Integrating Personalized Learning into Online Education through Content Aggregation, Data Mining, and Reinforcement Learning

by
Ethan Prihar

A Dissertation
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in
Data Science

_____

May 2023

APPROVED:

_____
Prof. Neil Heffernan
Major Adviser, Computer Science Department, WPI

_____
Prof. Adam Sales
Dissertation Committee, Mathematical Sciences Department, WPI

_____
Prof. Jacob Whitehill
Dissertation Committee, Computer Science Department, WPI

_____
Prof. Anna Rafferty
Dissertation Committee, Computer Science Department, Carleton College

Table of Contents

Abstract

Personalized learning stems from the idea that students benefit from instructional material tailored to their needs. While the concept of giving each student the content that helps them learn the most is straightforward, implementing this at scale requires overcoming a gauntlet of challenges. One must aggregate a breadth of content such that enough variety exists to support each students' specific preferences, calculate quantifiable aspects of students' behaviors and traits that correlate with which content is most effective for them, design metrics that accurately measure learning, and create algorithms that can learn the relationships between students' features and the effects of different content on their learning across thousands of students in real time. This dissertation discusses different approaches for collecting, interpreting, and recommending instructional content to students with a focus on learning interpretable insight that can inform educational pedagogy outside of online learning as well as within it. Ultimately, we designed a content recommendation algorithm that performed equivalently or better than similar existing algorithms while also allowing for unbiased statistical analysis of the data.

Introduction

In my time at WPI I have worked to integrate personalized learning into ASSISTments via the delivery of on demand student supports. I have approached this task from three different angles. Firstly, I have worked to create and collect features of the students and content in ASSISTments that are predictive of what the most effective support is likely to be for each student. Secondly, I have worked to aggregate student supports for the ASSISTments platform, and thirdly, I have integrated existing and novel recommendation algorithms into ASSISTments that use reinforcement learning, specifically multi-armed bandit algorithms, to effectively personalize the support provided to students. In each section below I will briefly summarize the works I have contributed to, my contributions, whether it has been published, and if so, where and when.

Chapter 1
Feature Creation, Collection, and Analysis

The following research papers all revolve around investigating methods of creating machine learned features, or investigating existing features for opportunities to use them for personalizing the support ASSISTments gives to students.

The first paper, "**Identifying Struggling Students by Comparing Online Tutor Clickstreams**", is an anomaly detection algorithm which was accepted as a short paper at AIED 2021. In this extended version of the AIED short paper, which was also my masters thesis, I designed and evaluated a novel algorithm to detect when students were behaving unusually. This algorithm can be used as a feature in a recommendation model because anomalous students might need different support than students with typical behavior.

The second paper, "**The Effects of Socioeconomic Status and Teachers' Pedagogy on Student Engagement During Remote Learning**", is an analysis of the effects that COVID-19 had on the 2020-2021 school year. This paper evaluated features of students and teachers engagement during COVID, which are predictive of their performance within ASSISTments, and therefore might be useful in a recommendation algorithm. In this work I aggregated all the data and performed the analysis. The extended version of a poster accepted at LAK 2021 is included.

The third paper, "**The Effect of an Intelligent Tutor on Performance on Specific Posttest Problems**", investigates whether or not teachers' use of assignments in ASSISTments related to specific skills had a significant impact on students' performance on state test score problems related to those skills. This paper helps determine the value of information related to students prior practice with similar content. If these features are useful, they can be used to better model student performance and predict what content is most likely to benefit them. I supported this work by providing data from ASSISTments on students' use of skill-related assignments. This full paper was published at EDM 2021.

The fourth paper, "**Classifying Math Knowledge Components via Task-Adaptive Pre-Trained BERT**", uses a fine-tuned BERT model to out-perform existing models for predicting what mathematics skills are required to solve a mathematics problem. In this work I did not create the model, but I collected and provided data from ASSISTments to Penn State to use during the fine-tuning and training process. ASSISTments can use this model to identify the relevant skills in problems and support messages and use these skills as features in a recommendation algorithm. This full paper was published at AIED 2021.

The fifth paper, "**Deep Learning or Deep Ignorance? Comparing Untrained Recurrent Models in Educational Contexts**", is an evaluation of affect detection algorithms using deep recurrent untrained networks such as echo state networks. The internal state of the networks were shown to predict students' emotional state during their assignments based on their clickstream data. These models can be used to extract features from clickstream data that can be used by a recommendation algorithm to personalize students' learning. In this work I

implemented and tested all the deep recurrent neural networks. This full paper was published at AIED 2022.

The sixth paper, "**Using Auxiliary Data to Boost Precision in the Analysis of A/B Tests on an Online Educational Platform: New Data and New Results**", uses a novel method to analyze the results of online educational experiments. This method requires the development of a machine learning model to predict students' performance prior to their participation in the experiment. In this work, I developed and trained a neural network that combined statistical and temporal features of students. This work required that I develop an expansive set of features for students and allowed me to evaluate which features were impactful in predicting students' performance. This provided insight into which features of students would be most descriptive of their habits when attempting to personalize their learning. This paper was submitted to JEDM, and is an expansion of a poster that was published at AIED 2022.

The seventh paper "**Effective Evaluation of Online Learning Interventions with Surrogate Measures**" investigated which features of students were the best surrogate measure of their posttest score after an assignment. In this work I created features and used these features to train models in an attempt to create an effective surrogate measure of learning. In this work, none of the features or models out-performed a simple next-problem correctness measure of learning, which helped elucidate the lack of commonality in the relationship between students' behavioral trends and their performance across different assignments. This paper has been submitted to EDM 2023.

Chapter 1.1

Identifying Struggling Students by Comparing Online Tutor Clickstreams

Identifying Struggling Students by Comparing Online Tutor Clickstreams*

Ethan Prihar[1], Alexander Moore[1], and Neil Heffernan[1]

Worcester Polytechnic Institute

Abstract. New ways to identify students in need of assistance are imperative to the evolution of online tutoring platforms. Currently implemented models to identify struggling students use costly and tedious classroom observation paired with student's platform usage, and are often suitable for only a subset of students. With the recent influx of new students to online tutoring platforms due to COVID-19, a simple method to quickly identify struggling students could help facilitate effective remote learning. To this end, we created an anomaly detection algorithm that models the normal behavior of students during remote learning and recognizes when students deviate from this behavior. We demonstrated how anomalous behavior not only revealed which students needed additional assistance, but also helped predict student learning outcomes and reduced the confidence intervals in research experiments performed within the online tutoring platform.

Keywords: Online Learning · Tutoring · Unsupervised Learning · Anomaly Detection · Outlier Detection

1 Introduction

Finding patterns in student behavior that correlate negatively with learning is often costly, requiring professional observers to watch students as they complete assignments [22, 3, 12, 15]. Algorithms created to identify these behaviors can be biased toward correctly identifying patterns in select populations [6] and can provide too specific or too great a quantity of information to be practically deployed by an instructor to help their students [12]. Furthermore, a model that requires expensive labeled data is unlikely to be updated often, which introduces model bias as populations and use cases change over time.

* I would like to thank Neil Heffernan, Lane Harrison, Alex Moore, and multiple NSF grants (e.g., 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, 1535428, 1440753, 1316736, 1252297, 1109483, DRL-1031398), as well as the US Department of Education for three different funding lines; a) the Institute for Education Sciences (e.g., IES R305A170137, R305A170243, R305A180401, R305A120125, R305A180401, R305C100024), b) the Graduate Assistance in Areas of National Need program (e.g., P200A180088 P200A150306), and c) the EIR. We also thank the Office of Naval Research (N00014-18-1-2768), Schmidt Futures, and anonymous philanthropy.

These common problems have been exacerbated by recent events. COVID-19 has lead to an unprecedented demand for remote learning [27] and within the online learning platform ASSISTments [11, 20] the number of users has grown tenfold since schools have switched to teaching remotely. Many students and teachers who have made the transition to remote learning have not previously used an online tutoring platform. This can cause inequity in students' quality of learning due to a lack of available resources and access to technology in lower income districts, exacerbating the achievement gap [17, 16, 9].

Unsupervised anomaly detection algorithms are a quickly trainable and deployable method to support instructors during this transition. Anomaly detection can identify unusual student clickstream patterns without needing a labelled dataset. This mitigates the time, expense, and subjectivity associated with manual classroom observation. Once trained, the model can be used to alert instructors when students are behaving abnormally and allow the instructor to assist the students as they see fit.

We define our objectives as follows:

1. Train a model capable of predicting student behavior using only students' clickstream data.
2. Use the student behavior model to identify abnormally behaving students.
3. Investigate the extent to which our measure of anomalous behavior correlates with learning outcomes and engagement.
4. Determine if our anomaly detection algorithm can improve researcher's confidence in experiments performed in ASSISTments.

2 Background

2.1 ASSISTments

ASSISTments is an online learning platform that enables teachers to assign content from their curriculum and assesses student progress in the classroom or remotely [11]. Within ASSISTments, as students complete assigned work, the clickstream data of each student is recorded, aggregated into statistics, and then provided to teachers in reports. These reports inform teachers of the common wrong answers and low performing students in their class. ASSISTments also supports randomized controlled experimentation using its content libraries, allowing independent researchers to test experimental pedagogies. Researchers can create assignments in which students are randomly assigned to different experimental conditions. Each condition contains either no additional tutoring (control) or a new tutoring strategy (treatment). As students complete the experimental assignment, ASSISTments collects data on their performance, which is used to evaluate the effectiveness of the new tutoring strategy [11]. For our anomaly detection algorithm, we used the raw clickstream data collected from students using the ASSISTments tutor to model student behavior, and the data collected during two experiments performed within the platform to determine whether we could increase experimental confidence.

2.2 Related Work

Evaluating Students' Latent Qualities For more than 40 years, knowledge tracing has used data on students' problem responses to estimate subject mastery, which can be used to identify students in need of instructor intervention [8]. Knowledge tracing and its variants stem from mastery learning, an assumption that students can achieve expertise if the domain knowledge is shaped into a hierarchy of component skills, and learning experiences are structured such that prerequisite skills to mastery are taught before subsequent ones [8, 23]. The knowledge tracing process estimates the probability that the student has learned each of the requisite skills necessary to master a task as the student solves exercises. While knowledge tracing can be used to identify struggling students, it does so only by estimating students' mastery of skills. Our anomaly detection algorithm has the potential to recognize struggling students by recognizing atypical behavior, which can include behavior indicative of a lack of skill mastery among other behaviors counterproductive to learning.

Students' clickstream data has also been used to predict their emotional state. Affect detection identifies the emotional state of students and relates that state to their learning gains. Past work has shown that emotions like boredom correlate negatively with learning, while emotions like frustration correlate positively with learning [22, 15]. Initially, affect models were created by observing students' emotional state in class and correlating it with their test scores. Since then, student clickstream data correlated with classroom observation have been used to train affect models, but this method has fallen short at generalizing to different types of students. For example, affect models were less accurate for students from rural areas when the model was trained on data gathered from urban and suburban areas [6]. These models require labeled datasets that are difficult to update without further human observation of students. Generalization of our algorithm to new groups of students comes naturally as new students use the platform, which facilitates custom models for specific groups of students if necessary.

Predicting Students' Behavior In previous studies related to online student behavior, experts created features indicative of cheating based on students' behavior within a massive open online course and trained a classification model to identify labeled cases of cheating [1]. Furthermore, the similar nature of cheating behaviors was used to generalize this model to recognize when other types of cheating occurred [2]. Although this process identified cheating students, it required the creation of informative features and relied upon manually labeled cheating examples. If another type of cheating arises, in which students behave differently than in the initial type of cheating, this method would require new labeled data and potentially new features which would pose a significant ongoing overhead cost. The unlabelled data used to train our anomaly detection algorithm is readily generated as students interact with the online learning platform. No human observation is necessary. If circumstances change, the algorithm can be quickly retrained and implemented.

Another student behavior that has been of interest to the learning science community is gaming. Gaming is an attempt by the student to exploit properties of the tutoring platform to progress, rather than learn the material [4]. In past research, gaming behaviors were identified by experts, and were either algorithmically or manually derived into indicative features [14, 18, 28, 19, 5, 21]. These features were used to create models that could identify students within the tutoring platform who were trying to game the system. These methods relied on experts to confirm which patterns were indicative of gaming, and as new gaming patterns arose, these algorithms fell short. Our anomaly detection algorithm can perform ongoing learning of current and emerging undesired student behavior without the need for expert analysis.

3 Methodology

In order to identify anomalous students, we first trained a model to predict typical student behavior and then used the error in the model's predictions to identify students behaving anomalously. In the following sections we provide details on the data available for model training and evaluation, the structure of the models, and the model's training and validation process.

3.1 Data Processing

Within ASSISTments every action a student takes is recorded. The action records consist of action-timestamp pairs grouped by student and assignment. Working with this clickstream data is an extremely low-level interpretation of students' interactions with ASSISTments; it does not contain additional information such as features of the student, classroom, learning material, or past performance. The types of student actions contained in this data are described in Table 1.

Table 1: Student Actions Recorded in ASSISTments

Student Action	Description
Assignment Started	Student began an assignment
Assignment Resumed	Student returned to an incomplete assignment
Assignment Finished	Student completed an assignment
Problem Started	Student began a problem
Problem Finished	Student completed all parts of a problem
Tutoring Requested	Student viewed tutoring material
Correct Response	Student submitted a correct answer
Wrong Response	Student submitted a wrong answer
Open Response	Student submitted an open response question
Answer Requested	Student was shown the correct answer
Continue Selected	Student moved on to the next problem

Only actions from Skill Builder assignments were used to train the model. Skill Builders are assignments in ASSISTments in which students answer a sequence of problems addressing a single math skill until they answer three problems in a row correctly. Skill Builders were used for training because they have a consistent format and are unlikely to cause divergences in typical student behavior. The distribution of the number of actions taken in Skill Builders is a highly-skewed exponential distribution: almost all students took less than 50 actions to complete each of their assignments, but outlying observations show some students taking 100 to 400 actions.

3.2 The Behavior Prediction Model

For our anomaly detection algorithm to be successful, the behavior prediction model had to be complex enough to capture trends in student behavior, but not so complex that it became capable of predicting the behavior of abnormally behaving students as well. To find a suitable model, we trained a logistic regression [13], neural network [26], decision tree [25], and Bernoulli naïve Bayes classifier [29] to predict a student's next action, given only their previous action and the time since taking an action.

To prepare the clickstream data for model training, we formatted the data into previous-action next-action pairs. To prepare the time data for model training, the time since taking an action was binned into 10 discrete ranges of increasing length. The ranges of the time bins grow to parallel the distribution of time between actions. The models therefore had 21 binary inputs (11 one-hot encoded actions and 10 time bins) and 11 binary outputs (11 one-hot encoded next actions).

To evaluate model quality, 985,000 actions from 7,300 students were used in 5-fold cross validation. The average accuracy, ROC AUC [10], and Cohen's Kappa [7] for each model was calculated and used to select the model used to identify anomalous students in the following evaluation.

3.3 Identification of Anomalous Students

The best model from the previous section, which was a logistic regression, was trained on all the data used in the 5-fold cross validation and was then used to predict the next action of 985,000 actions from 7,300 different students the model had never seen data from before. The average absolute error of the model's predictions across each student's actions became their "anomaly score". To determine if anomaly scores correlated with student performance, we calculated Spearman correlations [24] between the students' anomaly scores and their average correctness and time on task for all the problems the students completed in ASSISTments, excluding the assignments used to calculate their anomaly scores.

In addition to measuring the anomaly score's correlation with performance metrics, we investigated differences between students in the 95th percentile of anomaly scores, which we labeled "anomalous students", and the rest of the students, which we labeled "normal students". We investigated differences in the

frequency of actions taken and the time spent waiting before and after taking actions.

3.4 Improvements to Experimental Confidence

Lastly, it was investigated whether anomaly score could narrow the confidence interval in experimental results. To measure this, the data from two experiments performed in ASSISTments were re-evaluated. Both experiments are randomized controlled trials that each provided an additional piece of instruction during an assignment to students in the treatment group. The first experiment measured if the intervention reduced the number of problems required for students to master the material. The second experiment measured if the intervention reduced the time it took students to master the material.

For both experiments, We recomputed the 95% confidence interval of each experimental condition using a weighted standard deviation, where each student's weight was inversely proportional to their anomaly score; calculated across all their work aside from the work they did during the experiment. If weighting anomalous students less than their peers reduced the confidence intervals, that would support the claim that anomalous students have outlier behavior in experimental settings.

4 Results

4.1 Behavior Prediction Model Evaluation

The four models trained to predict students' next actions all performed relatively well. Each of the models scored highest in at least one of the three metrics calculated, and logistic regression scored highest in two of the metrics. For this reason, logistic regression was the model of choice to evaluate the relationship between anomaly score and student behavior, discussed in the following section. Table 2 shows the cross-validated performance metrics for all models.

Table 2: Performance Metrics for the Proposed Behavior Prediction Models

Model	Accuracy	ROC AUC	Cohen's Kappa
Logistic Regression	**0.71**	**0.96**	0.67
Neural Network	0.70	0.95	**0.68**
Naïve Bayes Classifier	**0.71**	0.94	0.66
Decision Tree	0.65	**0.96**	0.66

4.2 The Behavior of Anomalous Students

The students' anomaly scores, as defined in Section 3.2, correlated significantly with average correctness and time on task. The Spearman correlation coefficient

[24] and p-value of the correlations are shown in Table 3. Students with higher anomaly scores took only slightly less time than students with lower anomaly scores, but got significantly more problems wrong. These results could indicate that students with high anomaly scores have more difficulty learning the material, or exhibit more gaming behavior [5]. This is an encouraging implication as it indicates that anomaly score could be used to inform teachers of struggling students in their classes.

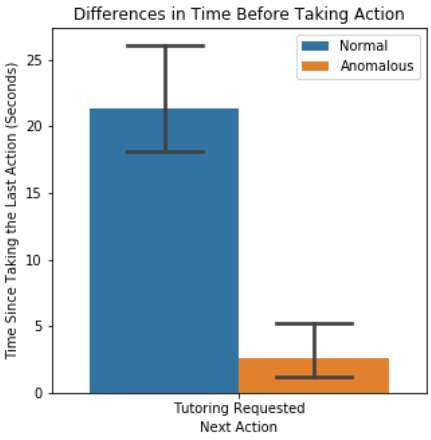
Table 3: Correlation Between Anomaly Score and Student Performance Metrics

Metric	Spearman's Rho	p-Value
Average Correctness	-0.21	<.001
Average Time-on-Task	-0.04	<.001

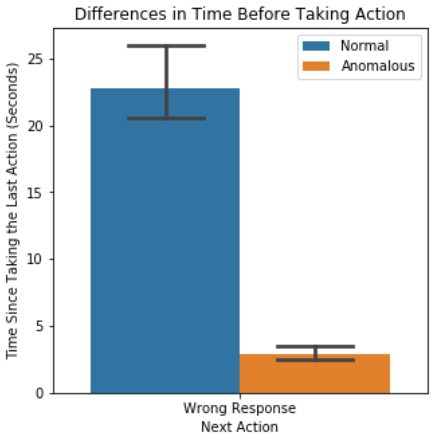
Additionally, when investigating the differences between normal and anomalous students, as defined in Section 3.2, wrong answers occurred 60% more frequently and correct responses occurred 32% less frequently in anomalous students' action sequences. The time a student waited before and after they submitted a wrong answer or received tutoring was also significantly different between normal and anomalous students. Figure 1 shows the average and 95% confidence intervals for the time before and after taking these actions. Figures 1a and 1b show that anomalous students spent about 20 seconds less looking at the problem before requesting tutoring or submitting a wrong answer. Figure 1c shows that anomalous students spent about 30 seconds less looking at tutoring and Figure 1d shows that anomalous students spent about 50 seconds less thinking about their wrong response before performing another action. These statistics paint the picture of a student that rushes to answer a problem, frequently submits wrong responses, and quickly requests tutoring. Then, without spending the time to process the new information, submits more wrong answers until they are eventually able to move on. This behavior is essentially the definition of gaming [5], and would certainly be of interest to teachers as it is counterproductive to learning and should be corrected. Students' anomaly scores could therefore be a useful tool for identifying students in need of instructional intervention without having to define, or even be aware of, the specific kinds of negative behaviors of the students.

4.3 The Effects of Anomalous Students on Experimental Confidence

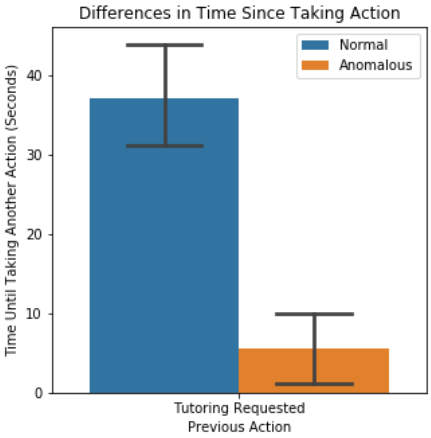
The unweighted and weighted confidence intervals for each experimental condition are shown in Table 4. In three of the four conditions, the size of the confidence interval decreased. If weighting each student inversely proportional to their anomaly score reduced the confidence intervals of the experimental conditions, this implies that anomalous students were often the outliers in these experiments. Using a weighted confidence interval could help reduce noise in experimental outcomes when the clickstream data of students are available.



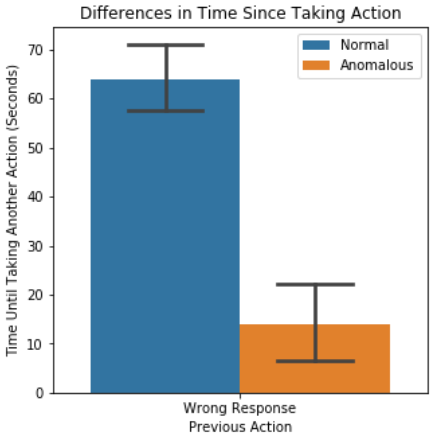
(a) Time Spent Before Requesting Tutoring



(b) Time Spent Before Submitting a Wrong Response



(c) Time Spent After Requesting Tutoring



(d) Time Spent After Submitting a Wrong Response

Fig. 1: The Average Time Spent by Normal and Anomalous Students Before and After Requesting Tutoring and Submitting Wrong Responses with 95% Confidence Bars

Table 4: Unweighted and Weighted 95% Confidence Interval for Each Experimental Condition

Condition	Regular CI	Weighted CI
Experiment 1 Control	**0.88** Problems	0.98 Problems
Experiment 1 Treatment	0.71 Problems	**0.70** Problems
Experiment 2 Control	122 Minutes	**117** Minutes
Experiment 2 Treatment	98 Minutes	**93** Minutes

5 Limitations and Future Work

While using students' clickstream data to identify anomalous students has the potential to improve educational practices, there are no guarantees that this algorithm will identify students with the same unproductive behaviors that we have found in ASSISTments clickstream data. By creating an unsupervised metric for student behavior we have removed the bias introduced by human labels but have also removed human values from our algorithm. This could pose an issue if a majority of students needed assistance. In such a scenario, the anomalous students would be the high achievers. Care should be taken when implementing this algorithm to manually examine the behavior of anomalous students to make sure that the algorithm's determination of anomalous behavior matches the expectation for the proposed use case. In the future, work could be done to modify this algorithm to accept an example of anomalous behavior, which it could generalize in a semi-supervised context. This could alleviate the need to manually examine the behavior of students, which while time consuming, is still preferable to creating a labelled dataset.

