouped distractors compared to jumbled distractors on students' scores?
- RQ2. What is the impact of including visually grouped distractors compared to jumbled distractors on the amount of time students spend on questions?
- RQ3. To what degree does the inclusion of jumbled and visually grouped distractors improve an item's quality compared to not including distractors?

In filling these gaps, we hope inform the design and usage of distractors of Parsons problems in summative assessments.

2 BACKGROUND

2.1 An Overview of Parsons Problems

In a review of the literature on Parsons problems by Du et al. [6], they identified four motivating areas the drive the investigation and the utilization of Parsons problems. First, instructors can more easily analyze students responses in order to determine the errors students make and the difficulties they face. Second, when using computer-based Parsons problems, students can submit their responses and receive immediate feedback on their correctness. Third, and inline with the original motivations for the creation of the problem type, is improving students' engagement. Finally, by asking students to organize blocks of existing code, Parsons problems reduce cognitive load while still allowing students practice skills related to programming (e.g., planning, patterns).

There have been a number of iterations on Parsons problems each aiming to improve their effectiveness in as an instrument for teaching and learning. Faded parsons problems add an additional dimension of difficulty to the problems by removing certain portions of the code blocks, which requires students to correctly complete the code in addition to organizing the blocks correctly [18, 36, 37]. To help students manage cognitive load and improve engagement, "Adaptive Parsons problems" have been introduced wherein a student can use a hint button to merge blocks and remove distractors, in order to reduce the difficulty of the problem after some number of failed attempts [10, 13, 22]. Recently "Micro Parsons Problems", which ask students to arrange blocks horizontally, have been introduced as a method of teaching students concepts which involve constructing a single line of code, such as SQL and regex [38, 39].

Much of the work on Parsons problems has been done with their usage in formative contexts primarily in mind [12, 17]. Here, considerations such as engagement and learning are the primary objectives. However, in a summative context, issues such as what the items measure and the information the questions provide are often given priority. Prior work has also indicated that there was a strong correlation between performance code writing and Parsons problems indicating that they measure similar skills [5, 17, 25]. This, in combination with the easier grading that Parsons problems facilitates [6, 12], provides a strong argument in favor of using Parsons problems on CS assessments.

2.2 Visually Paired vs Jumbled Distractors

The most commonly discussed form of visually grouped distractors are "visually paired", where one distractor is paired with the correct, alternative option. Denny et al. [5] compared paper-based Parsons problems containing no distractors to those with distractors that were either "jumbled" (Figure 2a) or "visually paired" (Figure 2b). For these problems, students were expected to look at the list of code segments and use them to construct a solution on paper. Think-aloud interviews with students revealed that the jumbled distractors tended to overwhelm students compared to the visually paired distractors. This makes intuitive sense, as pairing distractors with their correct alternatives reduces the effort necessary for students to compare related correct and distractor blocks. These findings have lead to the recommendation that distractors should be visually associated with their correct alternatives [12].

More recently, tools have emerged for providing Parsons problems in a computer-based environment [15, 24]. However, there has been little follow-up work comparing the two distractor conditions in a computer-based environment. Compared to the paper-based tests presented by Denny et al. [5], the computer-based Parsons problems can include syntax highlighting and allow students to drag and reorder blocks which could allow students to perform the grouping of similar blocks of code themselves. Both in general and for computer-based environments in particular, better understanding the impacts of paired and unpaired distractors is important for validating the recommendation that distractors be visually paired.

2.3 The Relationship Between Distractors, Item Quality, and Test Quality

Looking beyond the design considerations of distractors and their impact on students in a formative environment, there remains the question of whether distractors provide a useful signal for differentiating students who have a poor understanding of the material from those who do not. Answering this question is integral to understanding: 1) if distractors should be included at all and 2) selecting between jumbled and visually grouped distractors [19]. Though Denny et al. [5] found that students were more overwhelmed by jumbled distractors, it is possible that having to manually differentiate correct blocks of code from the jumbled distractors evaluates an additional skill that is not being measured by visually grouped distractors.