Using this anomaly detection algorithm to calculate a weighted confidence interval for experimental conditions also poses some limitations. The primary limitation is that there is no guarantee that the anomalous students are not important to the results. For example, a treatment condition could remediate anomalous behavior. If this is the case, giving lower weights to anomalous students could make the treatment appear ineffective when really it is particularly effective on anomalous students. Knowing what causes students to be labeled anomalous would help inform when to use this anomaly detection algorithm. Future work could develop an algorithm to explain the behavior of anomalous students.

6 Conclusion

Students' anomaly scores, calculated only by comparing their clickstreams, negatively correlated with their average correctness and time on task. Additionally, anomalous students spent significantly less time thinking about a problem before getting the answer wrong or requesting tutoring, and once they were told they got the answer wrong or shown tutoring, they spent significantly less time before attempting the problem again. Using ASSISTments data, the anomaly detection algorithm was able to identify a common mode in unusual student behavior: rushing to complete assignments without trying to learn, i.e., gaming [5]. While this algorithm has the potential to be used to inform teachers in real time if their students need assistance, the behaviors identified as anomalous must be examined before choosing how to address them, lest students receive irrelevant interventions because of an incorrect assumption of what it means to be anomalous.

References

1. Alexandron, G., Lee, S., Chen, Z., Pritchard, D.E.: Detecting cheaters in moocs using item response theory and learning analytics. In: UMAP (Extended Proceedings) (2016)
2. Alexandron, G., Ruipérez-Valiente, J.A., Pritchard, D.: Towards a general purpose anomaly detection method to identify cheaters in massive open online courses (06 2019)
3. Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z.: Off-task behavior in the cognitive tutor classroom: when students" game the system". In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 383–390 (2004)
4. d Baker, R.S., Corbett, A.T., Roll, I., Koedinger, K.R.: Developing a generalizable detector of when students game the system. User Modeling and User-Adapted Interaction **18**(3), 287–314 (2008)
5. d Baker, R.S., Mitrović, A., Mathews, M.: Detecting gaming the system in constraint-based tutors. In: International Conference on User Modeling, Adaptation, and Personalization. pp. 267–278. Springer (2010)
6. Botelho, A.F., Baker, R.S., Heffernan, N.T.: Improving sensor-free affect detection using deep learning. In: International Conference on Artificial Intelligence in Education. pp. 40–51. Springer (2017)
7. Cohen, J.: A coefficient of agreement for nominal scales. Educational and psychological measurement **20**(1), 37–46 (1960)
8. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction **4**(4), 253–278 (1994)
9. DeWitt, P.: Teachers work two hours less per day during covid-19: 8 key edweek survey findings. Education Week (2020)
10. Fawcett, T.: An introduction to roc analysis. Pattern recognition letters **27**(8), 861–874 (2006)
11. Heffernan, N.T., Heffernan, C.L.: The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. International Journal of Artificial Intelligence in Education **24**(4), 470–497 (2014)
12. Holstein, K., McLaren, B.M., Aleven, V.: Student learning benefits of a mixed-reality teacher awareness tool in ai-enhanced classrooms. In: International conference on artificial intelligence in education. pp. 154–168. Springer (2018)
13. Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X.: Applied logistic regression, vol. 398. John Wiley & Sons (2013)
14. Johns, J., Woolf, B.: A dynamic mixture model to detect student motivation and proficiency. In: Proceedings of the national conference on artificial intelligence. vol. 21, p. 163. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2006)
15. Lehman, B., Matthews, M., D'Mello, S., Person, N.: What are you feeling? investigating student affective states during expert human tutoring sessions. In: International conference on intelligent tutoring systems. pp. 50–59. Springer (2008)
16. Levinson, M., Cevik, M., Lipsitch, M.: Reopening primary schools during the pandemic (2020)
17. Middleton, K.V.: The longer-term impact of covid-19 on k–12 student learning and assessment. Educational Measurement: Issues and Practice (2020)

18. Muldner, K., Burleson, W., Van de Sande, B., VanLehn, K.: An analysis of students' gaming behaviors in an intelligent tutoring system: Predictors and impacts. User modeling and user-adapted interaction **21**(1-2), 99–135 (2011)
19. Murray, R.C., VanLehn, K.: Effects of dissuading unnecessary help requests while providing proactive help. In: AIED. pp. 887–889. Citeseer (2005)
20. Ostrow, K.S., Heffernan, N.T.: Advancing the state of online learning: Stay integrated, stay accessible, stay curious. Learning science: Theory, research, & practice pp. 201–228 (2019)
21. Paquette, L., Baker, R.S.: Comparing machine learning to knowledge engineering for student behavior modeling: a case study in gaming the system. Interactive Learning Environments **27**(5-6), 585–597 (2019)
22. Pardos, Z.A., Baker, R.S., San Pedro, M.O., Gowda, S.M., Gowda, S.M.: Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In: Proceedings of the third international conference on learning analytics and knowledge. pp. 117–124 (2013)
23. Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., Sohl-Dickstein, J.: Deep knowledge tracing. Advances in neural information processing systems **28**, 505–513 (2015)
24. Schober, P., Boer, C., Schwarte, L.A.: Correlation coefficients: appropriate use and interpretation. Anesthesia & Analgesia **126**(5), 1763–1768 (2018)
25. Steinberg, D., Colla, P.: Cart: classification and regression trees. The top ten algorithms in data mining **9**, 179 (2009)
26. Svozil, D., Kvasnicka, V., Pospichal, J.: Introduction to multi-layer feed-forward neural networks. Chemometrics and intelligent laboratory systems **39**(1), 43–62 (1997)
27. UNESCO: 290 million students out of school due to covid-19: Unesco releases first global numbers and mobilizes response. UNESCO (2020)
28. Walonoski, J.A., Heffernan, N.T.: Prevention of off-task gaming behavior in intelligent tutoring systems. In: International Conference on Intelligent Tutoring Systems. pp. 722–724. Springer (2006)
29. Zhang, H.: Exploring conditions for the optimality of naive bayes. International Journal of Pattern Recognition and Artificial Intelligence **19**(02), 183–198 (2005)

Chapter 1.2

The Effects of Socioeconomic Status and Teachers' Pedagogy on Student Engagement During Remote Learning

The Effects of Socioeconomic Status and Teachers' Pedagogy on Student Engagement During Remote Learning

ETHAN PRIHAR, ANTHONY BOTELHO, JOSEPH YUEN, MIKE CORACE, ANDREW SHANAJ, ZEKUN DAI, and NEIL HEFFERNAN, Worcester Polytechnic Institute, USA

The COVID-19 pandemic has driven a tremendous increase in the demand for transparency in online learning platforms in order to investigate the various effects on students from differing socioeconomic and demographic backgrounds. The ASSISTments learning platform has grown exponentially in users since the pandemic-induced shift to remote learning in March 2020, which has provided an unprecedented opportunity to understand the effects of remote learning on groups that had not previously used online tutoring platforms. To support the learning science community, ASSISTments has compiled a comprehensive dataset on 9,609 teachers and 286,596 students who used the ASSISTments platform during the 2019-2020 school year in periods both before and after the shift to remote learning. This paper presents this dataset in conjunction with a set of exploratory analyses that compare how high-income and low-income students' performance was effected by these events. Based on the data, teachers in low-income districts who used ASSISTments for the entire school year did not experience a decrease in average assignment completion after the transition to fully-remote learning. Investigation into factors that correlated with these teachers' success revealed that viewing the information on students' progress and performance, provided to teachers by the ASSISTments platform, reliably correlated with stable assignment completion percentages.

Additional Key Words and Phrases: COVID-19, Dataset, Socioeconomic Status, Opportunity Zone

ACM Reference Format:

Ethan Prihar, Anthony Botelho, Joseph Yuen, Mike Corace, Andrew Shanaj, Zekun Dai, and Neil Heffernan. 2022. The Effects of Socioeconomic Status and Teachers' Pedagogy on Student Engagement During Remote Learning. 1, 1 (January 2022), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The events surrounding the COVID-19 pandemic has had a significant impact on educational practices and policies. On March 13, 2020, the United States declared a state of emergency in response to rising COVID-19 cases, which resulted in the closure of schools across the United States [15]. Due to the increased demand for remote learning tools, many new technologies were rapidly developed to address the difficulties associated with fully-remote learning. However, the limited resources in low-income areas prevented many students from having access to equitable educational conditions as a result of teachers needing to restructure their classes depending on available resources and access to technology [7, 9, 10]. In order to investigate the extent of the impact that fully-remote learning has had on students in such low-income school districts, and to provide the learning science community with the data to further investigate and

Authors' address: Ethan Prihar, ebprihar@wpi.edu; Anthony Botelho, abotelho@wpi.edu; Joseph Yuen, jhyuen@wpi.edu; Mike Corace, mlcorace@wpi.edu; Andrew Shanaj, ashanaj@wpi.edu; Zekun Dai, zdai@wpi.edu; Neil Heffernan, nth@wpi.edu, Worcester Polytechnic Institute, 100 Institute Road, Worcester, Massachusetts, USA, 01609.

Manuscript submitted to ACM

develop methods to address this growing achievement gap, we have compiled an extensive dataset containing the complete records of the teachers and students who used the ASSISTments learning platform during the 2019-2020 school year. Using this dataset, the impact of socioeconomic conditions on student engagement post-closure was investigated by using assignment completion as a proxy for engagement. Additionally, this dataset was able to provide insight into factors that correlated with teachers' ability to keep students engaged post-closure, discussed later in this paper.

While it is the case that in this paper we explore several research questions pertaining to student and teacher interactions with a computer-based learning platform before and after the shift to remote learning, the larger purpose of this work is the open and public release of data to the greater scientific community [1]. With this data, we encourage researchers within, across, and beyond the learning science and learning analytics communities to utilize this dataset to further investigate questions pertaining to student learning and the impact of COVID-19 in educational contexts.

With this in mind, the current paper seeks to utilize this data to explore the following two research questions:

(1) What are the differences in student engagement observed in low-income districts before and after the shift to remote learning?
(2) What are the teacher-level factors that correlate with consistent student engagement observed before and after the shift to remote learning?

2 BACKGROUND

2.1 ASSISTments

ASSISTments is an online learning platform that enables teachers to assign content from their curriculum and assess student progress in the classroom or remotely [8]. The ASSISTments platform has historically released data for the study and benefit of learning [5, 12, 14, 17], and similarly has been developed as a tool that supports the sharing of crowdsourced content [6, 11] to assist both students and teachers. ASSISTments also supports randomized controlled experimentation using its content libraries, allowing independent researchers to test experimental pedagogies. Within the ASSISTments platform, as students complete assigned work, ASSISTments records the clickstream data of these students, aggregates statistics on assignments, and then provides reports to teachers. Assignment reports and student reports are shown in Figure 1.

Assignment reports are provided to teachers to inform them of the overall performance of their class on a specific assignment. Information like common wrong answers on problems and the problems students struggled with the most is available through the assignment report. If a teacher is interested in a finer level of detail, student reports contain information on each student's progress during the assignment. The information provided by student reports includes how many attempts were required for them to correctly answer each problem and when they requested tutoring. With the information gathered from these reports, teachers are able to shape their lesson plan to meet the needs of the students by addressing any misconceptions or topics students particularly struggled with during the assignment.

2.2 The Impact of COVID-19 on Education

The COVID-19 pandemic affected the United States K-12 education system for approximately three months during the 2019-2020 academic year and is effecting 2020-2021 academic year [10]. When the United States declared a national state of emergency on March 13, 2020 [15] schools were closed and the majority of students transitioned from in-person learning to remote learning. This switch to remote learning negatively impacted students to varying degrees. Students from low-income communities were negatively impacted more than their high-income counterparts because many

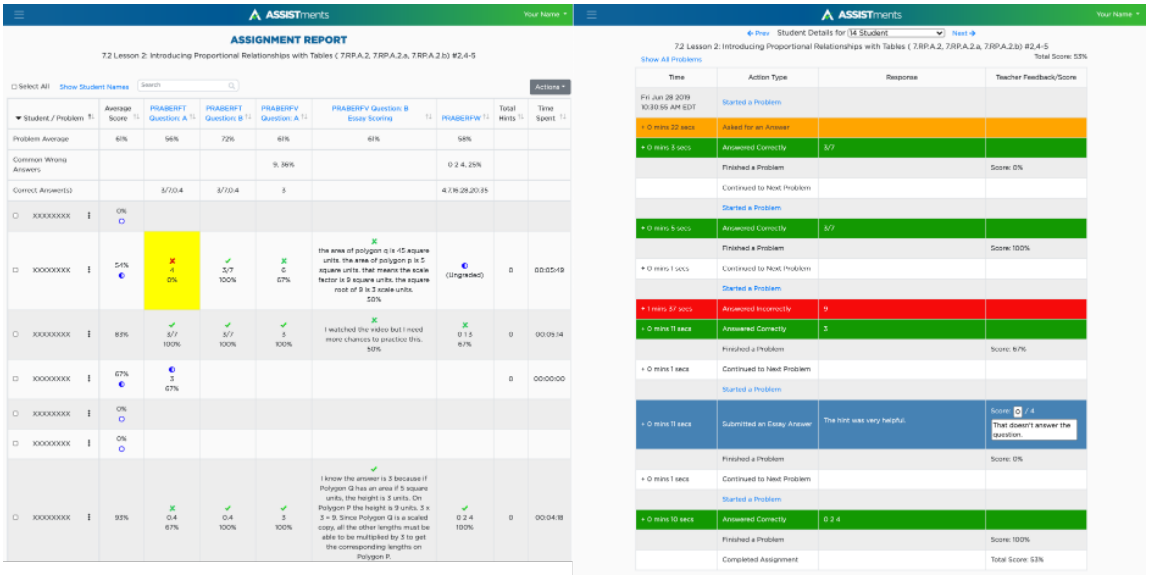


Fig. 1. Left: An Assignment Report in ASSISTments, Right: A Student Report in ASSISTments

students from low-income communities did not have access to resources, e.g., high-speed internet, quiet environments, access to computing devices, and free or reduced priced lunch [7, 9, 10].

Non-school factors pose the primary source of education inequality. Students that receive free and reduced-price lunch suffer from food insecurity when not attending school, which is correlated with low educational attainment [16]. Some districts have begun to offer meals for pickup to these students, but in large districts, a student walking to school to pick up a meal might not be a feasible option. Additionally, as winter approaches, many students do not have access to proper cold weather attire, increasing the difficulty of picking up their meals. Another impediment to low-income student's remote education is unemployment. The unemployment increase has been greatest among the young and those without higher education [4]. For many families, a student of working age bringing home an income is of high importance, and if students are not required to attend school, it may be a higher priority for them to contribute financially to their family.

According to online surveys from teachers, 76% of teachers in districts with at least 75% of students coming from low-income families taught less new material on average compared to 55% of teachers in districts with less than 25% of students from low-income families [7]. In addition, 72% of teachers in high-income districts reported that their schools provided devices for students to use for online learning, while 44% of teachers in low-income districts reported that their schools provided devices [7]. Although student engagement in schools across the country decreased, evidence supports that students in low-income communities suffered from the switch to fully-remote learning more than students in higher-income districts due to the unequal distribution of learning resources and environmental factors [10].

2.3 Opportunity Zones in the United States

In order to identify students from low-income areas, we identified which students were from school districts in opportunity zones. Opportunity zones were defined by law in 2017 as economically-distressed communities, and these

Table 1. Number of Entries in Each Table of the ASSISTments COVID-19 Dataset

Table	Number of Entries
Student Logs	95,868,119
Student Details	286,596
Problem Logs	20,753,208
Problem Details	134,655
Assignment Logs	2,505,263
Assignment Details	197,025
Class Details	17,003
Teacher Logs	1,401,702
Teacher Details	9,609
District Details	1,822

opportunity zones were created to encourage investors and businesses to move their businesses to low-income areas [3]. Even prior to the pandemic, socioeconomic status has been a significant predictor of students' test scores and overall achievement [2, 18]. Students from lower classes consistently perform worse than their peers [2], and students' family's income has been shown to correlate significantly with their standardized test scores [18]. Opportunity zones are located in these communities of lower socioeconomic status and information on the location of opportunity zones is publicly available; therefore students in low-income communities can be identified by whether or not their school district is located in an opportunity zone. These districts are likely to show evidence of the differing effects of remote learning on low-income students.

3 THE ASSISTMENTS 2019-2020 SCHOOL YEAR DATASET

This work's primary contribution to the learning science community is the ASSISTments 2019-2020 school year dataset, which is comprised of the entirety of the interactions of students and teachers within the ASSISTments platform during the 2019-2020 school year. The dataset is comprised of ten tables, each providing a different level of resolution for statistical analysis. The highest resolution tables provide clickstream data on students and teachers. In addition to these high resolution action logs, the ASSISTments dataset aggregates the student action logs into problem logs, in which each log contains the details of a student completing a problem, and assignment logs, in which each log contains the details of a student completing an entire assignment. In addition to logs of teacher and student information, statistical and demographic details are provided on the students, problems, assignments, classes, teachers, and school districts that used ASSISTments during the 2019-2020 school year. The number of entries in each table is shown in Table 1.

3.1 Dataset Description

The student logs table contains information about when and what kind of actions were taken by students in the ASSISTments tutor. For example, if a student starts a problem, answers a question incorrectly, requests a hint, and then answers correctly, the student logs table will have four entries corresponding to those four actions. These entries include the actions taken, the times the actions were taken, and which problem and assignment the student was completing when the actions were taken.

The student details table contains one entry for each student in the dataset. Each entry consists of information on the student like which class they are in and when their account was created, as well as summary statistics of the student

including how many assignments they started, how many they finished, how many problems they've completed, and their average problem correctness and time-on-task.

The problem logs table aggregates information from the student logs table, providing one entry for each problem completed by each student. The problem logs table contains information on when the student started the problem, how long the student spend on the problem, whether the student used tutoring and if so how much, the student's number of attempts, and the student's correctness on the problem.

The problem details table contains one entry for each problem in the dataset. Each entry contains information on the problems source, which Common Core State Standards skill codes apply to the problem, what kind of tutoring is available, how many students answered the problem, and the mean correctness and time-on-task of students answering the problem.

The assignment logs table aggregates information from the problem logs table, providing one entry for each assignment completed by each student. The assignment logs table contains information on when the student started the assignment, the student's average correctness on the problems in the assignment, the student's time-on-task during the assignment, and whether or not the student completed the assignment.

The assignment details table contains one entry for each assignment in the dataset. Each entry contains information on the type of assignment, which class the assignment was assigned to, when it was assigned and due, the number of students that were assigned, started, and completed the assignment, how many problems were in the assignment, and the average correctness and time-on-task of students who completed the assignment.

The class details table contains one entry for each class in the dataset. Each entry contains information on which teacher taught the class, when the class was created, how many students were in the class, and how many assignments were assigned to the class.

The teacher logs table contains information about when and what kind of actions were taken by teachers in the ASSISTments learning platform. For example, if a teacher viewed an assignment report, then opened a student report, then graded a student's open response question, the teacher logs table will have three entries corresponding to those three actions. These entries include the actions taken, the times the actions were taken, and which assignments, students, and problems the teacher was viewing reports for or grading.

The teacher details table contains one entry for each teacher in the dataset. Each entry contains information on the frequency that the teacher viewed reports and graded open response problems, as well as when the teacher created their account and which district the teacher was in.

The district details table contains one entry for each district in the dataset. Each entry contains information on where the district is located, whether or not the district is in an opportunity zone, and if available, a description of the district's location, e.g., rural or suburban.

4 THE IMPACT OF COVID-19 ON ASSISTMENTS USERS

ASSISTments saw an unprecedented increase in users during the 2019-2020 school year, most of this influx was due to teachers being required to teach remotely from March 13, 2020 until the end of the school year. Figure 2 shows the number of active teachers by district income level each month of the 2019-2020 school year. For the purposes of analysis, if the school district was located in an opportunity zone, it was considered low-income and if a school district was not located in an opportunity zone, it was considered regular-income. A teacher is considered active if they assigned at least one assignment to their students. April 2020 had the highest number active users, having over ten times the number of pre-closure active teachers. Interestingly, the number of users in low-income districts grew and fell faster

than the number of users from regular-income districts. In the following sections we explore the difference in student engagement between students from low-income and regular-income districts.

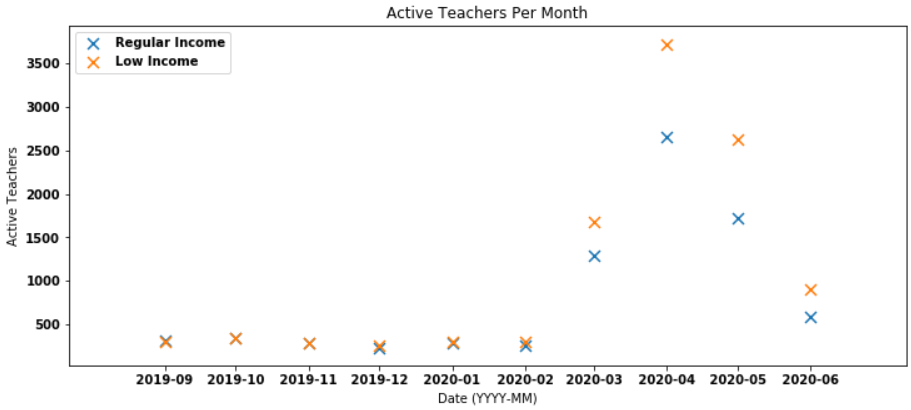


Fig. 2. Number of Active Teachers and Students for each Month in the 2019-2020 School Year

5 DIFFERENCES BETWEEN STUDENT ENGAGEMENT IN LOW-INCOME AND REGULAR-INCOME DISTRICTS

To further investigate the impact of mandatory remote learning on students, we compared students' assignment completion from before the mandatory school closure to after the closure. The corpus of student data was filtered to only include data from classes with more than ten students, where teachers assigned work at least once a month for almost every month during the school year, and where at the beginning of the class, at least one assignment was completed by at least 75% of students. The data was filtered this way to avoid data from classes where students never participated or classes where teachers were only practicing using ASSISTments. The remaining consistently active teachers made up only about 11% of the total pre-closure teachers, and about 41% of the post-closure teachers.

Figure 3 shows the gap in average assignment completion between low-income and regular-income districts using 16,486 assignments due before school closures, and 93,266 assignments due after school closures. The difference in assignment completion grew from about 4.7% to about 11.4%. This change was due to a decrease in the average assignment completion of low-income districts. Regular-income districts didn't experience a significant decrease.

To explore the decrease in low-income students' assignment completion, the change in assignment completion was calculated separately for teachers in low-income areas who were consistently active both before and after the closure, who are referred to as persistent, and teachers from low-income areas who either started or stopped using ASSISTments after the closure, who are referred to as new. The percent completion of 5,051 assignments from persistent teachers and 61,579 assignments from new teachers in low-income districts is show in Figure 4. Figure 4 reveals that the significant drop in low-income students' assignment completion is entirely due to new teachers. There was no statistically significant change in the assignment completion of low-income students in classes taught by persistent teachers.