The issues of measuring test and item quality are complex and have been studied for over half a century [8]. Among the commonly used metrics for evaluating test quality is "test reliability", which measures the degree to which items in an exam are interrelated. This in turn reflects the degree to which the assessment measures the underlying trait or traits it purports to measure [7, 8, 27]. One of the most common metrics for evaluating test reliability is Cronbach's alpha [4]. Two commonly suggested methods of improving a test's reliability are to: 1) increase the number of items on the assessment and 2) improve the quality of the items themselves [1, 7, 8]. The former of these two options is often less feasible than the latter. Improving the reliability of an assessment would require a substantial, and likely unreasonable, number of items to be added to the assessment [7].

This then raises the question of what constitutes a high quality item. Similar to test reliability, items of high quality are those that measure the trait the individual who constructed the question intended to measure. We would expect an individual who performs well on the item to also perform well on the test, and vice versa. Thus, the quality of an item can be measured by the correlation between the item score and the test score. This correlation, commonly calculated as a Pearson's product-moment correlation, is known as the item discrimination index in Classical Test Theory [2, 9, 32]. By improving the discrimination of individual items, the overall reliability of the test can be improved as well [8]. Embedded within this is the assumption that there is enough variance in responses that a meaningful correlation can be calculated. This indicates that questions should also be designed to be neither too easy nor too difficult [8, 23, 26].

(a) The interface for answering questions with no distractors (ND).

(b) The interface for answering questions with jumbled distractors (JD).

(c) The interface for answering questions with grouped distractors (GD).

Figure 1: The interface as it appears for each version of the Parsons problems used in this study.

```
public void printStringNTimes(word, n)
for (int i = 0; i < n; i++)
public void printstringntimes(string word, int n)
for (int i = 1; i < n; i++)
println(word);
System.out.println(word);
```

(a) Parsons problem with Jumbled Distractors

```
System.out.println(word);
println(word);

for (int i = 0; i < n; i++)
for (int i = 1; i < n; i++)

public void printstringntimes(string word, int n)
public void printStringNTimes(word, n)
```

(b) Parsons problem with Paired Distractors

Figure 2: Examples of Parsons Problem Formats from Denny et al. [5]

With this in mind, we can consider the impact of distractors on test quality. Distractors have been primarily used and included in multiple choice questions where they bear the nearly sole responsibility for influencing an item's difficulty and discrimination [3, 19, 20]. This makes the selection of effective distractors an issue of great importance in this context. However, in Parsons problems, even if distractors are not included in the question, students must still correctly order the blocks to produce a correct solution. Additionally, in what are refered to as "two-dimensional" Parsons problems students must also provide the correct indentation of the block, a task intended to emulate languages such as Python where indentation is semantic rather than stylistic [6, 12]. Even if distractors are not included in Parsons problems, providing a correct solution is still a non-trivial task. This raises the question of how significant a role distractors play in influencing the item quality of Parsons problems.

3 METHODS

This study involves a between-subjects, randomized experiment. Students in the Fall 2022 semester (n = 576) were randomly assigned to one of two conditions: jumbled distractors (JD) or no distractors (ND). Students in the Spring 2023 semester (n = 456) were randomly assigned to either the visually grouped distractors (GD) or no distractors (ND) condition. The interface for ND, JD, and GD conditions are shown in Figure 1. In both semesters, the ND questions were kept largely the same and students' scores on those questions did not differ significantly between the two semesters (t=-1.63, p=0.11).

Each semester included four quizzes and four exams which were unproctored and proctored, respectively. All quizzes and exams were computer-based, with quizzes taken on the students' own devices and exams by scheduling a time in a proctored computer lab [40]. In each semester, 32 question pairs were used in the exams, and the questions not containing distractors were kept largely identical throughout both semesters with the exception of two items pairs which were replaced.

The Parsons problems included in this study were a mix of questions designed by the researchers (n = 24) and ones taken from the "Python for Everybody" interactive textbook on Runestone Academy (n = 8) [15, 29]. In either case, questions in the JD and GD conditions contained 3-4 distractors. For questions taken from Runestone, the original distractors were used except in cases where

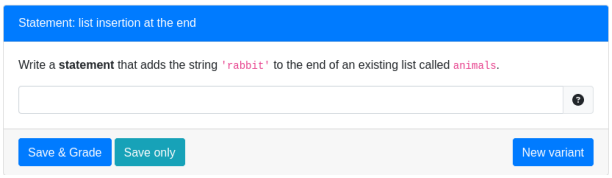


Figure 3: Example of a statement question.

additional distractors were needed in order to meet this threshold. The selection of distractors for the other questions included in this study is covered in Section 3.1. The analysis used to compare score, duration, and item-quality are covered in Section 3.2. The analysis was repeated independently for both sets of questions (Runestone and those we developed) and the results were qualitatively similar. As such, we present the results of the aggregated analysis.