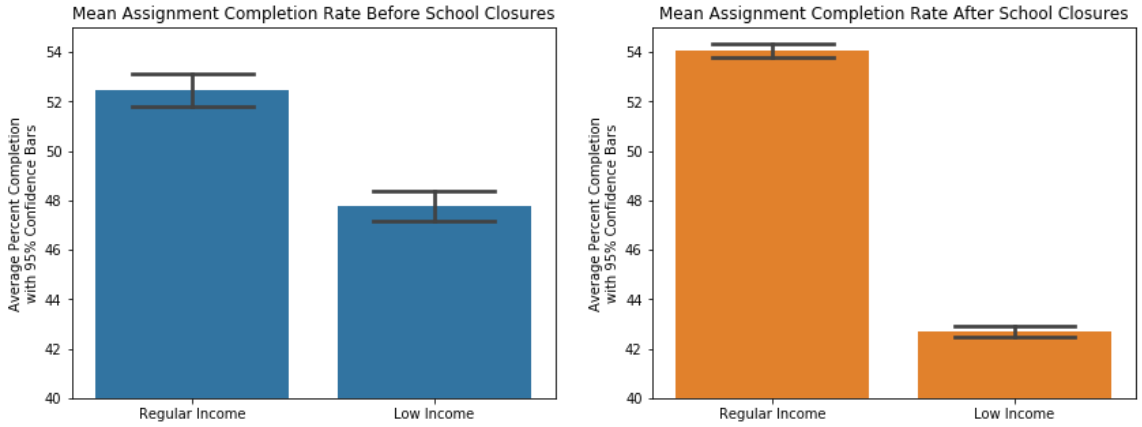


Fig. 3. Average Assignment Completion Between Low-Income and Regular-Income Districts Before and After the Closure

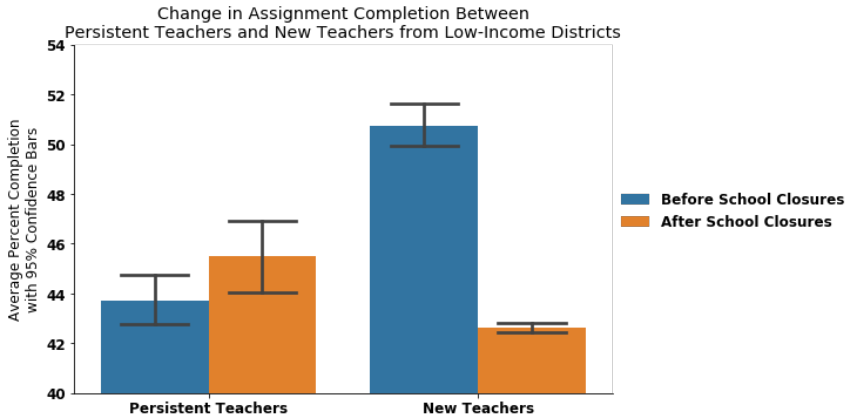


Fig. 4. The Change in Average Assignment Completion Before and After School Closures Between New and Persistent Teachers in Low-Income Districts

5.1 Differences Between New and Persistent Teachers in Low-Income Districts

To investigate whether there were any noticeable differences between how new and persistent teachers from low-income districts used ASSISTments, the teachers' assigned work loads, report viewing, and grading habits were compared. Figure 5 shows the mean values for different measurements of teachers' habits.

The two largest differences between new and persistent teachers are that persistent teachers viewed about 10% more of their assignment reports and commented on about 4% more of their students' open response questions. Viewing more reports and leaving more comments on students' work suggests that persistent teachers used ASSISTments more to engage with their students. Alternatively, these findings suggest that persistent teachers were more familiar with the ASSISTments platform, and were therefore able to access reports and student open response questions more easily.

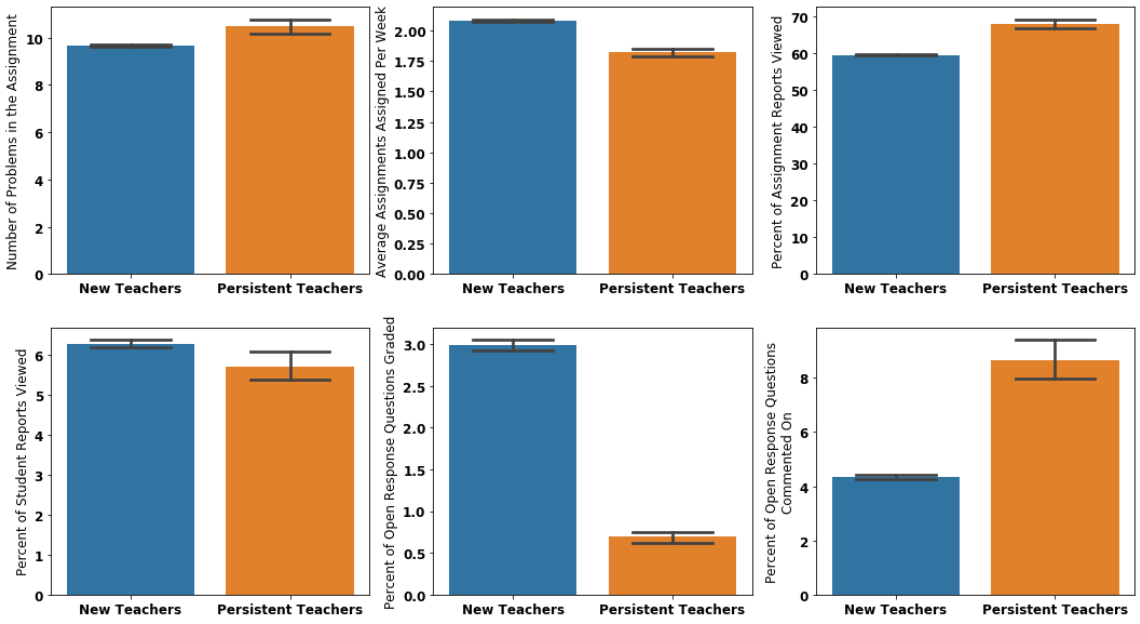


Fig. 5. The Differences Between New and Persistent Teachers in Low-Income Districts

6 FACTORS CORRELATED WITH SUCCESSFUL REMOTE LEARNING

To determine if the differences between new and persistent teachers were indicative of a larger correlation between teacher behavior and assignment completion, the correlations between average assignment completion, problems per assignment, assignments per week, percentage of assignment reports viewed, percentage of student reports viewed, percentage of open response problems graded, and percentage of open response problems commented on were calculated, and are shown below in Figure 6 for the 87,716 assignments due after school closures.

Figure 6 shows that the two teacher statistics that most significantly correlate with assignment completion is assignment reports and student report views. This correlation implies that when teachers are more engaged with the performance of their students through their review of reports, it results in the students being more engaged with learning the material. The correlation between student report views and assignment completion is more steep than the correlation between assignment report views and assignment completion. Student level reports offer finer details on the behavior of a student during the assignment and imply a teacher is more curious about the performance of each student. It logically follows that if report views are a measure of teacher engagement, and teacher engagement leads to student engagement, then student reports would show a stronger influence on assignment completion than assignment reports. The finding that assignment report views correlate with assignment completion supports the previous finding that low-income persistent teachers viewed more assignment reports than the low-income new teachers, and had on average higher assignment completion percentages. While these findings are promising, they are only observational in nature. From these results one cannot conclude that report viewing causes high completion rates, it is likely that many other factors influence both completion rates and report views. However, knowing that report views correlate with

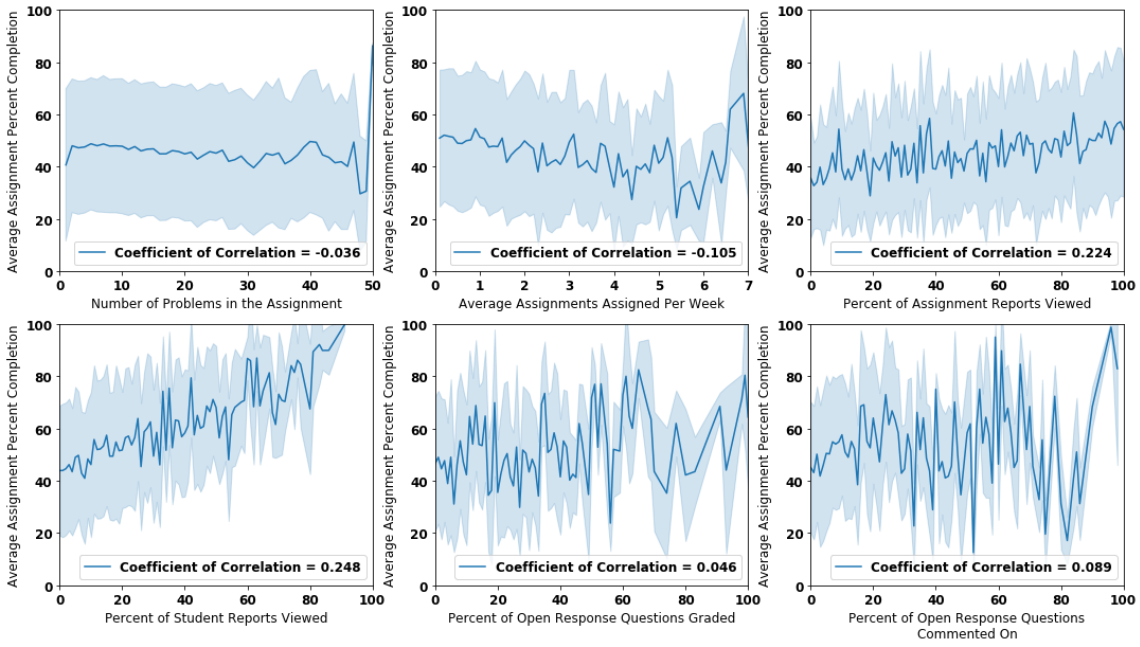


Fig. 6. The Correlation Between Teacher Behaviors and Assignment Completion

higher assignment completion can still be used to help direct programs meant to instruct teachers on how to maximise the benefits of using ASSISTments.

7 CONCLUSION

Although we can't directly identify the cause of high or low assignment completion percentages, the investigation lead to some reassuring correlational findings. Primarily, that while most teachers' student's average assignment completion fell after school closures, some teachers, mostly from low-income districts, were able to maintain their students' engagement. The teachers who maintained their students' assignment completion did so while viewing more reports and leaving more comments than teachers whose students' assignment completion fell. Given these finding, a subsequent experiment could be proposed in which some teachers are given access to ASSISTments, but without the full suite of reporting and grading features, and other teachers are given full access to ASSISTments. This would help measure the impact of the tools ASSISTments provides to teachers.

Conveniently, an experiment like this was published in 2016. In this study, 2,850 students from 43 schools showed significant improvements to their state test scores when their teachers had full access to ASSISTments compared to teachers that only had limited access [13]. However, during the study teachers met in person with students but correlations found in this paper are from entirely remote learning. More experiments should be conducted to investigate any causal claims related to the correlations found in this dataset. At the time of the previously mentioned study, the resolution and quantity of data available was much lower than what is now available.

Moving forward, the ASSISTments 2019-2020 school year dataset, and future datasets with the same format, can be used to understand the magnitude of the effect of different aspects of online instruction on student learning. The

ASSISTments 2019-2020 school year dataset has potential use beyond investigating the effects of remote learning on students. The data could be used to train more robust knowledge tracing models using the skill tags associated with problems, or to create simulations of classroom environments using the student and teacher action logs. We encourage the learning science community to explore the provided data. As we receive feedback, we can improve upon the data export process and provide the learning science community with complete and open access to ASSISTments data.

ACKNOWLEDGMENTS

We would like to thank multiple NSF grants (e.g., 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, 1535428, 1440753, 1316736, 1252297, 1109483, DRL-1031398), the US Department of Education Institute for Education Sciences (e.g., IES R305A170137, R305A170243, R305A180401, R305A120125, R305A180401, R305C100024) and the Graduate Assistance in Areas of National Need program (e.g., P200A180088 and P200A150306), and EIR the Office of Naval Research (N00014-18-1-2768 and other from ONR) and finally Schmidt Futures.

REFERENCES

[1] [n.d.]. https://osf.io/q7zc5/.
[2] Juan Battle and Michael Lewis. 2002. The increasing significance of class: The relative effects of race and socioeconomic status on academic achievement. *Journal of poverty* 6, 2 (2002), 21–35.
[3] Joseph Bennett. 2018. Lands of Opportunity: An Analysis of the Effectiveness and Impact of Opportunity Zones in the Tax Cuts and Jobs Act of 2017. *J. Legis.* 45 (2018), 253.
[4] Richard Blundell, Monica Costa Dias, Robert Joyce, and Xiaowei Xu. 2020. COVID-19 and Inequalities. *Fiscal Studies* 41, 2 (2020), 291–319.
[5] Anthony F Botelho, Ryan S Baker, and Neil T Heffernan. 2017. Improving sensor-free affect detection using deep learning. In *International Conference on Artificial Intelligence in Education*. Springer, 40–51.
[6] Anthony F Botelho and Neil T Heffernan. 2019. Crowdsourcing Feedback to Support Teachers and Students. *Sinatra, A.M., Graesser, A.C., Hu, X., Brawner, K., and Rus, V. (Eds.). (2019). Design Recommendations for Intelligent Tutoring Systems: Volume 7 - Self-Improving Systems. Orlando, FL: U.S. Army Research Laboratory* (2019), 101–108.
[7] P DeWitt. 2020. Teachers work two hours less per day during COVID-19: 8 key EdWeek survey findings. *Education Week* (2020).
[8] Neil T Heffernan and Cristina Lindquist Heffernan. 2014. The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education* 24, 4 (2014), 470–497.
[9] Meira Levinson, Muge Cevik, and Marc Lipsitch. 2020. Reopening primary schools during the pandemic.
[10] Kyndra V Middleton. 2020. The Longer-Term Impact of COVID-19 on K–12 Student Learning and Assessment. *Educational Measurement: Issues and Practice* (2020).
[11] Thanaporn Patikorn and Neil T Heffernan. 2020. Effectiveness of Crowd-Sourcing On-Demand Assistance from Teachers in Online Learning Platforms. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*. 115–124.
[12] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in neural information processing systems*. 505–513.
[13] Jeremy Roschelle, Mingyu Feng, Robert F Murphy, and Craig A Mason. 2016. Online mathematics homework increases student achievement. *AERA open* 2, 4 (2016), 2332858416673968.
[14] Douglas Selent, Thanaporn Patikorn, and Neil Heffernan. 2016. Assistments dataset from multiple randomized controlled experiments. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*. 181–184.
[15] Donald J Trump. 2020. Proclamation on Declaring a National Emergency Concerning the Novel Coronavirus Disease (COVID-19) Outbreak. *White House* (2020).
[16] Wim Van Lancker and Zachary Parolin. 2020. COVID-19, school closures, and child poverty: a social crisis in the making. *The Lancet Public Health* 5, 5 (2020), e243–e244.
[17] Xiaolu Xiong, Siyuan Zhao, Eric G Van Inwegen, and Joseph E Beck. 2016. Going deeper with deep knowledge tracing. *International Educational Data Mining Society* (2016).
[18] Rebecca Zwick. 2004. Is the SAT a "wealth test?" The link between educational achievement and socioeconomic status. *Rethinking the SAT: The future of standardized testing in university admissions* (2004), 203–216.

Chapter 1.3

The Effect of an Intelligent Tutor on Performance on Specific Posttest Problems

The Effect of an Intelligent Tutor on Performance on Specific Posttest Problems

Adam Sales
Worceseter Polytechnic Institute
asales@wpi.edu

Ethan Prihar
Worceseter Polytechnic Institute
ebprihar@gmail.com

Neil Heffernan
Worceseter Polytechnic Institute
nth@wpi.edu

John F. Pane
Rand Corporation
jpane@rand.org

ABSTRACT

This paper drills deeper into the documented effects of the Cognitive Tutor Algebra I and ASSISTments intelligent tutoring systems by estimating their effects on specific problems. We start by describing a multilevel Rasch-type model that facilitates testing for differences in the effects between problems and precise problem-specific effect estimation without the need for multiple comparisons corrections. We find that the effects of both intelligent tutors vary between problems– the effects are positive for some, negative for others, and undeterminable for the rest. Next we explore hypotheses explaining why effects might be larger for some problems than for others. In the case of ASSISTments, there is no evidence that problems that are more closely related to students' work in the tutor displayed larger treatment effects.

Keywords

Causal impact estimates,multilevel modeling,intelligent tutoring systems

1. INTRODUCTION: AVERAGE AND ITEM-SPECIFIC EFFECTS

The past decade has seen increasing evidence of the effectiveness of intelligent tutoring systems (ITS) in supporting student learning [7][13]. However, surprisingly little detail is known about these effects such as which students experience the biggest benefits, under what conditions. This paper will focus on the question of which areas of learning had the largest impact in two different year-long randomized trials: of the Cognitive Tutor Algebra I curriculum (CTA1) [17] and of the ASSISTments ITS [22].

Large-scale efficacy or effectiveness trials in education research, including evaluations of ITS [17][18][22], often estimate the effect of an educational intervention on student scores on a standardized test. These tests consist of many items, each of which tests student abilities in, potentially, a separate set of skills. Prior to estimating program effects, analysts collapse data across items into student scores, often using item response theory models [25] that measure both item- and student-level parameters. Then, these student scores are compared between students assigned to the intervention group and those assigned to control.

This approach has its advantages, in terms of simplicity and (at least after aggregating item data into test scores) model-free causal identification. If each item is a measurement of one underlying latent construct (such as "algebra ability") aggregating items into test scores yields efficiency gains. However, in the (quite plausible) case that posttest items actually measure different skills, and the impact of the ITS varies from skill to skill, item-specific impacts can be quite informative.

In the case of CTA1 and ASSISTments, we find that, indeed, the ITS affect student performance differently on different posttest items, though at this stage it is unclear why the affects differed.

The following section gives an overview of the two large-scale ITS evaluations we will discuss, including a discussion of the available data and of the two posttests. Next, Section 3 will discuss the Bayesian multilevel model we use to estimate item-specific effects, including a discussion of multiple comparisons; Section 4 will discuss the results—estimates of how the two ITS impacted different posttest items differently; Section 5 will present a preliminary exploration of some hypotheses as to why ASSISTments may have impacted different skills differently; and Section 6 will conclude.

2. THE CTA1 AND ASSISTMENTS TRIALS

This paper uses data from two large-scale field trials of ITSs CTA1 and ASSISTments. The CTA1 intervention consisted of a complete curriculum, combining the Cognitive Tutor ITS, along with a student-centered classroom curriculum. CTA1 was a created and run by Carnegie Learning; an up-

dated version of the ITS is now known as Mathia. The Cognitive Tutor is described in more detail in [2] and elsewhere, and the effectiveness trial is described in [17]. ASSISTments is a free online-homework platform, hosted by Worcester Polytechnic Institute, that combines electronic versions of textbook problems, including on-demand hints and immediate feedback, with bespoke mastery-based problem sets known as "skill builders." ASSISTments is described in [10] and the efficacy trial is described in [22].

This section describes the essential aspects of the field trials and the data that we will use in the rest of the paper.

2.1 The CTA1 Effectiveness Trial

From 2007 to 2010, the RAND Corporation conducted a randomized controlled trial to compare the effectiveness of the CTA1 curriculum to business as usual (BaU). The study tested CTA1 under authentic, natural conditions, i.e., oversight and support of CTA1's use was the same as it would have been if there was not a study being conducted. Nearly 20,000 students in 70 high schools ($n = 13,316$ students) and 76 middle schools ($n = 5,938$) located in 52 diverse school districts in seven states participated in the study. Participating students in Algebra I classrooms took an algebra I pretest and a posttest, both from the CTB/McGraw-Hill Acuity series.

Schools were blocked into pairs prior to randomization, based on a set baseline, school-level covariates, and within each pair, one school was assigned to the CTA1 arm and the other to BaU. In the treatment schools, students taking algebra I were supposed to use the CTA1 curriculum, including the Cognitive Tutor software; of course, the extent of compliance varied widely [12][11].

Results from the first and second year of the study were reported separately for middle and high schools. In the first year, the estimated treatment effect was close to zero in middle schools and slightly negative in high schools. However, the 95% confidence intervals for both these results included negative, null, and positive effects. In the second year, the estimated treatment effect was positive–roughly one fifth of a standard deviation—for both middle and high schools, but it was only statistically significant in the high school stratum.

In this study, we make use of students' overall scores on the pretest, anonymized student, teacher, school, and randomization block IDs, and an indicator variable for whether each student's school was assigned to the CTA1 or BaU, along with item-level posttest data: whether each student answered each posttest item correctly. For the purposes of this study, skipped items were considered incorrect.

2.1.1 Posttest: The Algebra Proficiency Exam

The RAND CTA1 study measured the algebra I learning over the course of the year using the McGraw-Hill Algebra Proficiency Exam (APE). This was a multiple choice standardized test with 32 items testing a mix of algebra and pre-algebra skills. Table 1, categorizes the test's items by the algebra skills they require, and gives an example of a problem that would fall into each category. The categorization was taken from the exam's technical report [6].

2.2 The Maine ASSISTments Trial

From 2012–2014, SRI International conducted an randomized field trial in the state of Maine to estimate the efficacy of ASSISTments in improving 7th grade mathematics achievement. Forty-five middle schools from across the state of Maine were randomly assigned between two conditions: 23 middle schools were assigned to a treatment condition; mathematics teachers in these schools were instructed to use ASSISTments to assign homework, receiving support and professional development while doing so. The remaining 22 schools in the BaU condition were barred from using ASSISTments during the course of the study but were offered the same resources and professional development as the treatment group after the study was over. The study was conducted in Maine due to the state's program of providing every student with a laptop, which allowed students to complete homework online.

The 45 participating schools were grouped into 21 pairs and one triplet based on school size and prior state standardized exam scores; one school in each pair, and two schools in the triplet, were assigned to the ASSISTments condition, with the remaining schools assigned to BaU. Subsequent to random assignment, one of the treatment schools dropped out of the study, but its matched pair did not. Although the study team continued to gather data from the now-unmatched control school, that data was not included in the study. However, we are currently unable to identify which of the control schools was excluded from the final data analysis, so the analysis here includes 44 schools, while [22] includes only 43.

The study measured student achievement on the standardized TerraNova math test at the end of the second year of implementation, and estimated a treatment effect of 0.18 ± 0.12 standard deviations.

In this study, we make use of anonymized student, teacher, school, and randomization block IDs, and an indicator variable for whether each student's school was assigned to the ASSISTments or BaU, along with item-level posttest data: whether each student answered each posttest item correctly. For the purposes of this study, skipped items were considered incorrect. The initial evaluation included a number of student-level baseline covariates drawn from Maine's state longitudinal data system, include prior state standardized test scores. We do not currently have access to that data; the only covariate available was an indicator of whether each student was classified as special education.

2.3 The TerraNova Test

The primary outcome of the ASSISTments Maine trial was students' scores on the TerraNova Common Core assessment mathematics test, published by Data Recognition Corporation CTB. The TerraNova assessment includes 37 items, 32 of which were multiple choice and 5 of which were open response. Unfortunately, we detected an anomaly in the item-level data for the open-response questions, so this report will focus only on the 32 multiple choice questions.

The items are supposed to align with the Common Core State Standards, but the research team was not given a document aligning CCSS with the test items. Instead, a

Objective	Items	Example
Functions and Graphs	6, 8, 19, 20, 22, 23, 27, 31, 32	Which of these points is on the graph of [function]
Geometry	12, 18, 24, 29	Find the length of the base of the right triangle shown below
Graphing Linear Equations	5, 9, 15, 17, 26	Which of the lines below is the graph of [linear equation]?
Quadratic Equations and Functions	2, 25, 28, 30	Which of these shows a correct factorization of [quadratic equation]?
Solving Linear Equations and Linear Inequalities	1, 4, 11, 13, 16	Solve the following system of equations
Variables, Expressions, Formulas	3, 7, 10, 14, 21	Which of these expressions is equivalent to the one below?