3.1 Distractor Selection

Distractors were selected from the common mistakes made by students in code writing exercises [5, 21, 33]. In the context of the course in which this study was conducted, there exists a large corpus of responses to single line programming exercises which are collectively referred to as "statement questions" (Figure 3). These questions ask students to write a single line of code that will produce a desired effect (e.g., appending to a list). In previous work we analyzed a large number of these questions to identify common errors [33]. It is these errors which we used to generate distractors for this study.

3.2 Analysis

The analysis is broken into three parts. Section 3.2.1 operationalizes the metrics on which we will compare the various distractor conditions. Section 3.2.2 outlines the modeling approaches that will be used to examine the impact of the distractor conditions: 1) relative to having no distractors (ND) and 2) relative to each other (JD vs GD). Finally, in Section 3.2.3, we examine the impact of distractors on item quality using the lens of item-discrimination from Classical Test Theory as discussed in Section 2.3 of the literature review.

3.2.1 Metrics. For the analysis, we operationalize the following metrics:

- *Score:* Students on the platform are given three attempts to answer each question. The maximum score for each attempt are 6, 3, and 1 with partial credit being assigned on each attempt based on the edit distance between the student's answer and the correct answer based on the algorithm introduced by Poulsen et al. [31].
- *Duration:* The amount of time that the platform believes this question is the "active" questions. This is an imperfect proxy for how much time the student spent actively working on the problem.
- *Item-Discrimination:* The ability of an item to differentiate between a high and low performing student, which is a common measure of item quality in Classical Test Theory. We use the Pearson's r correlation coefficient to determine the item-test score correlation. As discussed in the literature this is a commonly used approach for determining item discrimination [1, 7].

3.2.2 Duration and Score Analysis. For determining the impact of including visually grouped and jumbled distractors on score and duration, we fit a Multi-Level Model (MLM) [34]. The purpose of using this modeling approach compared to a more standard linear regression is to help account for variance that might occur within the conditions (ND, JD, GD) and within student scores. Importantly, MLM allows us capture the unobserved variance in the conditions independent of unobserved variance in student scores, thereby allowing us to better model when outcomes are more impacted by student score or by question.

Regression 1 - Comparing No Distractors to Each Distractor Condition: The first regression is used to determine the impacts of including visually grouped and jumbled distractors relative to not including distractors. We fit the following pair of MLMs.

$$\text{Score}_{ij} = \beta_0 + \beta_1 \text{Condition}_{ij} + \beta_{2i} \text{S}_i + \text{Q}_j + \epsilon_{ij}$$
$$\text{Duration}_{ij} = \beta_0 + \beta_1 \text{Condition}_{ij} + \beta_{2i} \text{S}_i + \text{Q}_j + \epsilon_{ij}$$

Here, Score_{ij} and Duration_{ij} are the score and duration of student i on question j, respectively. Condition_{ij} is the condition student i was assigned to for question j (i.e., ND, JD, GD). S_i is included as a fixed effect to control for student ability on Parsons problems. Q_j is included as a random effect to control for question properties (i.e. average score or duration on a given question) and to help quantify how much variance in the model comes from variance between questions versus variance between student score within a question. ϵ_{ij} is the error term for student i on question j. Of the coefficients that result from these regressions, ones that are pertinant to the results are the intercept (β_0) which is the average for a given outcome on the ND condition and β_1 which is average outcome for each of the remaining two conditions relative to the intercept.

Regression 2 - Comparing JD to GD. To directly compare the impacts of visually grouped and jumbled distractors we fit the following second pair of MLMs.

$$\text{Score}_{ij} = \beta_0 + \beta_1 \text{Distractor}_{ij} + \beta_2 \text{Grouped}_{ij} + \beta_{3i} \text{S}_i + \text{Q}_j + \epsilon_{ij}$$
$$\text{Duration}_{ij} = \beta_0 + \beta_1 \text{Distractor}_{ij} + \beta_2 \text{Grouped}_{ij} + \beta_{3i} \text{S}_i + \text{Q}_j + \epsilon_{ij}$$

In this model the Condition_{ij} predictor is replaced with two binary predictors, Distractor_{ij} and Grouped_{ij} indicating if question j contains distractors and if those distractors are grouped. We choose to run this second set of models for a more fine-grained comparison, even though the models are functionally equivalent to the first set. In particular, reconfiguring the model in this fashion allows us to directly compare the impact of the JD and GD conditions, as opposed to comparing them against ND, enabling us to observe whether the differences between the GD and JD condition are statistically significant.