Table 1: Objectives required for the 32 items of the Algebra Proficiency Exam, the posttest for the CTA1 Evaluation

member of the ASSISTments staff with expertise in middle school education aligned them according to her best judgment. Table 2 gives this alignment. More information on specific standards can be found at the CCSS website [16].

3. METHODOLOGY: MULTILEVEL EFFECTS MODELING

In principal estimating program effects on each posttest item is straightforward: the same model used to estimate effects on student overall scores could be used to estimate effects on each item individually (perhaps—but not necessarily—adapted for a binary response). However, estimating 32 separate models for each stratum of the CTA1 study, and 32 separate models for the ASSISTments study ignores multilevel structure of the dataset, and leads to imprecise estimates. Moreover, doing so invites problems of multiple comparisons—between the four strata of the CTA1 study and the ASSISTments study, there are 160 separate effects to estimate. If each estimate is subjected to a null hypothesis test at level $\alpha = 0.05$, even if neither ITS affected test performance at all, we would still expect to find roughly eight significant effects.

Instead, we estimated item-specific effects with a multilevel logistic regression model model [8], based roughly on the classic "Rasch" model of item response theory [25][20]. That is, we estimated all item-specific effects for a particular experiment simultaneously, with one model, in which the item-specific effect estimates are random effects. The separate effects were modeled as if drawn from a normal distribution with a mean and standard deviation estimated from the data. This normal distribution can be thought of as a Bayesian prior distribution; the fact that its parameters are estimated from the data puts us in the realm of empirical Bayes [5]. This prior distribution acts as a regularizer, shrinking the several item-specific effect estimates towards their mean [15]. Although doing so incurs a small amount of bias, it reduces standard errors considerably while maintaining the nominal coverage of confidence intervals [23].

Gelman, Hill, and Masanao [9] argue that estimating a set of different treatment effects within a multilevel model also obviates the need for multiplicity corrections. Generally speaking, the reason for spurious significant results is that as a group of estimates gets larger, so does the probability that one of them will exceed the test's critical value. In other words, as a the set of estimates grows, so does their maximum (and their minimum, in magnitude). Multilevel modeling helps by shrinking the most extreme estimates towards their common mean. Since extreme values are less likely in a multilevel model, so are spuriously significant effect estimates.

A small simulation study in the Appendix (mostly) supports Gelman et al.'s argument. As the number of estimated effects grows, the familywise error rate (i.e. the probability of *any* type-I error in a group of tests) grows rapidly if effects are estimated and tested separately, but not if they are estimated simultaneously in a multilevel model. However, the error rates for the multilevel model effect estimates are slightly elevated—hovering between 0.05 and 0.075 throughout. There is good reason to believe that a fully Bayesian approach will improve these further (see, e.g., [21], p. 425).

3.1 The Model for the CTA1 Posttest

For the CTA1 RCT, we estimated a separate model for high school and middle school, but we combined outcome data across the two years. Let $Y_{ij} = 1$ if student i answered item j correctly, and let $\pi_{ij} = Pr(Y_{ij} = 1)$. Then the multilevel logistic model was:

$$\begin{aligned} logit(\pi_{ij}) &= \beta_0 + \beta_1 Year2_i + \beta_2 Trt_i + \beta_3 Pretest_i \\ &+ \beta_4 Year2_i Trt_i + \beta_5 Year2_i Pretest_i \\ &+ \gamma_{j0} + \gamma_{j1} Trt_i + \gamma_{j2} Year2_i + \gamma_{j3} Year2_i Trt_i \\ &+ \delta_i + \eta_{cls[i]} + \epsilon_{sch[i]} \end{aligned} \quad (1)$$

Where $Year2_i = 1$ if student i was in the 2nd year of the study and 0 otherwise, $Trt_i = 1$ if student i was in a school assigned to treatment, and $Pretest_i$ is i's pretest score. The coefficients β_0–β_5 are "fixed effects," that is, they are not given any probability model. γ_{j0}–γ_{j3} vary with posttest item j, and are modeled jointly as multivariate normal: $\boldsymbol{\gamma} \sim MVN(\mathbf{0}, \Sigma)$, where Σ is a 4×4 covariance matrix for the γ terms. Similarly, the random intercepts δ_i, $\eta_{cls[i]}$, and $\epsilon_{sch[i]}$, which vary at the student, classroom, and school level, are each modeled as univariate normal with mean 0 and a standard deviation estimated from the data.

Collecting like terms in model (1), note that for a student in the first year of the study, the effect of assignment to the CTA1 condition is $\beta_2 + \gamma_{j1}$ on the logit scale; in other words, the effects of assignment to CTA1 in year 1 are mod-

CCSS	Items
Expressions and Equations	17,28
Functions (8G)	26,27
Geometry	12,16,19,21,23,31
Make sense of problems and persevere in solving them (MP)	13
Ratios and Proportional Relationships	22,24,25,29
Reason abstractly and quantitatively (MP)	15,20
Statistics and Probability	10,11,32
The Number System	1,2,3,4,5,6,7,8,9,14,18

Table 2: Common Core State Standards (CCSS) for the 32 multiple choice TerraNova items, as identified by the ASSISTments team. Standards are from grade 7 except where indicated–grade 8 (8G) or Mathematical Practice (MP)

eled as normal with a mean of β_2 and a variance of Σ_{22}. The variance Σ_{22} estimates the extent to which the effect of assignment to the CTA1 condition varies from one problem to another. If the effect were the same for every posttest problem, we would have $\Sigma_{22} = 0$. For students year 2, the effect on problem j is $\beta_2 + \beta_4 + \gamma_{j1} + \gamma_{j3}$ on the logit scale—the effects are normally distributed with a mean of $\beta_2 + \beta_4$ and a variance of $\Sigma_{22} + \Sigma_{44} + 2\Sigma_{24}$. The Σ matrix also includes the covariance between the effects of the intervention on items in year 1 and the effects on the same items in year 2 as

$$Cov(\gamma_{j1}, \gamma_{j1} + \gamma_{j3}) = Var(\gamma_{j1}) + Cov(\gamma_{j1}, \gamma_{j3}) = \Sigma_{22} + \Sigma_{23}$$

Likelihood ratio tests using the χ^2 distribution can test the null hypothesis that the variance of treatment effects are 0. For simplicity, we did so using separate models for the two years, rather than the combined model (1).

The treatment effects themselves are estimated using the BLUPs (best linear unbiased predictors) for the random effects $\boldsymbol{\gamma}$. In many contexts, random effects are considered nuisance parameters, and primary interest is in the fixed (unmodeled) effects $\boldsymbol{\beta}$. However, there is a long tradition, mostly in the Bayesian and empirical Bayes literature, of using BLUPs for estimation of quantities of interest. The models were fit in `R` [19] using the `lme4` package [3], which provides empirical Bayesian estimates of the conditional (or posterior) variance of the BLUPs, which we use (in combination with the estimated standard errors for fixed effects) in constructing confidence intervals for item-specific effects.

3.2 The Model for the ASSISTments Posttest

The model for estimating item-specific effect of ASSISTments on TerraNova items was highly similar to model (1). There were three important differences: first, there was only one year of data. Second, we did not have access to pretest scores, but we did include an indicator for special education status as a covariate. Lastly, the hierarchical variance structure for student errors was somewhat different—we included an error term for teacher instead of classroom, and included random intercepts for randomization block.[1]

[1]In linear models it is typically recommended to include fixed effects for randomization block [4]. In logistic regression, including a large number of fixed effects violates the assumptions underlying the asymptotic [1]. We tried it both ways and found that it made little difference.

All in all, the model was:

$$\begin{aligned} logit(\pi_{ij}) &= \beta_0 + \beta_1 Trt_i + \beta_2 SpEd_i \\ &+ \gamma_{j0} + \gamma_{j1} Trt_i \\ &+ \delta_i + \eta_{tch[i]} + \epsilon_{sch[i]} + \zeta_{pair[i]} \end{aligned} \quad (2)$$

where $SpEd_i = 1$ if student i is classified as needing special education, $\eta_{tch[i]}$ is a random intercept for i's teacher, and $\zeta_{pair[i]}$ is a random intercept for i's school's randomization block. The rest of the parameters and variables are defined the same as in (1). The treatment effect on problem j is modeled as $\beta_1 + \gamma_{j1}$ for multiple choice items. The random effects $\boldsymbol{\gamma} \sim N(\mathbf{0}, \Sigma)$ where Σ is a 2×2 covariance matrix.

4. MAIN RESULTS: ON WHICH ITEMS DID ITSS BOOST PERFORMANCE?

4.1 CTA1

Figure 1 gives the results from model (1) fit to the middle school and to the high school sample. Each point on the plot represents the estimated effect of assignment to the CTA1 condition on the log odds of a correct answer on one posttest item. The estimates are accompanied by approximate 95% confidence intervals.

It is immediately clear that the effect of assignment to CT vary between posttest items–indeed the χ^2 likelihood ratio test rejects the null hypothesis of no treatment effect variance with $p < 0.001$ in all four strata.

In the middle school sample, the average treatment effect across items was close to 0 for both years (-0.08 in year 1 and 0.03 in year 2 on the logit scale), and not statistically significant. However, the standard deviation of treatment effects between problems was much higher—0.31 in year 1 and 0.29 in year 2, implying that assignment to CTA1 boosted performance on some problems and hurt performance on others. To interpret the standard deviation of effects on the probability scale, consider that for a marginal student, with a 1/2 probability of answering an item correctly, a difference of 0.3 between two treatment effects would correspond to a difference in the probability of a correct answer of about 7.5% (using the "divide by 4 rule" of [8] p. 82). The effects are also moderately correlated across the two years, with $\rho \approx 0.4$—items that CTA1 impacted in year 1 were somewhat likely to be similarly impacted in year 2.

Many of the treatment effects in the upper pane of Figure 1 are estimated with too much noise to draw strong

Figure 1: Estimated treatment effects of CTA1 for each level—high school or middle school—implementation year, and posttest item, with approximate 95% confidence intervals

conclusions—the sample size was substantially smaller in the middle school stratum than in the high school stratum. However, some effects are discernible: in year 1, effects were negative, and on the order of roughly 0.4 on the logit scale (0.1 on the probability scale for a marginal student) on items 1, 2, 9, 10, 12, 19, 22, and 25, and on the order of approximately 0.7 for item 17 (which asks students to match a linear equation to its graph), and similarly-sized positive effects on items 27, 30, and 32. In year 2 there were fewer clearly negative effects—on items 1 and 7—and more positive effects, such as on items 16, 18, 22, 29, and 32. There is a striking difference between the year 1 and year 2 effects on item 22, which asks students to match a quadratic expression to its graph—the effect was quite negative in year 2 and quite positive in year 2.

In the high school sample, the average treatment effect across items was roughly -0.1 in year 1 and 0.13 in year 2, on the logit scale, neither statistically significant–though the difference between the average effect in the two years was significant ($p < 0.001$). The effects varied across items, though less widely in high school than in middle school—in both years the standard deviation of item-specific effects was roughly 0.17. Item-specific effects were more highly correlated across years ($\rho \approx 0.69$)—at some points in the lower pane of Figure 1 it appears as though the curve from year 2 was simply shifted up from year 1.

The item-specific effects in the high school sample were estimated with substantially more precision than in the middle school sample, due to a larger sample size. In year 1, there were striking negative effects on items 2, 14, and 25 which ask students to manipulate algebraic expressions, and on item 12, which ask students to calculate the length of the side of a triangle. In year 2, these negative effects disappeared. Instead, there were positive effects, especially on items 8 and 22, which both ask about graphs of algebraic functions, and on a stretch of items from 15–22. The difference in the estimated effects between years was positive for all items and highest for problems 2, 20, and 25, which ask students to manipulate or interpret algebraic expressions, and 12, the triangle problem. In items 2, 12, and 25, the effect was significantly negative in year 1 and closer to zero in year 2, while for item 20 the effect was close to zero in year 1 and positive in year 2.

Figure 2 plots the estimated effect on each posttest item as a function of the item's objective in Table 1. Some patterns are notable. There was a wide variance in the effects on the four geometry problems for middle schoolers in year 1, but in year 2 all the effects on geometry items were positive and roughly the same size. The geometry items in the high school sample follow a similar, if less extreme, pattern. Across both middle and high school, the largest positive effects were for Functions and Graphs problems, especially item 22 for year 2; on items 23, 27, 31 and 32, middle schoolers—especially in year 2—saw positive effects while high schoolers saw effects near 0.

4.2 ASSISTments

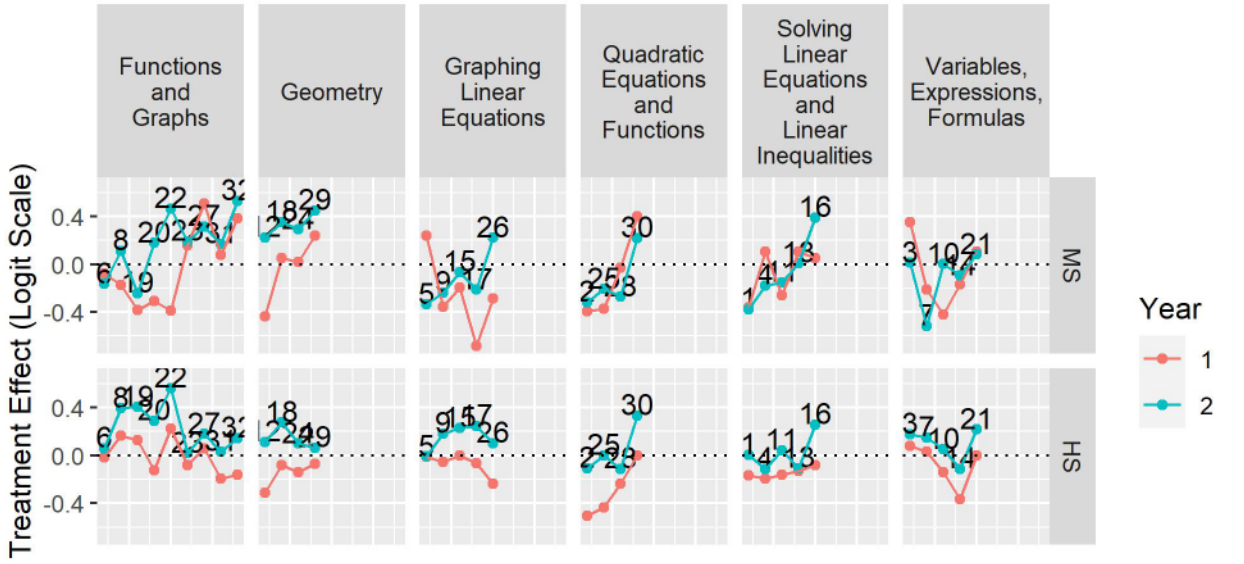


Figure 2: Estimated treatment effects of CTA1 posttest items arranged by the group of skills each item is designed to test. See Table 1 for more detail.

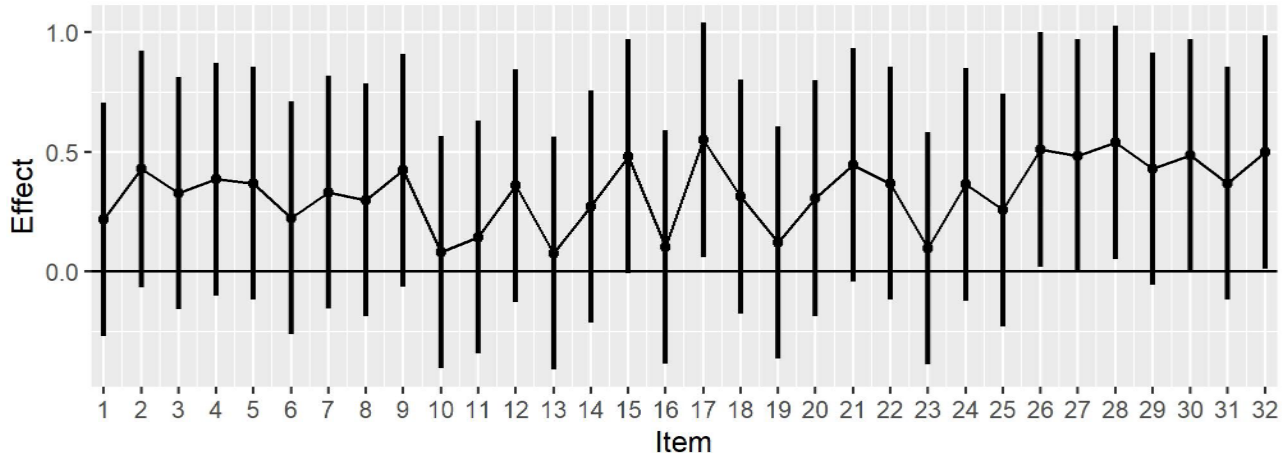


Figure 3: Estimated treatment effects of ASSISTments for each multiple choice posttest item, with approximate 95% confidence intervals

Figure 3 gives the results from model (2), plotting item-specific effect estimates with approximate 95% confidence intervals for each multiple choice TerraNova posttest item. The model estimated an average effect of 0.33, with a standard error of 0.23, for multiple choice problems. The standard deviation of item-specific effects was positive ($p<0.001$) but less than for the CTA1 items: it was estimated as 0.16 on the logit scale. The confidence intervals in Figure 3 are also much wider than those for CTA1; we suspect that a large part of the reason is that we did not have access to pretest scores, an important covariate.

The largest effects on the multiple choice items were 28 and 17, which both required students to plug in values for variables in algebraic expressions. The confidence intervals around the effects for items 26 and 32 also exclude 0.

Figure 4 plots item-specific effects for multiple choice TerraNova items grouped according to their CCSS, as in Table 2, with the non-grade-7 standards grouped together as "Other." Interestingly, the largest effects tended to be for items in this "Other" category—as did the smallest effect, for item 13. Effects for problems in the "Number System" and "Ratios and Proportional Relationships" categories had the most consistent effects, between 0.2 and 0.4 on the logit scale.

5. EXPLORING HYPOTHESES ABOUT *WHY* ASSISTMENTS EFFECTS DIFFERED

Researchers on the ASSISTments team have built on the CCSS links of Table 2, linking TerraNova posttest items to data on student work within ASSISTments, for students in the treatment condition. This gives us an opportunity to use student work within ASSISTments to explain some of the variance in treatment effects.

Like TerraNova items in Table 2, ASSISTments problems are linked with CCSS. By observing which problems treatment students worked on, and using this linkage, we could observe which Common Core standards they worked on the most within ASSISTments. We hypothesized that treatment effects might be largest for the TerraNova problems that were linked with the Common Core standards students spent the most time working on. In other words, we linked TerraNova items with worked ASSISTments problems *via* Common Core standards. The Common Core linkage we used in this segment was finer-grained than Table 2, so TerraNova items in the same category in Table 2 may not be linked with the same problems in this analysis.

We examined our hypothesis in two ways: examining the relationships between treatment effects and the number of related ASSISTments problems students in the treatment group worked, and the number of related ASSISTments problems students in the treatment group worked *correctly*. This analysis includes two important caveats: first, the linkages, both between TerraNova items and CCSS, and between ASSISTments problems and CCSS, were subjective and error-prone, possibly undermining the linkage between TerraNova items and ASSISTments problems. Secondly, student work in ASSISTments is necessarily a post-treatment variable—it was affected by treatment assignment. If the treatment randomization had fallen out differently, different schools would have been assigned to the ASSISTments condition and different ASSISTments problems would have been worked. Including the number of worked or correct related problems as a predictor in a causal model risks undermining causal interpretations [14].

Figures 5 and 6 plot estimated item-specific effects for multiple choice TerraNova items against the number of ASSISTments problems that students in the treatment arm worked or worked correctly, respectively, over the course of the RCT. The X-axis is on the square-root scale, and a loess curve is added for interpretation. Little, if any, relationship is apparent in either figure, suggesting either the lack of a relationship between specific ASSISTments work and posttest items, or issues with the linkage. This is hardly surprising, given both the difficulty in linking ASSISTments and TerraNova problems, and given the fact that topics in mathematics are inherently connected, so that improving one skill tends to improve others as well.

6. CONCLUSIONS

Education researchers are increasingly interested in "what works." However, the effectiveness of an intervention is necessarily multifaceted and complex—effects differ between students, as a function of implementation [24], and, potentially, as a function of time and location. In this paper we explored a different sort of treatment effect heterogeneity—differences in effectiveness for different outcomes—specifically, different posttest items measuring different skills. Collapsing item-level posttest data into a single test score has the advantage of simplicity (which is nothing to scoff at, especially in complex causal scenarios) but at a cost. Analysis using only summary test scores squanders a potentially rich source of variability and information about intervention effectiveness that is already at our fingertips. There is little reason *not* to examine item-specific effects.

In this paper, we showed how to estimate item specific effects using a Bayesian or empirical Bayesian multilevel modeling approach that, we argued, can improve estimation precision and avoid the need for multiplicity corrections. The estimates we provided here combine maximum likelihood estimation and empirical Bayesian inference; there is good reason to suppose that a fully Bayesian approach would provide greater validity, especially in standard error estimation and inference. However, fitting complex multilevel models using Markov Chain Monte Carlo methods is computationally expensive, and can be very slow, even with the latest software. We hope to explore this option more fully in future work.

While estimating item-specific effects is relatively straightforward, interpreting them presents a significant challenge. This is due to a number of factors: first, when looking for trends in treatment effects by problem attributes, the sample size is the number of exam items, not the number of students, so patterns can be hard to observe and verify. Secondly, there is a good deal of ambiguity and subjectivity involved in defining and determining item attributes and features, which is exacerbated by the fact that standardized tests generally cannot be made publicly available. Lastly, since student ITS work over the course of a study is necessarily post-treatment assignment, careful causal modeling (such as principal stratification [24]) may be necessary. Ex-

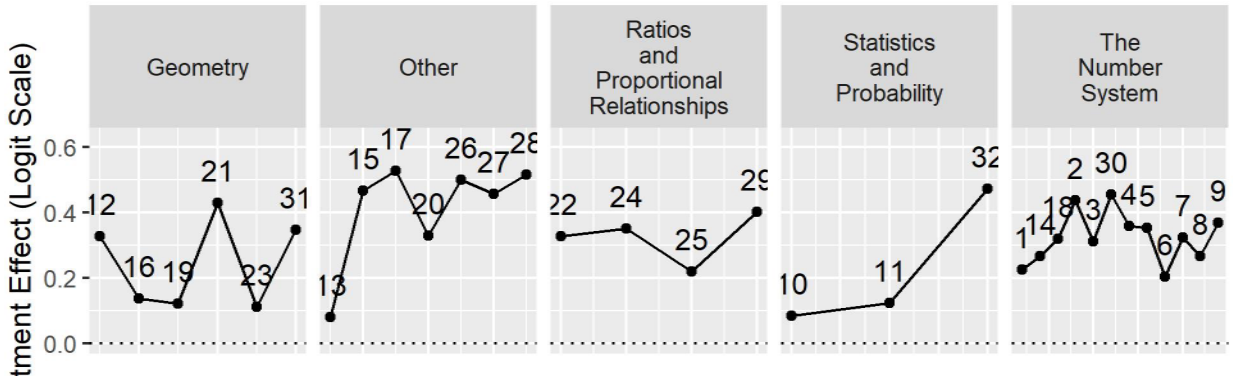


Figure 4: Estimated treatment effects of ASSISTments for each multiple choice posttest item, arranged according to CCSS, as in Table 2. The "Other" category includes Functions and the two Mathematical Practice standards, "make sense of problems and persevere in solving them" and "reason abstractly and quantitatively".