Measures of Effect Size and Model Fit. Standardized effect size for a given coefficient is calculated as

$$\text{Standardized Effect} = \frac{\beta}{\sqrt{\sigma}}$$

	Coefficient	SE	t-value	P
Intercept (ND)	4.92	0.59	8.37	<0.001***
JD	-0.55	0.20	-2.72	0.008**
GD	-0.22	0.21	-1.07	0.288
Standardized Effect Size (JD): -0.68				
Standardized Effect Size (GD): -0.27				
Intraclass Correlation (ICC): 0.25				

(a) Regression comparing the impact of JD and GD conditions on students' scores (points out of 6) relative to ND

	Coefficient	SE	t-value	P
Intercept (ND)	4.92	0.59	8.37	<0.001***
Distractors	-0.55	0.20	-2.72	0.008**
Grouped	0.33	0.21	1.56	0.121
Standardized Effect Size (Distractors): -0.68				
Standardized Effect Size (Grouped): 0.41				
Intraclass Correlation (ICC): 0.25				

(b) Regression comparing students' scores (in points out of 6) between the JD and GD conditions.

Table 1: The regressions indicate that in both cases, GD and JD lead to a decrease in score though the effect is only statistically significant in the JD condition and the impact was smaller in the GD condition.

	Coefficient	SE	t-value	P
Intercept (ND)	92.04	43.61	2.11	0.035*
JD	63.70	14.13	4.51	<0.001***
GD	44.98	14.35	3.13	0.002**
Standardized Effect Size (JD): 1.13				
Standardized Effect Size (GD): 0.80				
Intraclass Correlation (ICC): 0.23				

(a) Regression comparing the impact of JD and GD conditions on duration (in seconds) relative to ND.

	Coefficient	SE	t-value	P
Intercept (ND)	92.04	43.61	2.11	0.035*
Distractors	63.70	14.13	4.51	<0.001***
Grouped	-18.72	14.81	-1.26	0.209
Standardized Effect Size (Distractors): 1.13				
Standardized Effect Size (Grouped): -0.33				
Intraclass Correlation (ICC): 0.23				

(b) Regression comparing duration in JD and GD conditions with coefficients being duration in seconds.

Table 2: The regressions indicate that in both cases, GD and JD lead to a significant increase in duration compared to no distractors and GD resulted in a decrease from JD that was not statistically significant.

where β is a given coefficient from the results of one of the regressions, σ is the estimated within-question variance. We additionally compute the following intra-class correlation coefficient (ICC)

$$\text{ICC} = \frac{\tau^2}{\tau^2 + \sigma^2}$$

which is the ratio of variance between the questions (τ^2) and the total variance ($\tau^2 + \sigma^2$). This is used to determine if the majority of the variance comes from question level differences or are a result of the different distractor conditions.

3.2.3 Item-Discrimination Analysis. For item-discrimination, we compare the relative impact of including distractors for each item pair. A Shapiro-Wilks test was used to determine the normality of the item discrimination distributions (JD: W=0.86, p<0.001***; ND-Fall: W=0.98, p=0.73; ND-Spring: W=0.97, p=0.38; GD: W=0.97, p=0.42). As the majority of these tests indicate normally distributed data, we select a parametric test, the paired t-test, to determine if there was a significant difference in overall item discrimination between ND and JD in the Fall and ND and GD in the Spring.

4 RESULTS

4.1 Impact on Score (RQ1)

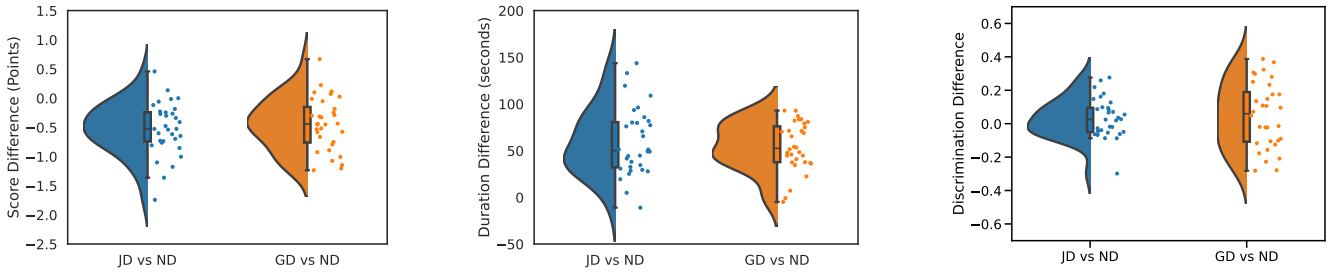
Observing the raw differences displayed in Figure 4a, both distractor conditions, JD and GD, increased the difficulty of the questions relative to the no distractor condition (ND). Additionally, the impact of including distractors in the GD condition was slightly less than the impact of including distractors in the JD condition. The results of the regression confirm this observation with the JD condition producing a large, statistically significant effect on score while the GD condition produced a smaller, non-significant effect (Table 1a). Furthermore, the results of the second regression comparing the JD and GD conditions indicates that the score difference between the two conditions was not statistically significant (Table 1b).

4.2 Impact on Duration (RQ2)

Similar to the findings regarding the impact on students' scores, when we examine the distribution of mean differences between the distractor and non-distractor pairs in Figure 4b, it is evident that both distractor conditions saw a large increase in the time students spent on the questions. However, the GD condition's distribution of differences is narrower than that of the JD condition. This may suggest that the use of visually grouped distractors can help mitigate extreme increases in duration associated with the presence of jumbled distractors. The results of the first regression indicates that both distractor conditions had a statistically significant impact on duration compared to no distractors, however the effect size of the GD condition was smaller than that of the JD condition (Table 2a). As before, while GD questions took less time than the JD questions this difference is not statistically significant when directly comparing them (Table 2b).

4.3 Impact on Item-Discrimination (RQ3)

As for item discrimination, Figure 4c indicates that both distractor conditions were associated with a small increase in item discrimination relative to the no distractor condition. Additionally, it appears



(a) Item Score Comparison **(b) Item Duration Comparison** **(c) Discrimination Comparison**

Figure 4: The item pairs are displayed with the difference in score (Figure 4a), duration (Figure 4b), and item-discrimination (Figure 4c) between the two items on the x axis. This is calculated by subtracting the given measure for the item containing distractors (i.e., JD, GD) from the measure of the item not containing distractors (ND).

that the GD condition had a slightly larger impact on item discrimination than the JD condition. The results of the t-tests indicate that in the comparison between ND (M=0.58, SD=0.098) and JD (M=0.61, SD=0.083), there was an increase in discrimination that was not statistically significant (t=1.42, p=0.16). The results for the comparison between between the ND (M=0.55, SD=0.17) and GD (M=0.61, SD=0.13) showed a similar increase however, once again, this increase was not statistically significant (t=1.24, p=0.22).

5 DISCUSSION

Prior work has indicated that one of the primary motivations of including distractors on Parsons problems is to add an additional dimension of difficulty to the questions [5]. However, the justification of expanding the difficulty of questions is typically based on the goal of increasing the signal of some ability being measured rather than simply adding additional noise. The findings of this study indicate that, relative to questions that include no distractors, both visually grouped and jumbled distractors were associated with a marginal, though statistically insignificant, increase in item discrimination. In both cases the increase was relative to an item discrimination score that already well exceeded the commonly suggested threshold of 0.4 [8].

Despite not increasing item quality in a significant way, instructors may wish to include distractors in their Parsons problems for diagnostic purposes even if they do not offer practical benefits from the item theory perspective. Distractors may offer a potential opportunity to gain insight into which errors and misconceptions are most prevelant [30]. Doing so may help instructors refine their instructional materials and approaches to help correct these misconceptions and errors in the future.