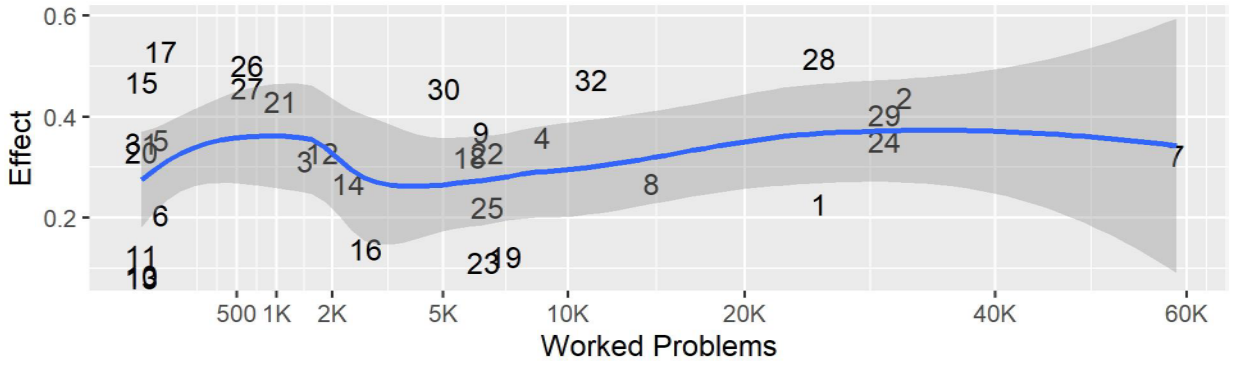


Figure 5: Estimated effects on multiple-choice TerraNova items plotted against the number of related ASSISTments problems that students in the treatment arm worked over the course of the study. The X-axis is plotted on the square-root scale, and a non-parametric loess fit is added for interpretation.

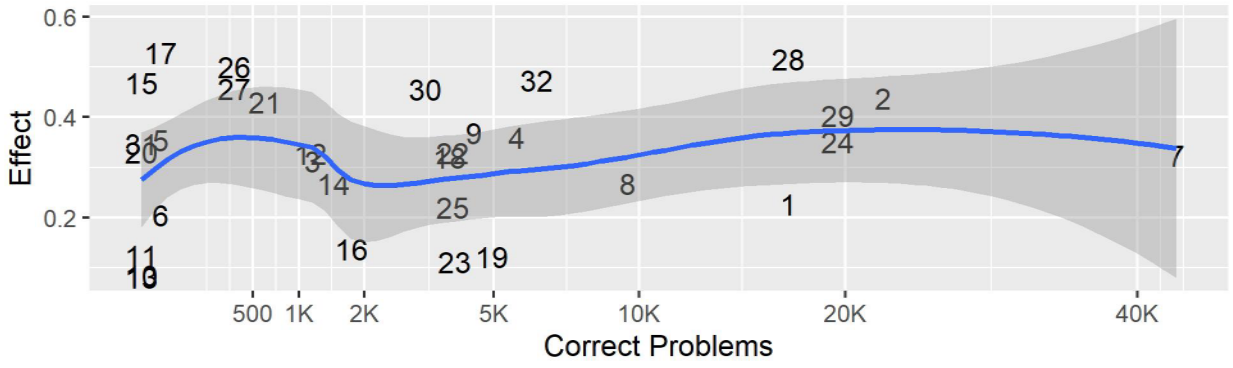


Figure 6: Estimated effects on multiple-choice TerraNova items plotted against the number of related ASSISTments problems that students in the treatment arm worked correctly over the course of the study. The X-axis is plotted on the square-root scale, and a non-parametric loess fit is added for interpretation.

amining heterogeneity between item-specific treatment effects may play a larger role in helping to generate hypotheses about ITS effectiveness than in confirming hypotheses.

Despite those difficulties, the analysis here uncovered important information about the CTA1 and ASSISTments effects. First, the discovery that the effects vary between items is notable in itself. In our analysis of CTA1 we noticed that some of the largest effects—and differences between first and second-year effects— were for posttest items involving manipulating algebraic expressions and interpreting graphs. In our analysis of ASSISTments, we discovered a large difference between negative effects on open-ended questions and positive effects on multiple choice questions, and also that the largest effects were on problems requiring students to plug numbers into algebraic expressions.

We hope that this research will serve as a proof-of-concept and spur further work delving deeper into data we already have.

7. REFERENCES

[1] A. Agresti. *Categorical data analysis*, volume 482. John Wiley & Sons, 2003.

[2] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207, 1995.

[3] D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

[4] H. S. Bloom, S. W. Raudenbush, M. J. Weiss, and K. Porter. Using multisite experiments to study cross-site variation in treatment effects: A hybrid approach with fixed intercepts and a random treatment coefficient. *Journal of Research on Educational Effectiveness*, 10(4):817–842, 2017.

[5] G. Casella. An introduction to empirical bayes data analysis. *The American Statistician*, 39(2):83–87, 1985.

[6] CTB/McGraw-Hill. Acuity algebra proficiency technical report. Monterey, CA, 2007.

[7] M. Escueta, V. Quan, A. Nickow, and P. Oreopoulos. Education technology: An evidence-based review. *NBER Working Paper*, (w23744), 2017.

[8] A. Gelman and J. Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.

[9] A. Gelman, J. Hill, and M. Yajima. Why we (usually) don't have to worry about multiple comparisons. *Journal of Research on Educational Effectiveness*, 5(2):189–211, 2012.

[10] N. T. Heffernan and C. L. Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[11] A. Israni, A. C. Sales, and J. F. Pane. Mastery learning in practice: A (mostly) descriptive analysis of log data from the cognitive tutor algebra i effectiveness trial, 2018.

[12] R. Karam, J. F. Pane, B. A. Griffin, A. Robyn, A. Phillips, and L. Daugherty. Examining the implementation of technology-based blended algebra i curriculum at scale. *Educational Technology Research and Development*, 65(2):399–425, 2017.

[13] J. A. Kulik and J. Fletcher. Effectiveness of intelligent tutoring systems: a meta-analytic review. *Review of educational research*, 86(1):42–78, 2016.

[14] J. M. Montgomery, B. Nyhan, and M. Torres. How conditioning on posttreatment variables can ruin your experiment and what to do about it. *American Journal of Political Science*, 62(3):760–775, 2018.

[15] C. N. Morris. Parametric empirical bayes inference: theory and applications. *Journal of the American statistical Association*, 78(381):47–55, 1983.

[16] National Governors Association Center for Best Practices, Council of Chief State School Officers. Common core state standards: Mathematics, 2010.

[17] J. F. Pane, B. A. Griffin, D. F. McCaffrey, and R. Karam. Effectiveness of cognitive tutor algebra i at scale. *Educational Evaluation and Policy Analysis*, 36(2):127–144, 2014.

[18] J. F. Pane, D. F. McCaffrey, M. E. Slaughter, J. L. Steele, and G. S. Ikemoto. An experiment to evaluate the efficacy of cognitive tutor geometry. *Journal of Research on Educational Effectiveness*, 3(3):254–281, 2010.

[19] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.

[20] G. Rasch. *Probabilistic models for some intelligence and attainment tests.* ERIC, 1993.

[21] S. W. Raudenbush and A. S. Bryk. *Hierarchical linear models: Applications and data analysis methods*, volume 1. sage, 2002.

[22] J. Roschelle, M. Feng, R. F. Murphy, and C. A. Mason. Online mathematics homework increases student achievement. *AERA open*, 2(4):2332858416673968, 2016.

[23] A. Sales, T. Patikorn, and N. T. Heffernan. Bayesian partial pooling to improve inference across a/b tests in edm. In *Proceeding of the Educational Data Mining Conference*, 2018.

[24] A. Sales, A. Wilks, and J. Pane. Student usage predicts treatment effect heterogeneity in the cognitive tutor algebra i program. In *Proceedings of the 9th International Conference on Educational Data Mining. International Educational Data Mining Society*, pages 207–214, 2016.

[25] W. J. van der Linden and R. K. Hambleton. *Handbook of modern item response theory*. Springer Science & Business Media, 2013.

APPENDIX

A. A SIMULATION STUDY OF MULTIPLE COMPARISONS

We ran a small simulation study testing [9]'s assertion that multiplicity corrections are unnecessary when estimating different effects from BLUPs in a multilevel model. [9] stated their case in terms of fully Bayesian models, whereas we used an empirical Bayesian approach that may differ somewhat.

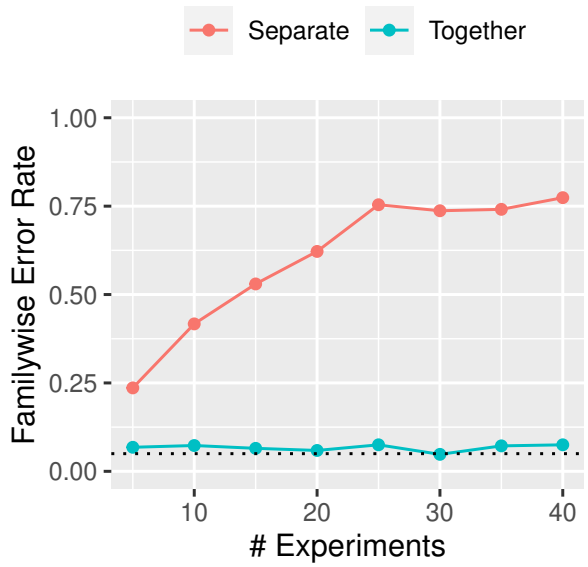


Figure 7: United we stand: results from a simulation of familywise error rate using separate t-tests for each experiment or using multilevel modeling.

In our simulation, in each simulation run, we generated data on $Nexpr$ experiments, where $Nexpr$ was a parameter we varied. In each experiment, there were $n = 500$ simulated subjects, half assigned to treatment and half to control. They were given "outcome" data $Y \sim N(0, 1)$, with no treatment effect.

We analyzed the experiment data in two ways. First, we estimated a p-value for each experiment separately, using t-tests. This is the conventional approach. Then, we we estimated a multilevel model:

$$Y_{ij} = \beta_0 + \gamma_{1j} Expr_j + \gamma_{2j} Trt_i + \epsilon_{ij}$$

where β_0 is an intercept, γ_{1j} are random intercepts for experiment, γ_{2j} is the treatment effect for experiment j, and ϵ_{ij} is a normally-distributed error term. $\boldsymbol{\gamma} \sim MVN\left(\{0, \gamma_{20}\}, \Sigma\right)$ where γ_{20} is the average effect across all experiments. The number of experiments in each simulation run, $Nexpr$, was varied from 5 to 40, in increments of 5. In each case, we estimated the familywise error rate, the probability of at least one statistically significant effect estimate (at $\alpha = 0.05$) across the $Nexpr$ experiments.

The results are in Figure 7. As expected, the familywise error rate increased rapidly when effects were estimated and tested separately in each of the $Nexpr$ experiments. When effects were estimated jointly in a multilevel model, in a way analogous to the method described in Section 3, the familywise error rate remained roughly constant as $Nexpr$ increased. However, the familywise error rate in the multilevel modeling approach was slightly elevated, ranging from roughly 0.05 to 0.075.

Chapter 1.4

Classifying Math Knowledge Components via Task-Adaptive Pre-Trained BERT

Classifying Math Knowledge Components via Task-Adaptive Pre-Trained BERT

Jia Tracy Shen[1], Michiharu Yamashita[1], Ethan Prihar[2], Neil Heffernan[2], Xintao Wu[3], Sean McGrew[4], and Dongwon Lee[1]

[1] Penn State University, University Park, PA 16802, USA
jqs5443@psu.edu
[2] Worcester Polytechnic Institute, Worcester, MA 01609, USA
ebprihar@gmail.com
[3] University of Arkansas, Fayetteville, AR 72701, USA
xintaowu@uark.edu
[4] K12.com, Herndon, VA 20170, USA
smcgrew@k12.com

Abstract. Educational content labeled with proper knowledge components (KCs) are particularly useful to teachers or content organizers. However, manually labeling educational content is labor intensive and error-prone. To address this challenge, prior research proposed machine learning based solutions to auto-label educational content with limited success. In this work, we significantly improve prior research by (1) expanding the input types to include KC descriptions, instructional video titles, and problem descriptions (i.e., three types of prediction task), (2) doubling the granularity of the prediction from 198 to 385 KC labels (i.e., more practical setting but much harder multinomial classification problem), (3) improving the prediction accuracies by 0.5-2.3% using Task-adaptive Pre-trained BERT, outperforming six baselines, and (4) proposing a simple evaluation measure by which we can recover 56-73% of mispredicted KC labels. All codes and data sets in the experiments are available at: https://github.com/tbs17/TAPT-BERT

Keywords: BERT · Knowledge Component · Text Classification · NLP

1 Introduction

In the math education community, teachers, Intelligent Tutoring Systems (ITSs) and Learning Management Systems (LMSs) have long focused on bringing learners to the target mastery over a set of skills, also known as **Knowledge Components (KCs)**. Common Core State Standards (CCSS)[5] is one of the most common categorizations of knowledge components skills in mathematics from kindergarten to high school in the United States with a full set of 385 KCs. For example, in the CCSS code *7.NS.A.1*, *7* stands for 7-th grade, *NS* stands for the domain *Number system*, *A.1* stands for the standard number of the code [5].

[5] www.corestandards.org

Table 1: Examples of three data types, all having the KC label "8.EE.A.1"

Data Type	Text
Description Text	Know and apply the properties of integer exponents to generate equivalent numerical expressions
Video Title	Apply properties of integer exponents to generate equivalent numerical expressions
Problem Text	Simplify the expression: (z2)2 *Put parentheses around the power if next to coefficient, for example: 3x2=3(x^2),x5=x^5

In the process of using KCs, the aforementioned stakeholders often encounter the challenges in three scenarios: (1) teachers need to know what KCs a student is unable to master by describing the code content (S_1), (2) ITSs need to tag instructional videos with KCs for better content management (S_2), and (3) LMSs need to know what KCs a problem is associated with in recommending instructional videos to aid problem solving (S_3).

The solutions to these scenarios typically framed the problem as the *multi-nominal classification*–i.e., given the input text, predicts one most relevant KC label out of many KCs: $I(nput) \mapsto text$ and $O(utput) \mapsto KC$. Prior research solutions included SVM-based [12], Non-negative Matrix Factorization (NMF) [6], Skip-gram Representation [17], Neural Network [18] or even cognitively-based knowledge representation [20]. Existing solutions, however, used relatively small number of labels (e.g., 39 or 198) from CCSS with the input of problem text only (similar to Table 1-Row 3) [17,12,18].

Toward this challenge, in this work, we significantly improve existing methods in auto-labeling educational content. First, based on three scenarios of S_1, S_2, and S_3, we consider three types of input, including KC descriptions, instructional video titles, and problem text (as shown in Table 1). Second, we solve the multinomial classification problem with 385 KC labels (instead of 198). Note that the problem becomes much harder. Third, we adopt the *Task-adpative Pre-trained* (TAPT) BERT [9] in solving the multinomial classification problem. Our solution outperforms six baselines, including three classical machine learning (ML) methods and two prior approaches, improving the prediction accuracies by 0.5-2.3% for the tasks of S_1, S_2, and S_3, respectively. Finally, we propose a new evaluation measure, *TEXSTR*, that enables 56-69% more KC labels to be correctly predicted than using the classical measure of *accuracy*.

2 Related Work

KC Models. Rose et al. [20] is one of the earliest work predicting knowledge components, which took a cognitively-based knowledge representation approach. The scale of KCs it examined was small with only 39 KCs. Later research extended the scale of KCs using a variety of techniques. For example, Desmariais [6] used non-negative matrix factorization to induce Q-matrix [3] from simulated data and obtained an accuracy of 75%. The approach did not hold when applying to real data and only got an accuracy of 35%. The two aforementioned studies

shared the same drawback: not using the texts from the problems. Karlovcec et al. [12] used problem text data from the ASSISTments platform [10] and created a 106-KC model using 5-fold cross validation via ML approach SVM, achieving top 1 accuracy of 62.1% and top 5 accuracy of 84.2%. Pardos et al. [17] predicted for 198 labels and achieved 90% accuracy via Skip-gram word embeddings of problem id per user (no problem text used). However, Patikorn et al. [18] did a generalizability study of Pardos et al. [17]'s work and only achieved 13.67% accuracy on a new dataset. They found that was because Pardos et al. [17]'s model was over-fitting due to memorizing the question templates and HTML formatting as opposed to encoding the real features of the data. Hence, Patikorn et al. [18] removed all the templates and HTML formatting and proposed a new model using Multi-Layer-Perceptron algorithm, which achieved 63.80% testing accuracy and 22.47% on a new dataset. The model of Patikon et al. [18] became the highest performance for the type of problem text. The preceding research is only focused on problem related content (ID or texts) whereas our work uses not only the problem text but also the KC descriptions and video title data covering a broad range of data.

Pre-Trained BERT Models. The state-of-the-art language model BERT (Bidirectional Encoder Representations From Transformer) [7] is a pre-trained language representation model that was trained on 16 GB of unlabeled texts including Books Corpus and Wikipedia with a total of 3.3 billion words and a vocabulary size of 30,522. Its advantage over other pre-trained language models such as ELMo [19] and ULMFiT [11] is its bidirectional structure by using the *masked language model* (MLM) pre-training objective. The MLM randomly masks 15% of the tokens from the input to predict the original vocabulary id of the masked word based on its context from both directions [7]. The pre-trained model then can be used to train from new data for tasks such as text classification, next sentence prediction.

Users can also further pre-train BERT model with their own data and then fine-tune. This combining process has become popular in the past two years as it can usually achieve better results than fine-tuning only strategy. Sun et al. [21] proposed a detailed process on how to further pre-train new texts and fine-tune for classification task, achieving a new record accuracy. Models such as FinBERT [16], ClinicalBERT [1], BioBERT [15], SCIBERT [2], and E-BERT [23] that were further pre-trained on huge domain corpora (e.g.billions of news articles, clinical texts or PMC Full-text and abstracts) were referred as *Domain-adaptive Pre-trained* (DAPT) BERT and models further pre-trained on task-specific data are referred as *Task-adaptive Pre-trained* (TAPT) BERT by Gururangan et al. [9] such as MelBERT [4] (Methaphor Detection BERT). Although DAPT models usually achieve better performance (1-8% higher), TAPT models also demonstrated competitive and sometimes even higher performance (2% higher) according to Gururangan et al. [9]. In Liu et al. [16], FinBERT-task was 0.04% higher than domain FinBERT in accuracy. In addition, TAPT requires less time and resource to train. In light of this finding, we use the task-specific data to further pre-train the BERT model.

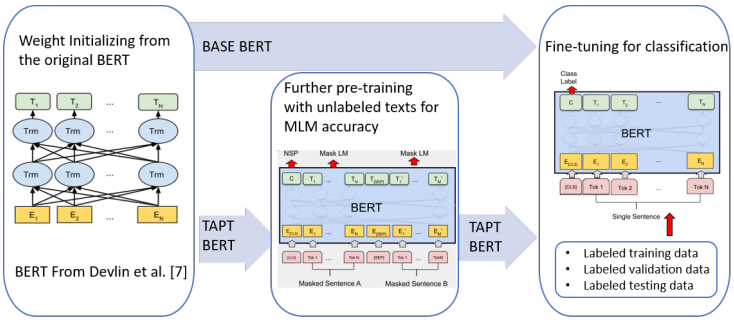


Fig. 1: An illustration of training and fine-tuning process of BASE vs. TAPT

3 The Proposed Approach

To improve upon existing solutions to the problem of auto-labeling educational content, we propose to exploit recent advancements by BERT language models. Since BERT can encode both linguistic structures and semantic contexts in texts well, we hypothesize its effectiveness in solving the KC labeling problem. By effectively labeling the KCs, we expect to solve the challenges incurred from three scenarios in Section 1.

3.1 Task-Adpative Pre-Trained (TAPT) BERT

In particular, we propose to adopt the Task-adaptive Pre-trained (TAPT) BERT and fine-tune it for three types of data. The "pre-training" process is unsupervised such that unlabeled task-specific texts get trained for MLM objective whereas the "fine-tuning" process is supervised such that labeled task-specific texts get trained for classification (see Fig. 1). We call a BERT model that only has a fine-tuning process as BASE. For TAPT, we first initialize the weights from the original BERT (i.e., BERT-base-uncased model). Then, we further pre-train the weights using the unlabeled task-specific texts as well as the combined task texts (see detail in Section 4.1) for MLM objective, a process of randomly masking off 15% of the tokens and predict their original vocabulary IDs. The pre-training performance is measured by the accuracy of MLM. Once TAPT is trained, we fine-tune TAPT with the task-specific labeled texts by splitting them into training, validation and testing datasets and feed them into the last softmax layer for classification. We measure the performance of fine-tuning via the testing data accuracy. For BASE, we do not further train it after initializing the weights but directly fine-tune it with the task-specific data for classification (see Fig. 1). To show the effectiveness of the TAPT BERT approach, we compare it against six baselines including BASE BERT for three tasks:

- T_d: to predict K-12 KCs using dataset D_d (description text) based on S_1
- T_t: to predict K-12 KCs using dataset D_t (video title text) based on S_2
- T_p: to predict K-12 KCs using dataset D_p (problem text) based on S_3

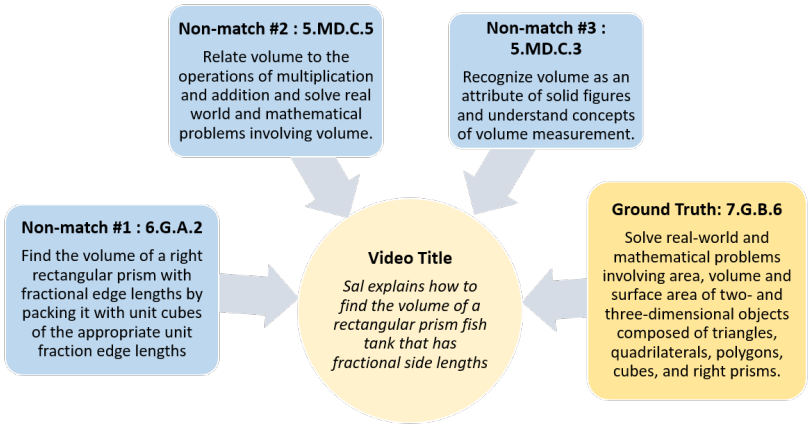


Fig. 2: An illustration of multiple possibilities of a correct label for a given video title text

3.2 Evaluating KC Labeling Problem Better: *TEXSTR*

In the regular setting of multinomial classification to predict KC labels, the evaluation is done as binary–i.e., exact-match or non-match. For instance, if a method predicts a KC label to be *7.G.B.6*, but its ground truth is *7.G.A.5*, *7.G.B.6* is considered to be a non-match. However, the incorrectly predicted label of *7.G.B.6* could be closely related to *7.G.A.5* and thus still be useful to teachers or content organizers. For example, in Fig. 2, the input to the classification problem is a video title "Sal explains how to find the volume of a rectangular prism fish tank that has fractional side lengths." Its ground truth label is *7.G.B.6* (7-th grade geometry KC), described as "Solve real world problem involving ... volume ... composed of ... prisms." When one looks at three non-match labels, however, their descriptions do not seem to be so different (see in Fig. 2). That is, all of the three non-match labels (*6.G.A.2*, *5.MD.C.5*, and *5.MD.C.3*) mention "volume solving" through "fine/relate/recognize with operations and concepts," which is quite similar to the KC description of the ground truth. However, due to the nature of exact-match based evaluation, these three labels are considered wrong predictions. Further, domain experts explain that some skills are prerequisites to other skills, or that some problems have more than one applicable skills (thus multiple labels) and they could all be correct.