However, the potential diagnostic benefits of including distractors must be weighed against their drawback, namely their impact on the amount of time students are spending on questions. The utility of including distractors is likely reduced if the marginal increase in discrimination comes at the cost of a significant increase in the amount of time. The findings of this study indicates that the inclusion of distractors lead to a significant increase in the amount of time students spent on the questions. In examining the impact of visually grouped and jumbled distractors, we found that visually grouped distractors were associated with a smaller increase in the amount of time (and had a smaller impact on score), though the difference was not statistically significant. We suspect that this may be a byproduct of the relatively small number of questions included in the study and the large amount of variance in the data. Despite this caveat, the majority of the results points to the use of visually grouped distractors as an effective way to mediate the negative impacts associated with distractors in general, even if the effect is not as large as might have initially been expected. This provides some further justification for the argument that visually grouped distractors are comparatively better than jumbled distractors.

6 LIMITATIONS

The primary limitations of our work are that our investigations are restricted to a single class and uses what is likely only a small subset of the types of distractors that could be created. We primarily use distractors based on common errors made by students while writing code. Other distractors, such as those that contain logic errors or suggest a different but incomplete solution path, could be used as well. As such, future work should further consider the impact of different types of distractors on item quality.

Our set of questions is modest in size, which may be the primary driver of the lack of statistically significant results in some comparisons. Replicating the experiment with larger question sets will be valuable to further validate the best use cases for different forms of Parsons problems.

7 CONCLUSION

Considering our results, we are inclined to recommend that distractors be avoided on Parsons problems deployed on exams due to the substantial increase in duration seen in both the JD and GD conditions. *However*, the marginal increase in discrimination and opportunity for diagnosing student difficulties may make distractors desirable for formative contexts, where increasing time-on-task and exposing students to a variety of common mistakes could help them avoid these mistakes in the future. If instructors wish to include distractors on their exams, particularly to assess skills related to avoiding distractors, visually grouped distractors appear to be a better choice than jumbled distractors.

REFERENCES

[1] Mary J Allen and Wendy M Yen. 2001. *Introduction to measurement theory*. Waveland Press.

[2] Robert L Brennan. 1972. A generalized upper-lower item discrimination index. *Educational and Psychological Measurement* 32, 2 (1972), 289–303.

[3] Derek C Briggs, Alicia C Alonzo, Cheryl Schwab, and Mark Wilson. 2006. Diagnostic assessment with ordered multiple-choice items. *Educational Assessment* 11, 1 (2006), 33–63.

[4] Lee J Cronbach. 1951. Coefficient alpha and the internal structure of tests. *psychometrika* 16, 3 (1951), 297–334.

[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*. 195–202.

[7] Robert L Ebel. 1967. The relation of item discrimination to test reliability. *Journal of educational measurement* 4, 3 (1967), 125–128.

[8] Robert L Ebel and David A Frisbie. 1972. *Essentials of educational measurement*. Prentice-Hall Englewood Cliffs, NJ.

[9] Max D Engelhart. 1965. A COMPARISON OF SEVERAL ITEM DISCRIMINATION INDICES 1. *Journal of Educational Measurement* 2, 1 (1965), 69–76.

[10] Barbara Ericson, Austin McCall, and Kathryn Cunningham. 2019. Investigating the affect and effect of adaptive parsons problems. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*. 1–10.

[11] 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.

[12] Barbara J Ericson, Paul Denny, James Prather, Rodrigo Duran, Arto Hellas, Juho Leinonen, Craig S Miller, Briana B Morrison, Janice L Pearce, and Susan H Rodger. 2022. Parsons problems and beyond: Systematic literature review and empirical study designs. *Proceedings of the 2022 Working Group Reports on Innovation and Technology in Computer Science Education* (2022), 191–234.

[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] Barbara J Ericson and Bradley N Miller. 2020. Runestone: A platform for free, online, and interactive ebooks. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 1012–1018.

[16] Barbara J Ericson, Janice L Pearce, Susan H Rodger, Andrew Csizmadia, Rita Garcia, Francisco J Gutierrez, Konstantinos Liaskos, Aadarsh Padiyath, Michael James Scott, David H Smith, et al. 2023. Conducting multi-institutional studies of Parsons problems. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2*. 571–572.

[17] Max Fowler, David H Smith IV, Mohammed Hassan, Seth Poulsen, Matthew West, and Craig Zilles. 2022. Reevaluating the relationship between explaining, tracing, and writing skills in CS1 in a replication study. *Computer Science Education* 32, 3 (2022), 355–383.

[18] Flynn Fromont, Hiruna Jayamanne, and Paul Denny. 2023. Exploring the Difficulty of Faded Parsons Problems for Programming Education. In *Proceedings of the 25th Australasian Computing Education Conference*. 113–122.

[19] Mark J Gierl, Okan Bulut, Qi Guo, and Xinxin Zhang. 2017. Developing, analyzing, and using distractors for multiple-choice tests in education: A comprehensive review. *Review of Educational Research* 87, 6 (2017), 1082–1116.

[20] Louis Guttman and Izchak M Schlesinger. 1967. Systematic construction of distractors for ability and achievement test items. *Educational and Psychological Measurement* 27, 3 (1967), 569–580.

[21] Kyle James Harms, Jason Chen, and Caitlin L Kelleher. 2016. Distractors in Parsons problems decrease learning efficiency for young novice programmers. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*. 241–250.

[22] Carl C Haynes and Barbara J Ericson. 2021. Problem-solving efficiency and cognitive load for adaptive parsons problems vs. writing the equivalent code. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.

[23] Angelica Hotiu. 2006. *The relationship between item difficulty and discrimination indices in multiple-choice tests in a physical science course*. Ph. D. Dissertation. Citeseer.

[24] Ville Karavirta, Juha Helminen, and Petri Ihantola. 2012. A mobile learning application for parsons problems with automatic feedback. In *Proceedings of the 12th koli calling international conference on computing education research*. 11–18.

[25] Raymond Lister, Tony Clear, Dennis J Bouvier, Paul Carter, Anna Eckerdal, Jana Jacková, Mike Lopez, Robert McCartney, Phil Robbins, Otto Seppälä, et al. 2010. Naturally occurring data as research instrument: analyzing examination responses to study the novice programmer. *ACM SIGCSE Bulletin* 41, 4 (2010), 156–173.

[26] Frederic M Lord. 1953. An application of confidence intervals and of maximum likelihood to the estimation of an examinee's ability. *Psychometrika* 18, 1 (1953), 57–76.

[27] Frederic M Lord and Melvin R Novick. 2008. *Statistical theories of mental test scores*. IAP. 330–332 pages.

[28] Wajiha Mahjabeen, Saeed Alam, Usman Hassan, Tahira Zafar, Rubab Butt, Sadaf Konain, and Myedah Rizvi. 2017. Difficulty index, discrimination index and distractor efficiency in multiple choice questions. *Annals of PIMS-Shaheed Zulfiqar Ali Bhutto Medical University* 13, 4 (2017), 310–315.

[29] Bradley N. Miller and David L. Ranum. 2012. Beyond PDF and ePub: Toward an Interactive Textbook. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education* (Haifa, Israel) *(ITiCSE '12)*. ACM, New York, NY, USA, 150–155. https://doi.org/10.1145/2325296.2325335

[30] 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*. 157–163.

[31] Seth Poulsen, Shubhang Kulkarni, Geoffrey Herman, and Matthew West. 2023. Efficient Feedback and Partial Credit Grading of Proof Blocks Problems. In *Artificial Intelligence in Education: 24th International Conference, AIED 2023, Tokyo, Japan, July 3–7, 2023, Proceedings, Part I*. Springer.

[32] Adi Setiawan. 2014. Simulation study of item validity testing and item discrimination index. In *1st International Conference on Electrical Engineering, Computer Science and Informatics 2014*. Institute of Advanced Engineering and Science.

[33] David H Smith IV and Craig Zilles. 2023. Discovering, Autogenerating, and Evaluating Distractors for Python Parsons Problems in CS1. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 924–930.

[34] Tom AB Snijders and Roel J Bosker. 2011. *Multilevel analysis: An introduction to basic and advanced multilevel modeling*. Sage.

[35] Rashmi Vyas and Avinash Supe. 2008. Multiple choice questions: a literature review on the optimal number of options. *Natl Med J India* 21, 3 (2008), 130–3.

[36] Nathaniel Weinman, Armando Fox, and Marti Hearst. 2020. Exploring challenging variations of parsons problems. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 1349–1349.

[37] 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.

[38] Zihan Wu. 2023. Investigating the Effectiveness of Variations of Micro Parsons Problems. In *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 2*. 120–122.

[39] Zihan Wu, Barbara J Ericson, and Christopher Brooks. 2023. Using Micro Parsons Problems to Scaffold the Learning of Regular Expressions. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. 457–463.

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