Therefore, we argue that using a strict exact-matching based method in evaluating the quality of the predicted KC labels might be insufficient in practical settings. We then propose a method that considers both semantic and structural similarities among KC labels and their descriptions to be an additional measure to evaluate the usability of the predicted labels.

– Semantic Similarity (C_t): We adopt the Doc2Vec algorithm [14] to capture the similarity between KC labels. Doc2Vec, derived from word-vector algorithm, generates similarity scores between documents instead of words and

is proved to have lower error rate (7.7-16%) than the word vector approach [14].
- Structural Similarity (C_s): We exploit prerequisite relationships among skills (KC labels) and capture such as edges and KC labels as nodes in a graph. The prerequisite relationships are extracted from a K-G8 math coherence map by Jason Zimba [24] and a high school (G9-G12) coherence map by UnboundEd Standard Institue [22]. Then, we adopt Node2Vec algorithm [8] that is efficient and flexible in exploring nodes similarity and achieved a new record performance in network classification problem [8].

In the end, we craft a new evaluation measure, named as *TEXSTR* (Λ), by combining both C_t and C_s as follows: $\Lambda = \alpha \cdot C_t + (1-\alpha) \cdot C_s$, where α controls the weight between C_t and C_s as an oscillating parameter.

4 Empirical Validation

4.1 Datasets and Evaluation Measure

Table 2 summarizes the details of the datasets for pre-training and fine-tuning processes. D_d contains 6,384 description texts (84,017 tokens) and 385 math KCs (an example shown in Fig. 1-a). Part of D_d are extracted from Common Core Standards website[6] and part are provided by k12.com[7], an education management organization that provides online education to American students from kindergarten to Grade 12. D_t contains 6,748 video title texts (62,135 tokens) and 272 math KCs (an example shown in Fig. 1-b) Part of D_t are extracted from *Youtube.com* (via youtube DataAPI[8]) and part are provided by k12.com. D_p contains 13,722 texts (589,549 tokens) and 213 math KCs provided by ASSISTments[9] (an example shown in Fig. 1-c). Further, D_{d+t}, D_{d+p}, D_{t+p}, and D_{all} are different combinations of the unlabeled texts from D_d, D_t, and D_p. They are only used in the TAPT pre-training process. We pre-process all aforementioned texts by removing all the templates and HTML markups to avoid over-fitting, suggested by the prior highest accuracy method [18]. In the TAPT pre-training process, 100% of the unlabeled texts from the aforementioned datasets are used for pre-training. In fine-tuning process for both TAPT and BASE , only D_d, D_t, and D_p are used and 72% of their texts and labels are used for training, 8% are for validation and 20% are for testing (see in Table 2 Row 1-3 and Col. 6-8).

As an evaluation measure, following prior research [18,17,20,6,12] for direct comparison, we use Accuracy@k as (TP + TN)/(TP + TN + FP + FN), when a method predicts top-k KC labels. Further, we evaluate our method using the proposed *TEXSTR* measure.

[6] http://www.corestandards.org/math
[7] http://www.k12.com
[8] http://developers.google.com/youtube/v3
[9] http://www.assistments.org/

Table 2: A summary statistics of datasets.

Name	# Labels	# Texts	# Tokens	Fine-tuning Partition		
				Training (72%)	Validation (8%)	Testing (20%)
D_d	385	6,384	84,017	4,596	511	1,277
D_t	272	6,748	62,135	4,858	540	1,350
D_p	213	13,722	589,549	9,879	1,098	2,745
D_{d+t}	/	13,132	146,152	/	/	/
D_{d+p}	/	20,106	673,566	/	/	/
D_{t+p}	/	20,470	651,684	/	/	/
D_{all}	/	26,854	735,701	/	/	/

4.2 Pre-training and Fine-tuning Details

To further pre-train, we follow the same pre-training process of original BERT with the same network architecture (12 layers, 768 hidden dimensions, 12 heads, 110M parameters) but on our own unlabeled task-specific texts (see Col. 4 in Table 2). With an 8-core v3 TPU, we further train all our models with 100k steps, achieving MLM accuracy of above 97% that lasts about 1-4 hours. We experiment hyper-parameters such as learning rate (lr) $\in \{1e-5, 2e-5, 4e-5, 5e-5, 2e-4\}$, batch size (bs) $\in \{8, 16, 32\}$, and max-sequence length (max-seq-len) $\in \{128, 256, 512\}$. The highest MLM accuracy was achieved when lr $\leftarrow$ 2e-5, bs $\leftarrow$ 32, and max-seq-len $\leftarrow$ 128 (for D_d and D_t) and max-seq-len $\leftarrow$ 512 with the same lr and bs (for D_p, D_{d+p}, D_{t+p}, D_{all}). To fine-tune, we also follow the original BERT script by splitting D_d, D_t, D_p into 72% for training, 8% for validation and 20% for testing per task. We experiment ep $\in \{5, 10, 25\}$ due to the small size of the data size and retain the same hyper-parameter search for lr, bs, max-seq-len. We find that the best testing accuracy is obtained when ep $\leftarrow$ 25, lr $\leftarrow$ 2e-5, bs $\leftarrow$ 32, and max-seq-len $\leftarrow$ 128 for D_d, D_t whereas the best testing accuracy for D_p is obtained when ep $\leftarrow$ 25, lr $\leftarrow$ 2e-5, bs $\leftarrow$ 32, and max-seq-len $\leftarrow$ 512. We find that after ep $\leftarrow$ 25, it is difficult to gain significant increase on the testing accuracy. Hence, the optimal hyper-parameters while task-dependent seem to have very minimal change across tasks. This finding is consistent with SCIBERT reported [2].

4.3 Result #1: TAPT BERT vs. Other Approaches

Table 3 summarizes the experimental results of six baseline approaches and TAPT for each task. For baseline methods, we group them into categories (see in Table 3) (1) classical ML, (2) prior work, and (3) BASE BERT. By including popular ML methods such as Random Forest and XGBoost, we aim to compare its performance to the one from prior ML work (SVM) proposed by Karlovec et al [12] in the literature review. As to comparing to the prior highest accuracy method [18], we applied the same 5-fold cross-validation on our own problem texts and obtain Acu@1 and Acu@3. Overall, we see that TAPT models outperform all other methods at both Acu@1 and Acu@3 across three tasks. Note TAPT models here are simply trained on the unlabeled texts from D_d, D_t, and

Table 3: Accuracy comparison (best and 2nd best accuracy in blue bold and underlined, respectively, $BL\dagger$ for baseline best, and * for statistical significance with p-value < 0.001)

Approach Type	Algorithm	D_d		D_t		D_p	
		Acu@1	Acu@3	Acu@1	Acu@3	Acu@1	Acu@3
Classical ML	SVM [12]	44.87	70.40	48.15	70.30	78.07	87.69
	XGBoost	43.07	71.34	45.33	66.15	77.63	87.94
	Random Forest	49.26	78.78	49.33	74.37	78.03	88.23
Prior Work	Skip-Gram NN [17]	34.07	34.15	43.00	43.52	76.88	77.06
	Sklearn MLP [18]	50.53	74.41	48.22	57.95	80.70	81.13
BERT	BASE	48.30	76.40	50.99	76.55	81.73	90.99
	TAPT	**50.60**	**79.29**	**52.71**	**78.83**	**82.43**	**92.51**
Improvement	$\|TAPT - BL\dagger\|$	0.07	0.51	1.72	2.28	0.70	1.52
	$\|TAPT - BASE\|$	2.30*	0.51*	1.72*	2.28*	0.70*	1.52*

D_p. Compared to the best method in baseline, TAPT has an increase of 0.70%, 1.72%, 0.07% at Acu@1 and 0.51%, 2.28%, 1.52% at Acu@3 across three tasks. Compared to BASE, TAPT shows an increase of 2.30%, 1.72%, 0.70% at Acu@1 and 0.51%, 2.28%, 1.52% at Acu@3 across three tasks. Acu@1 and Acu@3 from both TAPT and BASE models are the average performance over five random seeds with significant difference (see last row in Table 3). BERT variants such as FinBERT [16], SCIBERT [2], BioBERT [15] and E-BERT [23] were able to achieve a 1-4% increase when further trained on much larger domain knowledge corpus (i.e. 2-14 billion tokens). Our corpus although comparatively small with D_d (84,017 tokens), D_t (62,135 tokens), and D_p (589,549 tokens) still result in a decent improvement of 0.51-2.30%.

4.4 Result #2: Augmented TAPT and TAPT Generalizability

In addition to the simply trained TAPTs (referred as simple TAPT) in Table 3, we augment the pre-training data and form another four TAPTs ($TAPT_{d+t}$, $TAPT_{d+p}$, $TAPT_{t+p}$ and $TAPT_{all}$). We call them augmented TAPT. Table 4 showcases the differences in Acu@3 between simple and augmented TAPT. For D_d, augmented $TAPT_{d+p}$ outperforms all simple TAPT models (Acu@3 = 79.56%) and augmented $TAPT_{d+t}$ achieves the second best Acu@3 (79.40%). For D_t, all the augmented TAPT models only outperform simple $TAPT_p$. For D_p, augmented $TAPT_{t+p}$ outperforms all simple TAPTs with Acu@3 of 92.64%. To sum up, augmenting the pre-training data for TAPT seems to help increase the accuracy further.

Furthermore, we compare the generalizability of TAPT to BASE over different datasets. We define the *generalizability* as task accuracy (specifically Acu@3) that a model can obtain when applied to a different dataset. Both BASE and TAPT are pre-trained models and obtain task accuracy via fine-tuning on a different task data. The subscripts in Table 4 present the difference in Acu@3 between TAPT and BASE, showcasing who has stronger generalizability (− sign

Table 4: Acu@3: BASE vs. TAPT. (best and 2nd best per row in bold and underlined, and subscripts indicate outperformance over BASE)

Data	BASE	Simple			Augmented			
		$TAPT_d$	$TAPT_t$	$TAPT_p$	$TAPT_{d+t}$	$TAPT_{d+p}$	$TAPT_{t+p}$	$TAPT_{all}$
D_d	76.40	$79.29_{2.89}$	$78.78_{2.38}$	$77.84_{1.44}$	$\underline{79.40}_{3.00}$	$\mathbf{79.56}_{3.16}$	$79.01_{2.61}$	$79.01_{2.61}$
D_t	76.55	$\underline{77.85}_{1.30}$	$\mathbf{78.83}_{2.28}$	$76.30_{-0.25}$	$77.56_{1.01}$	$77.56_{1.01}$	$77.70_{1.15}$	$77.78_{1.23}$
D_p	90.99	$91.22_{0.23}$	$91.44_{0.45}$	$\underline{92.51}_{1.52}$	$92.06_{1.07}$	$92.50_{1.51}$	$\mathbf{92.64}_{1.65}$	$92.35_{1.36}$

indicates weak generalizability). For D_d, all simple and augmented TAPT models generalize better than BASE, especially augmented TAPTs have an average of about 3% increase. For D_t, all TAPT models have better generalizability than BASE with over 1% average increase except for $TAPT_p$. For D_p, we also see all the TAPTs generalize better than BASE model with the augmented $TAPT_{t+p}$ having the best generalizability.

4.5 Result #3: TEXSTR Based Evaluation

Following the definition of *TEXSTR* ($=\Lambda$) in Section 3.2, we vary the values of α by {0, 0.5, 1} and generate three variations of Λ for top-3 predictions. We then decide the percentage of miss-predictions to be reconsidered based on Λ value by three cut-off thresholds {0.5, 0.75, 0.9}. Before that, we make sure that the predicted labels are not subsequent to the ground truth, i.e., if the ground truth is *7.G.A.2*, a predicted label such as *8.G.A.3* shall not be reconsidered as correct because it is the skill to be learned subsequently "after" *7.G.A.2*. In such a case, we exclude predicted labels that have subsequent relations to the ground truth and calculate Λ. Table 5 presents the percentage of miss-predictions after removing the subsequent-relation labels by three Λ thresholds when $\alpha \in \{0, 0.5, 1\}$. Across three values of α and datasets, note that 56-73% of miss-predictions could be reconsidered as correct if $\Lambda > 0.5$, 5-53% of them could be reconsidered if $\Lambda > 0.75$, and 0-32% could be reconsidered if $\Lambda > 0.9$. The wide percentage range for $\Lambda \in \{0.75, 0.9\}$ infers that higher thresholds of Λ are more sensitive to the change of α.

To further ensure the *TEXSTR* measure to be useful in practice, we conduct an empirical study where eight experienced K-12 math teachers rate each pair of top-3 KC labels and the corresponding text (e.g., description, video title, or problem text) on a scale of 1 to 5. The Fleiss' kappa value to assess the multi-rater agreement among eight teachers is 0.436, which is considered as moderate agreement by Landis et al. [13]. We ensure that none of top-3 miss-predicted KCs are subsequent to ground truths and have Λ score at least 0.5. Then, we quantify the *relevance* (Υ) score as either Λ score (when $\alpha = 0.5$) or teachers' rating of [1,5] range divided by 5 (to be on the same scale as *TEXSTR*'s [0,1]). Table 6 summarizes three varying relevance scores ($\Upsilon \in \{0.5, 0.75, 0.9\}$) on the pair of top-3 predictions and the texts. For Top-1 predictions, *TEXSTR* considers all of them to have $\Upsilon > 0.5$ (due to the pre-selection) and 37.93% of all have $\Upsilon > 0.75$ and 3.45% have $\Upsilon > 0.9$. Teachers, on the other hand, think that

Table 5: % of miss-predictions recovered by *TEXSTR* (Λ)

Data	# Miss-predictions	$\Lambda > 0.5$			$\Lambda > 0.75$			$\Lambda > 0.9$		
		$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
D_d	248	70.16	68.95	72.98	52.82	24.19	8.87	32.26	2.42	0.81
D_t	240	58.33	55.83	57.5	37.92	17.08	6.67	17.08	0	1.25
D_p	166	60.84	56.63	58.43	38.55	16.27	5.42	18.67	1.2	1.2

Table 6: % of top-3 predictions by relevance (Υ) level when $\alpha = 0.5$

Υ	Top 1			Top 2			Top 3		
	Λ	Teachers	Δ	Λ	Teachers	Δ	Λ	Teachers	Δ
> 0.5	100	54.31	-45.69	100	40.95	-59.05	100	21.98	-78.02
> 0.75	37.93	43.53	5.60	20.69	27.16	6.47	6.9	13.79	6.89
> 0.9	3.45	31.03	27.58	0	13.79	13.79	0	9.48	9.48

only 54.31% of the texts have $\Upsilon > 0.5$ ($\downarrow$ 45.69% from Λ) but 43.53% have $\Upsilon > 0.75$ ($\uparrow$ 5.6% from Λ) and 31.03% have $\Upsilon > 0.9$ ($\uparrow$ 27.58% from Λ). We also find a similar pattern for Top-2 and Top-3 predictions where teachers find 6.47-6.89% more cases than *TEXSTR* that have $\Upsilon > 0.75$ and 9.48-13.79% more cases than *TEXSTR* that have $\Upsilon > 0.9$. This indicates that *TEXSTR* is more conservative than teachers in judging the relevance of KC labels to texts when $\Upsilon \in \{0.75, 0.9\}$, suggesting *TEXSTR* is effective in reassessing miss-predictions and "recover" them as correct labels in practice.

5 Conclusion

The paper classified 385 math knowledge components from kindergarten to 12th grade using three data sources (e.g., KC descriptions, video titles, and problem texts) via the *Task-adaptive Pre-trained* (TAPT) BERT model. TAPT has achieved a new record by outperforming six baselines by up to 2% at Acu@1 and up to 2.3% at Acu@3. We also compared TAPT to BASE and found the accuracy of TAPT increased by 0.5-2.3% with a significant p-value. Furthermore, the paper discovered that TAPT trained on the augmented data by combining different task-specific texts had better Acu@3 than TAPT simply trained on the individual datasets. In general, TAPT has better generalizability than BASE by up to 3% at Acu@3 across different tasks. Finally, the paper proposed a new evaluation measure *TEXSTR* to reassess the predicted KCs by taking into account semantic and structural similarity. *TEXSTR* was able to reconsider 56-73% of miss-predictions as correct for practical use.

6 Acknowledgement

The work was mainly supported by NSF awards (1940236, 1940076, 1940093). In addition, the work of Neil Heffernan was in part supported by NSF awards (1917808, 1931523, 1917713, 1903304, 1822830, 1759229), IES (R305A170137, R305A170243, R305A180401, R305A180401), EIR (U411B190024) and ONR (N00014-18-1-2768) and Schmidt Futures.

References

1. Alsentzer, E., Murphy, J.R., Boag, W., Weng, W.H., Jin, D., Naumann, T., Mcdermott, M.B.A.: Publicly Available Clinical BERT Embeddings. In: Proceedings of the 2nd Clinical Natural Language Processing Workshop. pp. 72–78 (2019), https://www.aclweb.org/anthology/W19-1909.pdf
2. Beltagy, I., Lo, K., Cohan, A.: SCIBERT: A pretrained language model for scientific text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing. pp. 3615–3620 (2019)
3. Birenbaum, M., Kelly, A.E., Tatsuoka, K.K.: Diagnosing Knowledge States in Algebra Using the Rule-Space Model. Journal for Research in Mathematics Education **24**(5), 442–459 (1993), https://www.jstor.org/stable/749153?seq=1&cid=pdf-
4. Choi, M., Lee, S., Choi, E., Park, H., Lee, J., Lee, D., Lee, J.: MelBERT : Metaphor Detection via Contextualized Late Interaction using Metaphorical Identification Theories. In: Proceedings of NAACL (2021)
5. corestandards.org: Coding the Common Core State Standards (CCSS). Tech. rep., http://www.corestandards.org/wp-content/uploads/Math_Standards.pdf
6. Desmarais, M.C.: Mapping Question Items to Skills with Non-negative Matrix Factorization. ACM SIGKDD Explorations Newsletter **13**(2) (2012), https://doi.org/10.1145/2207243.2207248
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. vol. 1, pp. 4171–4186 (2019)
8. Grover, A., Leskovec, J.: Node2vec: Scalable feature learning for networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. vol. 13-17, pp. 855–864 (2016). https://doi.org/10.1145/2939672.2939754
9. Gururangan, S., Marasovi´cmarasovi´c, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., Smith, N.A., Allen: Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)
10. Heffernan, N.T., Heffernan, C.L.: The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. International Journal of Artificial Intelligence in Education **24**(4), 470–497 (2014). https://doi.org/10.1007/s40593-014-0024-x
11. Howard, J., Ruder, S.: Universal Language Model Fine-tuning for Text Classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. pp. 328–339 (2018)
12. Karlovčec, M., Córdova-Sánchez, M., Pardos, Z.A.: Knowledge component suggestion for untagged content in an intelligent tutoring system. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **7315 LNCS**, 195–200 (2012)
13. Landis, J.R., Koch, G.G.: The Measurement of Observer Agreement for Categorical Data. Biometrics **33**(1), 159 (1977). https://doi.org/10.2307/2529310
14. Le, Q., Mikolov, T.: Distributed Representations of Sentences and Documents. In: Proceedings of the 31st International Conference on International Conference on Machine Learning. pp. II–1188–II–1196 (2014)

15. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Data and text mining BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics p. 1234–1240 (2020). https://doi.org/10.1093/bioinformatics/btz682
16. Liu, Z., Huang, D., Huang, K., Li, Z., Zhao, J.: FinBERT: A Pre-trained Financial Language Representation Model for Financial Text Mining. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence Special Track on AI in FinTech (2020)
17. Pardos, Z.A.: Imputing KCs with Representations of Problem Content and Context. In: Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization. pp. 148–155 (2017). https://doi.org/10.1145/3079628.3079689
18. Patikorn, T., Deisadze, D., Grande, L., Yu, Z., Heffernan, N.: Generalizability of methods for imputing mathematical skills needed to solve problems from texts. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **11625 LNAI**, 396–405 (2019)
19. Peters, M.E., Neumann, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of NAACL-HLT. pp. 2227–2237 (2018)
20. Rosé, C., Donmez, P., Gweon, G., Knight, A., Junker, B., Cohen, W., Koedinger, K., Heffernan, N.: Automatic and Semi-Automatic Skill Coding With a View Towards Supporting On-Line Assessment. In: Proceedings of the conference on Artificial Intelligence in Education. pp. 571–578 (2005)
21. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to Fine-Tune BERT for Text Classification? Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **11856 LNAI**(2), 194–206 (2019)
22. UnboundEd: A " Coherence Map " for the High School Standards. Tech. rep. (2017), https://www.unbounded.org/other/8610
23. Zhang, D., Yuan, Z., Liu, Y., Fu, Z., Zhuang, F., Wang, P., Xiong, H.: E-BERT: Adapting BERT to E-commerce with Adaptive Hybrid Masking and Neighbor Product Reconstruction (2020)
24. Zimba, J.: A Graph of the Content Standards. Tech. rep. (2012), https://achievethecore.org/page/844/a-graph-of-the-content-standards

Chapter 1.5

Deep Learning or Deep Ignorance? Comparing Untrained Recurrent Models in Educational Contexts

Deep Learning or Deep Ignorance? Comparing Untrained Recurrent Models in Educational Contexts

Anthony F. Botelho[1(✉)], Ethan Prihar[2], and Neil T. Heffernan[2]

[1] University of Florida, Gainesville, FL 32611, USA
abotelho@coe.ufl.edu
[2] Worcester Polytechnic Institute, Worcester, MA 01605, USA
{ebprihar,nth}@wpi.edu

Abstract. The development and application of deep learning methodologies has grown within educational contexts in recent years. Perhaps attributable, in part, to the large amount of data that is made available through the adoption of computer-based learning systems in classrooms and larger-scale MOOC platforms, many educational researchers are leveraging a wide range of emerging deep learning approaches to study learning and student behavior in various capacities. Variations of recurrent neural networks, for example, have been used to not only predict learning outcomes but also to study sequential and temporal trends in student data; it is commonly believed that they are able to learn high-dimensional representations of learning and behavioral constructs over time, such as the evolution of a students' knowledge state while working through assigned content. Recent works, however, have started to dispute this belief, instead finding that it may be the model's complexity that leads to improved performance in many prediction tasks and that these methods may not inherently learn these temporal representations through model training. In this work, we explore these claims further in the context of detectors of student affect as well as expanding on existing work that explored benchmarks in knowledge tracing. Specifically, we observe how well trained models perform compared to deep learning networks where training is applied only to the output layer. While the highest results of prior works utilizing trained recurrent models are found to be superior, the application of our untrained-versions perform comparably well, outperforming even previous non-deep learning approaches.

Keywords: Deep learning · LSTM · Echo state network · Affect · Knowledge tracing

1 Introduction

The availability of large-scale education datasets, often comprised of large numbers of interactions between learners and educational technologies over time, have coincided with an increase in applications of deep learning methodologies

M. M. Rodrigo et al. (Eds.): AIED 2022, LNCS 13355, pp. 281–293, 2022.
https://doi.org/10.1007/978-3-031-11644-5_23

to study various aspects of student learning. The data collected from massive open online courses (MOOCs), for example, researchers have been able to utilize large, complex models to study student learning strategies as well as unproductive behavior such as attrition and dropout [6,21,26]. Even beyond MOOCs, in K-12 classrooms, the adoption of educational technologies and learning platforms such as Cognitive tutor [20] and ASSISTments [13], among many others, have led to the recording and often public release of large datasets of anonymized student interaction logs. The application of deep learning, collectively referring to a growing variety of multi-layer neural network models, often require large amounts of data to learn from, assuming, of course, that these models are in fact well-structured to learn anything at all.

Due to the large number of learned parameters and often complex structure of many deep learning approaches, many researchers and developers attribute the success of these methods to their ability to learn rich high-dimensional representations of input data. While it is possible to interpret and visualize what is learned in some applications of deep learning, it is difficult to what is learned within certain deep learning model structures including, for example, recurrent neural networks (RNN) [25]; this also includes commonly applied variants of RNN such as long short term memory (LSTM) [14] and gated recurrent unit (GRU) [8] networks. These model structures are designed to learn dependencies within time-series data, which is common in educational contexts.

Prior research that suggests, contrary to initial assumptions, that many recurrent models are not learning rich representations of data. In fact, it was found that by randomizing network weights and only training the output layer (referred to in this paper as "untrained" models), such models performed nearly as well as their trained counterparts [9,24], as will be discussed further in the next section. This work seeks to build upon this prior research that has explored this phenomenon in the context of knowledge tracing [9], to compare trained and untrained recurrent models in another educational context, detecting student affect, where deep learning recurrent models have similarly been applied in recent years [3]. In addition, several related modeling approaches have been specifically designed to utilized randomized, untrained components. This work additionally explores the application of these approaches in educational contexts. Specifically, this work will address the following research questions:

1. How does the application of untrained recurrent models compare to similar trained models in detecting student affect?
2. Do the methods designed to utilize untrained components outperform other approaches in detecting student affect?
3. Do trained and untrained recurrent models exhibit an overlapping set of latent features within their hidden layers?

1.1 Representations Within Recurrent Networks

While the application of recurrent networks, or one of several common variants, has increased in recent years, it is important to examine the basic structure

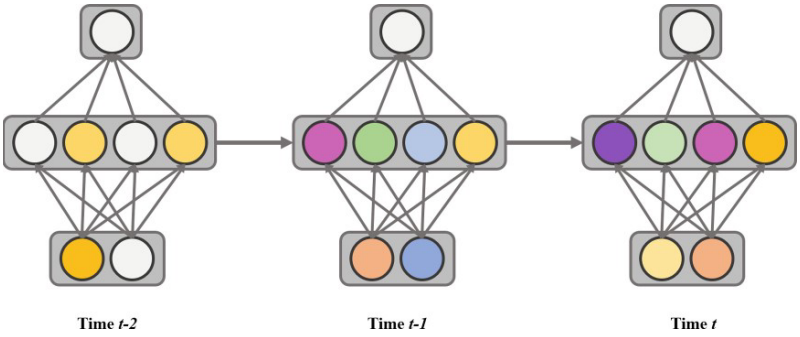


Fig. 1. This example illustrates how a recurrent network can build sequence representations within its hidden layer by combining new inputs with previous network states.

of these networks in order to better understand how such models could learn rich knowledge representations. Consider, for example, the simplified network representation depicted in Fig. 1. Like most "deep" models, recurrent networks are normally comprised of multiple layers of nodes representing values generated within the network structure. These values are calculated by multiplying the node values of earlier layers by learned weights that are traditionally fully-connected to all nodes in the subsequent layer. Unlike other network structures, recurrent models are designed to be applied to time-series data, where the values in the network's input layer (bottom layer in the figure) are combined with the previous hidden state (middle layer in the figure) for each time step. Intuitively, it is assumed that the model may learn how to combine new information with prior information within the series to make a more informed estimate for a given task; the network structure may learn how long to keep information, when to forget information, or certain conditions under which it should otherwise modify its understanding of the given sequence. The changing values within the hidden layer of the network contain and retain information from throughout the series.

How well these models are able to "understand" the given information, as represented by the values within the recurrent hidden layer, is a matter of recent speculation. It is precisely this question, as it applies to educational contexts, that is to be explored further within this work.

2 Background

In view of the application of recurrent models in education, it is important to better understand what applied models are learning from student data to understand how they can best be used to study various aspects of learning. Among the most well-known examples of using models to study learning, for example, is knowledge tracing [7]. In early models of knowledge tracing, interpretability was a primary goal; while later research has disputed how interpretable, or rather how identifiable traditional knowledge tracing models are [2,10], the structure of the models were built in alignment to learning theory. While the models themselves

are trained by predicting short-term student performance across items within a given knowledge component, the goal of these models was to build a representation of student knowledge and learning. With the development and high reported performance of deep knowledge tracing (DKT) [19], questions were raised as to whether the recurrent neural network at its core was learning a more complex representation of student knowledge over time. Among a number of subsequent works that explored how "deep" this method truly was, Yeung et al. [28] showed that DKT's representation of student knowledge contradicted traditional learning theory as well as common sense; the model seemed to believe that students fluctuated frequently between states of knowledge and non-knowledge. While the authors proposed a fix to this problem using a form of regularization during model training, this work suggests that the model is able to perform well without a strong grasp of how learning is likely to occur.

These works, however, are not the first to question whether recurrent models are able to learn rich representations within sequential and temporal data. Wieting and Kiela [24] found that untrained recurrent models could perform comparably well to trained versions in natural language processing tasks. It was suggested that the applied models act as a type of sequence encoding, rather than by embedding deeper contextual information; the models are able to learn high-dimensional *encodings* of sequential data, but may be ignorant of the latent constructs and other factors that explain the data. The findings by Wieting and Kiela and subsequently Ding and Larson [9] who extended that work to further explore deep knowledge tracing, raise questions as to how useful these models are in educational contexts if they are not learning representations of deeper constructs; it is important to emphasize that these works did incorporate model training in the output layer of their compared models, so it is not the case that no training is required, only that the deeper layers of the networks may not learn composite features that align to latent factors in the data.

Prior work in applying recurrent models have explored how well such models are able to learn effective features in the context of detecting student affect and other learning behaviors [4]. In that work, the authors found that the use of expert-engineered features, developed in alignment to learning theory, led to higher predictive performance in a number of modeling tasks as compared to allowing a machine learning model to learn from the raw data logs used by the experts. The authors similarly identify an inherent difficulty within these networks to learn from the data.

This does not mean, however, that these models cannot still be useful in studying aspects of learning. Prior work has led to the development of sensor-free models of student affect [3] developed from student interaction logs paired with human-coded classroom observations [17,18]. Utilizing LSTM networks, these models have been successfully applied to study student affect even without the ability to interpret the learned representations within the model [5]; even if it is the case that many recurrent models are unable to learn deep representations of latent constructs, this does not mean that the estimates produced by these models cannot be useful to study learning (c.f. "discovery with models" [22].

3 Methodology[1]

Although there are many advantages in utilizing interpretable models to study learning, there are several practical benefits made possible if recurrent models are truly "ignorant" to deeper representations within data. As has been found in the prior works described in the previous section (i.e. [9,24]), untrained variants of recurrent models may perform comparably to similar trained models in several applications. If this finding holds in other contexts, these models may have increased potential for integration in a number of educational technologies and settings by significantly reducing training times or potentially even the amount of data needed to successfully fit such a model (needing to train only the output layer would be equivalent, in most cases, to training a linear or logistic regression model which traditionally requires fewer data samples to train).

In this work, we use student affect detection and knowledge tracing as two example cases of comparison for trained and untrained recurrent models on previously-published benchmark results ([4] and [27] for affect detection and knowledge tracing, respectively). While Ding and Larson [9] did explore applications of untrained models in the context of knowledge tracing, this work further expands upon this work by introducing two additional methods, Bag of Random Embeddings [24] and Echo State Networks [15,24], within educational contexts.

3.1 Affect and Knowledge Tracing Data

In this paper, we observe applications of these trained and untrained recurrent network models within two publicly-available datasets collected within ASSISTments [13]. ASSISTments is a free web-based learning platform used by primarily middle-school teachers and students for mathematics homework and classwork. Among a number of other features, the learning system allows teachers to assign traditional "complete all problems" assignments as well as mastery-based "skill builder" assignments. While working through problems, the system allows students to make multiple attempts to answer problems and offers supports in the form of on-demand hints and scaffolding problems. To support educational research, the system has also released a number of publicly-available datasets such as those utilized within this paper.

The first dataset observed in this work was released in [4], which was derived from several prior works focused on the development of sensor-free detectors of student affect (e.g. detectors that utilize only interaction logs without additional sensors such as video). ASSISTments data was used to develop affect detectors using expert-engineered features based on both theory and an iterative development process [18]. Additional works subsequently experimented with different features within a number of rule- and regression-based modeling methods [23] before recurrent deep learning methods were explored [3].

The dataset itself is comprised of student interaction logs paired with human-coded classroom observations of four states of student affect: engaged concentration, boredom, confusion and frustration. Following [4], the data exists in

[1] The code utilized by this work is made publicly available:https://osf.io/ubr2v/.

two forms, the first consisting of the 92 expert-engineered features used in prior works, and second consisting of the raw action-level logs that were used to build these features. While that work found that the use of expert-engineered features led to superior model performance as compared to the raw features, we explore both feature sets in this work utilizing untrained models to examine the performance benefits of training these models.

The second dataset observed in this work has been previously used to examine methods of knowledge tracing [27]. As described in Sect. 2, knowledge tracing is among the most widely studied problems in learner analytics, AI in education, and educational data mining communities. The original knowledge tracing (KT) model [7] and its bayesian implementation (BKT), attempt to model student latent knowledge using student performance metrics. The ASSISTments knowledge tracing dataset used in this work was made publicly available in [27] (specifically, the dataset refered to as "09–10 (c)" in that paper), after fixing several identified errors in the original version of that dataset used in Piech et al.'s original deep knowledge tracing paper [19]. This dataset is comprised of 275,459 math problems across 146 knowledge components answered by 4,217 students.

3.2 Leveraging Untrained Networks

This work explores the application of several untrained model structures. These model structures were applied across both the affect and knowledge tracing datasets, predicting the affect labels (as a multi-dimensional categorical outcome, as was done in prior works) and next problem correctness, respectively. As previously introduced, the terminology of "untrained" in the context of this work (in alignment with prior works [9,24]), refers to a partially-trained model. In most machine learning contexts, especially those observing deep learning approaches, models are typically trained by randomizing the initial values of a set of weights or coefficients that are then updated iteratively through an optimization procedure [16]. Considering deep learning models, this process is believed to help the model learn sets of features in lower layers of the network, with the final output layer (often functionally equivalent to a linear or logistic regression) then learning how to map those features to a set of outcomes. An "untrained" method effectively skips the optimization procedure, relying on the randomized weights to produce a large number of un-tuned features; in this process, a single regression model can be trained using these un-tuned features to map them to observed outcomes. This is an important distinction as this creates somewhat of a misnomer in that these methods still rely on some degree of training, but do not rely on training to "model" the data. These methods, as well as the application procedure, is described in this section.

Bag of Random Embeddings. The bag of random embeddings was the simplest untrained network. This method is used to simply project the time series data to a higher dimensional space. To create a bag of random embeddings for a time series of f features and t time-steps, the approach projects the time series

into a n dimensional embedding by first creating a t by f matrix, referred to as the time-series matrix, where each row in the matrix is the full set of features from one time-step. Next, the approach generates an f by n matrix full of random values, referred to as the projection matrix. The time-series matrix is then multiplied by the projection matrix, resulting in a t by n matrix, referred to as the embedding matrix. Finally, a pooling operation is applied across all the time-steps in the embedding matrix, resulting in a final n dimensional vectorized embedding of the initial time series.

Following the advice of [24], the random numbers of the projection matrix were initialized between $\frac{-1}{\sqrt{f}}$ and $\frac{1}{\sqrt{f}}$. To find the best bag of random embeddings, all combinations of an n of 1, 2, 4, 8, 16, 32, 64, 128, 256, and 512, and both max and mean pooling, for five random seeds each were used to project the time-series data before using a 5-fold cross-validated logistic regression to classify affect or predict next problem correctness, depending on the dataset. The average performance of every fold of every random seed for each combination of hyper-parameters was used to determine the best values.

Long Short-Term Memory Networks. The Long Short-Term Memory Network (LSTM) [11,14] is a common recurrent network structure for modeling time-series data. The LSTM network is a form of recurrent neural network that in addition to utilizing information from its past state, is designed to learn when to incorporate new information into its state and when to forget previous information. In this context, the value of the LSTM network is often viewed as being in its internal state structure which incorporates a type of memory that is designed to capture long- and short-term dependencies within the series (thus its name). Even without training, the state of the LSTM network, if complex enough, can capture useful, predictive information from the time-series in certain contexts [24]. To determine if an untrained LSTM network would be capable of predicting either affect or next problem correctness, an LSTM network was created with all combinations of zero through four hidden layers (i.e. additional fully connected layers on top of the LSTM layer), and 1, 2, 4, 8, 16, 32, 64, 128, 256, and 512 nodes per layer, including the output layer, for five random seeds. Each network's output layer was given to a 5-fold cross-validated logistic regression and used to classify affect or predict next problem correctness. The average performance of every fold of every random seed for each combination of hyper-parameters was used to determine the best combination of hyper-parameters.

Echo State Networks. Echo State networks are similar to recurrent networks in that they have connections from forward nodes to their predecessors, but these networks usually lack the formality of layers. Instead, an echo state network has a reservoir of nodes that have many connections to many other nodes in the reservoir. The input layer connects to any subset, or all of the nodes in the reservoir, and the output layer receives the output from the reservoir nodes. The weights in the reservoir are never trained, but the weights of the output layer are [15]. The echo state network is designed to exploit the properties of a recurrent

Table 1. Comparison of trained and untrained models applied to the affect dataset

Model	Features	Best model	AUC	Kappa
Untrained models				
LSTM Network	Raw	$n = 512$, 0 *added hidden*	0.661	0.098
Bag of Random Emb.	Raw	$n = 64$, *max pooling*	0.631	0.066
Echo State Network	Raw	$n = 512$, 1 *added hidden*	0.673	0.121
LSTM Network	Expert	$n = 512$, 1 *added hidden*	0.701	0.152
Bag of Random Emb.	Expert	$n = 64$, *mean pooling*	0.741	0.128
Echo State Network	Expert	$n = 512$, 0 *added hidden*	0.694	0.127
Trained models				
LSTM (Botelho et al., 2019)	Raw		0.695	0.041
LSTM (Botelho et al., 2019)	Expert		0.760	0.172

network's state similarly to how the previous section uses the state of an LSTM network. Within the untrained weights of the reservoir lies the state of the echo state network. This state is designed to capture the latent information of the time-series data presented to it and when the output layer is trained.

To determine if an echo state network would be capable of predicting either affect or next problem correctness, the output of each of the random LSTM networks from the previous section was combined with the intermediate output from every node in the network, essentially converting the LSTM network to an echo state network. The outputs from every node were again used to classify affect or predict next problem correctness in a logistic regression, which functions as the output layer of the echo state network. The average performance of every fold of every random seed for each combination of hyper-parameters was used to determine the best combination of hyper-parameters.

4 Results

The results of our applied untrained models are compared to the results generated from trained models as reported in prior works utilizing the same respective datasets used here. For consistency, these results are compared using the same metrics as have been used in comparison in prior works; in regard to the affect data, the AUC measure is calculated using the multi-class categorical evaluation method as used in previous works [12].

The results of the untrained models applied in this work in comparison to the trained models described in [4] are reported in Table 1. The highest-performing of each model type is compared to the reported results of the prior work across measures of AUC and Kappa (in alignment to that prior work). In this table, it can be seen that the trained LSTM utilizing expert-engineered features exhibits the highest model performance across both metrics. However, the untrained LSTM and Bag of Random Embedding models each perform comparably close in regard to AUC and Kappa; these even outperform the trained LSTM model utilizing the raw dataset.

Table 2. Comparison of trained and untrained knowledge tracing models.

Model	Best model	AUC
Untrained models		
LSTM Network	$n = 512$, 0 *added hidden*	0.706
Bag of Random Emb.	$n = 512$, *mean pooling*	0.692
Echo State Network	$n = 512$, 0 *added hidden*	0.725
LSTM (Ding & Larson, 2019)		0.730
Trained models		
DKT (LSTM; Xiong et al., 2016)		0.750
BKT (Xiong et al., 2016)		0.630

Similarly, the results of the untrained models applied in this work in comparison to previous results are reported in Table 2. In this table, we also compare our untrained model results to the untrained model applied in [9]. Here, the trained DKT model does exhibit the highest AUC performance, but the untrained LSTM as reported in [9] and Echo State Network applied in this study perform comparably well. What is perhaps particularly worth noting, is that all untrained recurrent models outperformed the BKT model.

5 Exploring Latent Feature Overlap

We have seen over the previous set of analyses that the untrained models perform comparably well to their trained counterparts. This raises several questions including what, if anything, is being learned within the hidden layer of these trained recurrent models (i.e. is there an overlap of latent features utilized by these models). In addressing our third research question, we conduct a final analysis to explore the latent features represented by trained and untrained models in detecting student affect.

In this analysis, we compare an LSTM-based model architecture as presented in [3] as a basis of comparison. We train this method using one LSTM layer consisting of 200 nodes feeding to a dense output layer of 4 nodes corresponding to the four affective states, similar to those previously described. We train this model and then extract the hidden layer from the network. Similarly, we generate five untrained counterparts using the same model architecture differing only in the number of nodes used in the hidden layer (using 200, 400, 600, 800, and 1000). We similarly extract the hidden layers of these models corresponding with each sample of the affect detection dataset.

We conduct an exploratory factor analysis (EFA) to identify latent constructs represented by each set of hidden features. EFA is a common dimensionality reduction method that identifies latent factors, or features, that exist as the linear combination of other features [1]. With this, we want to observe whether the factors that emerge from the trained model overlap, or are meaningfully

correlated, with the untrained model factors. If the trained model is not learning effectively from the data, we would expect that there would be a large overlap in factors when compared with the untrained models.

Table 3. Number of factors and overlap between untrained and trained models.

Model	EFA factors	N overlapping factors (Rho>0.6)
Trained LSTM (200)	31	—
Untrained LSTM (200)	35	5
Untrained LSTM (400)	50	1
Untrained LSTM (600)	74	5
Untrained LSTM (800)	91	4
Untrained LSTM (1000)	103	5

From our EFA, reported in Table 3, 31 features emerge from our trained model, with an increasing number of factors emerging from larger untrained dimensions (the number of factors were determined based on the number of factors with an eigenvalue greater than 1, following common practice). Using these features, we conducted a complete pair-wise comparison of untrained factors to trained model factors and computed a Spearman (Rho) ranked correlation for each pairing. We then simply counted the number of factor pairs that exhibited a Rho value greater than 0.6 as a measure of pseudo-overlapping feature sets. From the table, it can be seen that despite the increasing number of emerging factors, the number of overlapping factors remained relatively constant. This suggests that, while the untrained models constructed large feature sets, these were mostly uncorrelated with the trained model features.

6 Discussion

While it is surprising that the untrained models perform comparably well to their trained counterparts, the results of our analyses suggest that the trained models are learning effectively from the data; particularly from the EFA, we argue that the learned features are not simply random combinations of features due to the notable lack of overlap with the factors emerging from the untrained models. This lack of overlap is unexpected given the comparable model performance, suggesting that there are a small number of highly-predictive factors present.

In both affective and knowledge tracing contexts, the untrained models perform remarkably well, even outperforming other benchmarks (e.g. the trained LSTM using the raw data in Table 1 and the BKT model in Table 2). This work represents a step toward better understanding how deep learning models learn from given data. It is difficult to conclude that our findings will generalize to *all* recurrent models and applications, but the analyses conducted in this work in

conjunction with those presented in prior works [9,24] have found similar results across multiple contexts. It is the goal that this work will lead to further work to better understand knowledge representations within deep learning models to either better utilize them in various contexts, or to improve them so that they may exhibit higher utility for the study of learning.

Following the results reported in this paper, it is important to clarify and emphasize the contribution and potential impact of our findings. First, as the untrained models were found to be comparable to prior results across both applications observed in this paper, this finding aligns with prior research that suggests that the trained recurrent models may not be learning deep representations. However, the lack of overlap between factors emerging from the trained and untrained models suggests that the trained model is learning a distinctive set of latent factors related to affect. This finding supports the use of such models to both detect affect, but also to better study the latent structures that indicate affect and other learning constructs (i.e. these features are not simply randomly generated or encoded features). With that said, the untrained models may additionally provide utility. As the models perform well above chance and other simple baselines, the estimates produced by these models may still highly correlate with outcomes of interest and may be used to study learning.

Acknowledgements. We would like to thank NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, & R305A120125), GAANN (e.g., P200A180088 & P200A150306), EIR (U411B190024), ONR (N00014-18-1-2768) and Schmidt Futures.

References

1. Bandalos, D.L., Finney, S.J.: Factor analysis: exploratory and confirmatory. In: The Reviewer's Guide to Quantitative Methods in the Social Sciences, pp. 98–122. Routledge, London (2018)
2. Beck, J.E., Chang, K.: Identifiability: a fundamental problem of student modeling. In: Conati, C., McCoy, K., Paliouras, G. (eds.) UM 2007. LNCS (LNAI), vol. 4511, pp. 137–146. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73078-1_17
3. Botelho, A.F., Baker, R.S., Heffernan, N.T.: Improving sensor-free affect detection using deep learning. In: André, E., Baker, R., Hu, X., Rodrigo, M.M.T., du Boulay, B. (eds.) AIED 2017. LNCS (LNAI), vol. 10331, pp. 40–51. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61425-0_4
4. Botelho, A.F., Baker, R.S., Heffernan, N.T.: Machine-learned or expert-engineered features? exploring feature engineering methods in detectors of student behavior and affect. In: The 12th International Conference on Educational Data Mining (2019)
5. Botelho, A.F., Baker, R.S., Ocumpaugh, J., Heffernan, N.T.: Studying affect dynamics and chronometry using sensor-free detectors. Int. Educ. Data Min. Soc. (2018)

6. Chaplot, D.S., Rhim, E., Kim, J.: Predicting student attrition in MOOCs using sentiment analysis and neural networks. In: AIED Workshops, pp. 54–57 (2015)
7. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. User Model. User-Adap. Inter. **4**(4), 253–278 (1994)
8. Dey, R., Salem, F.M.: Gate-variants of gated recurrent unit (GRU) neural networks. In: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 1597–1600. IEEE (2017)
9. Ding, X., Larson, E.C.: Why deep knowledge tracing has less depth than anticipated. In: International Educational Data Mining Society (2019)
10. Doroudi, S., Brunskill, E.: The misidentified identifiability problem of Bayesian knowledge tracing. In: International Educational Data Mining Society (2017)
11. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM (1999)
12. Hand, D.J., Till, R.J.: A simple generalisation of the area under the roc curve for multiple class classification problems. Mach. Learn. **45**(2), 171–186 (2001)
13. Heffernan, N.T., Heffernan, C.L.: The assistments ecosystem: building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. Int. J. Artif. Intell. Educ. **24**(4), 470–497 (2014)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
15. Jaeger, H.: Echo state network. Scholarpedia **2**(9), 2330 (2007)
16. Le, Q.V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Ng, A.Y.: On optimization methods for deep learning. In: ICML (2011)
17. Ocumpaugh, J.: Baker Rodrigo Ocumpaugh monitoring protocol (BroMP) 2.0 technical and training manual. Technical report, Teachers College, Columbia University, New York, NY (2015)
18. Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., Heffernan, C.: Population validity for educational data mining models: a case study in affect detection. Br. J. Edu. Technol. **45**(3), 487–501 (2014)
19. Piech, C., et al.: Deep knowledge tracing. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, pp. 505–513 (2015)
20. Ritter, S., Anderson, J.R., Koedinger, K.R., Corbett, A.: Cognitive tutor: applied research in mathematics education. Psychonomic Bull. Rev. **14**(2), 249–255 (2007)
21. Rosé, C.P., et al.: Social factors that contribute to attrition in MOOCs. In: Proceedings of the 1st ACM Conference on Learning@ Scale Conference, pp. 197–198 (2014)
22. Siemens, G., Baker, R.S.d.: Learning analytics and educational data mining: towards communication and collaboration. In: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, pp. 252–254 (2012)
23. Wang, Y., Heffernan, N.T., Heffernan, C.: Towards better affect detectors: effect of missing skills, class features and common wrong answers. In: Proceedings of the 5th International Conference on Learning Analytics and Knowledge, pp. 31–35. ACM (2015)
24. Wieting, J., Kiela, D.: No training required: exploring random encoders for sentence classification. arXiv preprint arXiv:1901.10444 (2019)
25. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural Comput. **1**(2), 270–280 (1989)
26. Xing, W., Chen, X., Stein, J., Marcinkowski, M.: Temporal predication of dropouts in MOOCs: reaching the low hanging fruit through stacking generalization. Comput. Hum. Behav. **58**, 119–129 (2016)

27. Xiong, X., Zhao, S., Van Inwegen, E.G., Beck, J.E.: Going deeper with deep knowledge tracing. In: International Educational Data Mining Society (2016)
28. Yeung, C.K., Yeung, D.Y.: Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In: Proceedings of the 5th Annual ACM Conference on Learning at Scale, pp. 1–10 (2018)

Chapter 1.6

Using Auxiliary Data to Boost Precision in the Analysis of A/B Tests on an Online Educational Platform: New Data and New Results

Using Auxiliary Data to Boost Precision in the Analysis of A/B Tests on an Online Educational Platform: New Data and New Results

Adam C Sales
Worcester Polytechnic Institute
asales@wpi.edu

Ethan B Prihar
Worcester Polytechnic Institute
ebprihar@gmail.com

Johann Gagnon-Bartsch
University of Michigan
johanngb@umich.edu

Neil T Heffernan III
Worcester Polytechnic Institute
nth@wpi.edu

Randomized A/B tests within online learning platforms represent an exciting direction in learning sciences. With minimal assumptions, they allow causal effect estimation without confounding bias and exact statistical inference even in small samples. However, often experimental samples and/or treatment effects are small, A/B tests are under-powered, and effect estimates are overly imprecise. Recent methodological advances have shown that power and statistical precision can be substantially boosted by coupling design-based causal estimation to machine-learning models of rich log data from historical users who were not in the experiment. Estimates using these techniques remain unbiased and inference remains exact without any additional assumptions. This paper reviews those methods and applies them to a new dataset including over 250 randomized A/B comparisons conducted within ASSISTments, an online learning platform. We compare results across experiments using four novel deep-learning models of auxiliary data, and show that incorporating auxiliary data into causal estimates is roughly equivalent to increasing the sample size by 20% on average, or as much as 50-80% in some cases, relative to t-tests, and by about 10% on average, or as much as 30-50%, compared to cutting-edge machine learning unbiased estimates that use only data from the experiments. We show the gains can be even larger for estimating subgroup effects, that they hold even when the remnant is unrepresentative of the A/B test sample, and extend to post-stratification population effects estimators.
Keywords: A/B Tests Deep Learning Evaluation

1. Introduction

In randomized A/B tests on an online learning platform, students are randomized between different educational conditions or strategies, and their subsequent educational outcomes of interest are compared between different conditions. For instance, (Harrison et al., 2020) studied data from 2,152 middle- and high-school students whose teachers assigned a specific module—a "skill builder"—on the ASSISTments online tutoring platform (Heffernan and Heffernan, 2014). Prior to the students' work, the authors designed four different educational conditions, which

differed in how the numbers and symbols in arithmetic expressions were spaced. As students logged on to the platform, in the course of their usual schoolwork, they were each individually randomized to one of the four conditions, and completed their work under that condition. Subsequently, the authors of the study compared the average number of problems students in each condition had to work before achieving mastery, defined as answering three problems correct in a row. They found that students who were assigned the the "congruent" condition—in which the spacing between numbers corresponded to the order of operations—needed to work on roughly one fewer problem, on average, than students in the "incongruent" condition. This finding, and others reported in the paper, validated their previous scientific hypotheses regarding embodied cognition, the relationship between abstract learning and the arrangement of objects in physical (or virtual) space.

In general A/B tests have two significant advantages over observational study designs, which do not include randomization, and additional advantages over studies conducted in a lab. First, they are (famously) free of confounding bias—since students are randomly allocated between conditions, differences in outcomes must be due to either a causal effect of the randomized conditions or to random error, but not to baseline differences between students, observed or unobserved. Perhaps less famously, randomization forms a "reasoned basis for inference" (Fisher, 1935): the (known) probabilities of allocation of students between experimental conditions provide nearly all of the necessary justification for the unbiased estimation of causal effects, as well as standard errors, confidence intervals, and p-values. No other distributional assumptions or modeling assumptions are necessary. These properties allowed (Harrison et al., 2020) to estimate causal effects of spacing conditions, as well as to statistically rule out other alternative explanations.[1] Causal effect and standard error estimators that rely only on the experimental design are referred to as "design-based" (Schochet, 2015).

On the other hand, A/B tests can be hobbled by statistical imprecision. For instance, (Harrison et al., 2020) was unable to confirm or disconfirm one of their initial hypotheses, regarding differences in causal effects between subgroups of students, because the standard errors of the relevant estimates were too high. Unlike observational studies using data from online tutors, the sample size in A/B tests is necessarily limited to those students who worked on the relevant modules while the study was taking place. In contrast, an observational study can use data from all students who have ever worked on the relevant modules, including the (often large) number of students who worked on them before the onset of the study, and can sometimes use data from students who worked on similar modules as well. Analysis of A/B tests must discard data from these students, who were not randomized between treatment conditions and are subject to confounding. Unlike lab studies, A/B tests are subject to the haphazard unpredictability of real life, which only increases the statistical imprecision—even a sample as large as the 2,152 of (Harrison et al., 2020) may not be enough to answer some causal questions.

However, recent methodological innovations (Gagnon-Bartsch et al., 2021; Sales et al., 2018a) have argued that data from the "remnant" from an experiment—students who were not randomized between conditions, but for whom covariate and outcome data are available—need not be discarded, but can play a valuable role in causal estimation. In fact, researchers can use data from the remnant to decrease experimental standard errors without sacrificing the unbiased estimation and design-based inference that recommend A/B testing. The basic idea is to

[1] Actually the authors of that paper did make modeling assumptions in their analysis, but they could have conducted a non-parametric analysis.

first use the remnant data to train a machine learning model predicting outcomes as a function of covariates; then, use that fitted model to generate predicted outcomes for participants in the experiment. Finally, use those predictions as a covariate in a design-based covariate-adjusted causal estimator (Wu and Gagnon-Bartsch, 2018a; Aronow and Middleton, 2013; Wager et al., 2016a; Chernozhukov et al., 2018). Variants of the the method use the predictions from the remnant alongside other covariates to estimate causal effects.

These methods can help alleviate another weakness, shared by A/B tests and observational studies—the dependence of conclusions on statistical modeling choices. By observing outcome data prior to selecting and fitting statistical models, researchers (often inadvertently) choose models most favorable to their desired conclusions and undermine statistical objectivity and the logic of inference. Two proposed solutions to this issue are (1) to split the sample prior to data analysis, and use one part to choose a model and the second part to estimate effects (Heller et al., 2009) or (2) to rely on flexible non-parametric models that can be specified prior to data collection (Van der Laan and Rose, 2011). Design-based estimators incorporating remnant data rely on both these techniques: model-fitting in the remnant can be interactive and based on human judgement, without adversely affecting the objectivity or validity of statistical inference using the experimental sample. Design-based covariate adjustment often uses robust or non-parametric models.

This paper reviews design-based effect estimation from A/B tests, along with a set of design-based causal estimators that use remnant data (Section 2). Next (Section 3) we describe a new dataset which we used to test these methods: a collection of 68 multi-armed A/B tests run on the ASSISTments TestBed (Ostrow et al., 2016), which together include 277 different two-way comparisons, and 38,035 students. Alongside this experimental data, we collected log data for an additional 193,218 students who worked on similar skill builders in ASSISTments but did not participate in any of the 68 experiments—the remnant. The following section (Section 4) describes the Deep Learning model that we trained in the remnant to predict student outcomes as a function of prior log data.

The next four sections use that data and those models to address four research questions regarding the use of remnant data to assist in the analysis of A/B tests. The first research question (Section 5) regards the overall efficacy of our approach: to what extent might remnant data improve the precision of effect estimates from A/B tests? Does it ever harm precision, in practice? As part of this research question, we also investigated the roles various types of remnant data may play in the process. The second research question (Section 6) regards subgroup effects—treatment effects may be present for some groups of students but not others, or may differ between groups of students. However, breaking A/B test data into subsets further exacerbates sample size issues—is this something remnant data may help with? The third research question (Section 7) regards differences between the remnant and A/B testing data—in particular, what if the remnant is known to be drawn from a different population than the participants in A/B tests? Can it still be useful? To answer this question, we purposely constructed a new remnant that we believe is composed mostly of white and Asian males, and used it to analyze A/B testing data from primarily other demographic groups. The last research question (Section 8 asks if remnant data may be helpful in generalizing effects estimated from an A/B test to a wider population, even when subjects in the A/B test were not randomly drawn from that population.

Across the board, we find that estimates using the remnant are often substantially more precise than estimates that do not, and very rarely are much less precise. This holds for overall estimates, estimated subgroup effects, population average effects, and even when the remnant is

Name	Abbreviation	Explanation	Avg. Effect
RCT Set	RCT	Participants in the RCT, randomized between $Z = 0$ and $Z = 1$ conditions	$\overline{\tau}_{RCT}$
Population of interest	POP	The total population for which researchers wish to estimate effects	$\mathbb{E}_{POP}[\tau]$
Subgroup k	$G = k$	One of K disjoint subsets of POP or RCT	$\overline{\tau}_{G=k}$ or $\mathbb{E}_{POP}[\tau \mid G = k]$
Remnant	REM	Subjects with covariate ($\boldsymbol{x}$) and outcome (Y) data available, but who were randomized between conditions in the RCT	n/a

Table 1: Descriptions of sets of subjects described in the text, and associated causal estimands.

unrepresentative of the A/B test by construction. Our results give a much clearer picture of the potential impacts of using remnant data in design-based causal inference than was previously available.

2. Background

2.1. Framework: Different (Groups of) Users, Different (Average) Treatment Effects

For the method we are describing, it will be useful to define several different sets of subjects or users, summarized in Table 1 and Figure 1 (also see (Imbens, 2004)).

Consider an A/B test in which subjects $i = 1, \ldots, n$ are randomized between two conditions, which we denote as $Z_i = 0$ or $Z_i = 1$, with the goal of estimating effects of Z_i on an outcome Y_i. Call the set of randomized subjects i the "RCT set," or RCT. Typically, researchers running A/B tests are interested in the effect of Z on a broader population than RCT, such as all users of the system, or all users of a particular type; denote this target population as POP. For instance, students in a set of participating classrooms (RCT), working on a mastery-based homework assignment, may be randomized to either receive tutoring in the form of multi-step hints ($Z = 1$) or complete explanations of problem solutions ($Z = 0$), with the ultimate goal of estimating the effects of hints versus explanations on assignment completion (Y) for all users of the educational software (POP). (We focus on binary treatments for the sake of simplicity, though the methods and concepts we discuss extend easily to experiments with more than two conditions.)

Following (Neyman, 1923; Rubin, 1978) let $y_i(z)$, $z = 0, 1$ represent the outcome that subject i would experience if randomized to z—that is, if $Z_i = 0$, the observed outcome $Y_i = y_i(0)$, and if $Z_i = 1$ then $Y_i = y_i(1)$. Then, define the treatment effect for subject i as $\tau_i = y_i(1) - y_i(0)$, the difference between the outcome i would experience under condition 1 versus what they would experience under condition 0.

The challenge of causal inference is that for each i, only one of $y_i(0)$ or $y_i(1)$ is observed.

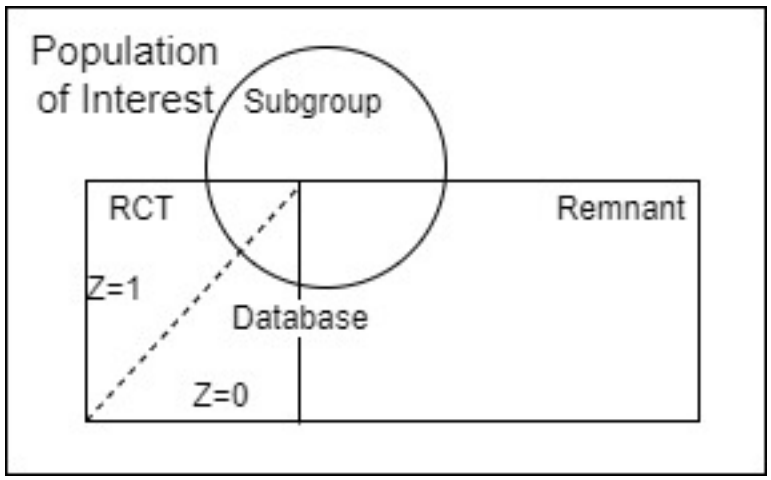


Figure 1: A Venn Diagram for the sets of subjects described in the text and Table 1

Hence, individual treatment effects τ_i cannot be estimated directly (at least, not precisely), but under some circumstances, average treatment effects can be estimated.

2.1.1. The Sample Average Treatment Effect

First, consider the sample average treatment effect,

$$\overline{\tau}_{RCT} = \sum_{i=1}^{n} \tau_i / n = \overline{y(1)} - \overline{y(0)}$$

where $\overline{y(1)}$ is the sample average of $y(1)$ over every subject in the RCT (whether $Z = 1$ or $Z = 0$), and $\overline{y(0)}$ is the sample average of $y(0)$. Hence, $\overline{\tau}_{RCT}$ is never observed, but can often be estimated. Claims about $\overline{\tau}_{RCT}$ pertain only to the participants in RCT, not (necessarily) about the treatment effect among other subjects.

2.1.2. The Population Average Treatment Effect

When researchers' interest goes beyond the average effect in RCT, and actually pertains to the larger population POP, then the estimand of interest is the population average effect, denoted $\mathbb{E}_{POP}[\tau]$[2]. If RCT is a random sample of POP, then there is little difference between estimating $\overline{\tau}_{RCT}$ and estimating $\mathbb{E}_{POP}[\tau]$. However, it is often the case that experimental participants are not representative of POP.

2.1.3. Subgroup Effects

If the population is partitioned into K subgroups—for instance, students with high or low performance prior to randomization, students in different school districts, or students in different demographic categories—then let $G_i \in 1, \ldots, K$ denote subject i's group membership, so that

[2]We use the notation of expected value $\mathbb{E}[\cdot]$ instead of sample average $\bar{\cdot}$ since it will often be mathematically convenient to think of POP as an infinite "super-population" from which subjects are drawn randomly (see, e.g. (Ding et al., 2017))

if $G_i = k$, then i is in the k^{th} subgroup. Then $\overline{\tau}_{G=k}$ and $\mathbb{E}_{POP}[\tau \mid G = k]$ are average treatment effects for members of the subgroup k in RCT or the population POP, respectively. In general, $\overline{\tau}_{RCT} = \sum_{k=1}^{K} p_k \overline{\tau}_{G=k}$ and $\mathbb{E}_{POP}[\tau] = \sum_{k=1}^{K} \pi_k \mathbb{E}_{POP}[\tau \mid G = k]$, where p_k and π_k are the proportions of RCT and POP, respectively, that belonging to group k.

2.2. Estimation and Types of Causal Bias

Estimation, and possibility of bias, depends on the causal estimand of interest, and can be due to bias in estimating $\overline{\tau}_{RCT}$, which we will call "internal" bias, bias in estimating $\mathbb{E}_{POP}[\tau]$ due to differences between subjects in the experiment and the population, which we will call "external bias," or a combination of the two. Our terminology mirrors the distinction between internal and external validity in, e.g. (McDermott, 2011).

2.2.1. Aside: Why do We Care about Statistical Bias?

While a good amount of early work in theoretical statistics focused on unbiased estimators, recent decades have seen increasing acknowledgement that unbiased estimators are often sub-optimal according to alternative estimation criteria and that a small amount of statistical bias may be a reasonable price to pay for improved statistical precision. That being the case, what accounts for our focus on unbiased estimation in this paper?

Although unbiasedness may not be an important goal for estimation in general, the concept of bias remains a useful formalization of some very important problems in estimation. For instance, the widely-known problems of estimating population quantities from unrepresentative or non-random samples or estimating causal effects from observational studies with unobserved confounding variables are both—in our opinion—most easily and clearly expressed in terms of bias. Extrapolation from unrepresentative samples and confounding can cause estimators to be inconsistent or inadmissible, and for confidence intervals and hypothesis tests to under-cover or over-reject, respectively. Our focus is on bias since we it to be the simplest and most straightforward way to formalize confounding and unrepresentative sampling.

2.2.2. Estimating $\overline{\tau}_{RCT}$ and Internal Bias

In a completely randomized experiment, the set of subjects with $Z = 1$ are a random sample of all the experimental participants, so $\overline{Y}_{Z=1} = (\sum_{i=1}^{n} Y_i Z_i)/(\sum_{i=1}^{n} Z_i)$, the average observed outcome for treated subjects, is an unbiased estimate of $\overline{y(1)}$, and likewise $\overline{Y}_{Z=0}$ is an unbiased estimator of $\overline{y(0)}$. (In general, let $\overline{X}_G$ be the sample mean of X for subjects for whom G is true $(\sum_{i=1}^{n} X_i \mathbf{1}\{G_i\})/(\sum_{i=1}^{n} \mathbf{1}\{G_i\})$, where $\mathbf{1}\{G_i\} = 1$ if G is true for i and 0 otherwise.) Then

$$\hat{\tau}^{DM} = \overline{Y}_{Z=1} - \overline{Y}_{Z=0}$$

the "difference-in-means" or "T-Test" estimator, is (internally) unbiased for $\overline{\tau}_{RCT}$. However, if treatment Z is not randomized—or if randomization is "broken" due attrition or some other irregularity—then $\hat{\tau}^{DM}$ will be biased due to confounding. Similarly, if treatment was randomized, but with different probabilities of treatment assignment for different subjects, $\hat{\tau}^{DM}$ may be a biased estimate of $\overline{\tau}_{RCT}$.

Even in a completely randomized experiment without other complications, some common effect estimators are biased for $\overline{\tau}_{RCT}$. For instance, say a vector of covariates $\boldsymbol{x}_i$ is observed for each subject. The ANCOVA estimator for $\overline{\tau}_{RCT}$, the estimated coefficient on Z from an ordinary

least squares regression of Y on Z and $\boldsymbol{x}$, is biased for $\overline{\tau}_{RCT}$. That said, when $\boldsymbol{x}$ has low dimension relative to n, the bias of the ANCOVA estimator is negligible (under suitible regularity conditions it decreases roughly with $1/n$; Freedman (2008)). However, if $\boldsymbol{x}$ has high dimension relative to n, or if a prediction algorithm other than OLS is used (improperly), the bias might be substantial.

2.3. Internally Unbiased Estimators using Auxiliary Data

2.3.1. The Remnant

While the difference-in-means estimator $\hat{\tau}^{DM}$ is unbiased for $\overline{\tau}_{RCT}$ in a completely randomized experiment, it may be imprecise, especially when the sample size is small. This problem may be exacerbated if a researcher is interested in estimating subgroup effects, either because of scientific interest in subgroups or for the sake of post-stratification as in (6). The reason is that $\overline{\tau}_{RCT}$ depends on unobserved counterfactual potential outcomes, $y_i(0)$ if $Z_i = 1$ and $y_i(1)$ if $Z_i = 0$, which must be imputed. $\hat{\tau}^{DM}$ relies on very rudimentary imputation strategy: the imputed $\hat{y}_i(0) = \overline{Y}_{Z=0}$ for all i such that $Z_i = 1$, and $\hat{y}_i(1) = \overline{Y}_{Z=1}$ for all i such that $Z_i = 0$. This strategy ignores all observed differences between subjects in the experiment, instead imputing one of the same two values for every subject.

In many cases, covariate and outcome data from an experiment are drawn from a larger database. For instance, educational field trials may use state longitudinal data systems to collect covariate data on student demographics and prior achievement, as well as on post-treatment standardized test scores, the outcome of interest, and medical trials may gather baseline and outcome data from databases of medical records. Most relevant for our purposes, analysis of A/B tests within online applications can access rich baseline data from users' logs prior to the onset of the experiment, and often draw outcome data from that same source. In these cases, researchers have the option of gathering additional auxiliary data—covariate and outcome data from users who were not part of the experiment. This includes historical data from before the onset of the experiment, as well as concurrent users who were not part of the experiment for some other reason. We refer to this set of users as the "remnant" from the experiment (Sales et al., 2018b) (rounding out the list of sets described in Table 1 and Figure 1).

2.3.2. A Naive Estimator using the Remnant

Say, for the sake of argument, that every subject in the remnant was in the $Z = 0$ condition; this will be the case if, for instance, $Z = 0$ represents a "business as usual" condition. Then, say researchers used the remnant to train an algorithm $\hat{y}^{REM}(0)(\boldsymbol{x}; \boldsymbol{\beta})$ predicting outcomes from covariates, with parameters $\boldsymbol{\beta}$, estimated with remnant data as $\hat{\boldsymbol{\beta}}$, and calculated imputations for participants in the experiment as $\hat{y}^r_i(0) = \hat{y}^{REM}(0)\left(\boldsymbol{x}; \hat{\boldsymbol{\beta}}\right)$. Then, for each experimental participant with $Z_i = 1$, estimate a individual treatment effect of $\hat{\tau}_i = Y_i - \hat{y}^r_i(0)$ and estimate $\bar{\tau}$ or $\mathbb{E}_{POP}[\tau]$ as $\hat{\tau}_{naive} = \overline{\hat{\tau}}_{Z=1}$.

The estimator $\hat{\tau}_{naive}$ has the potential to be much more precise than $\hat{\tau}^{DM}$, since it can account for observed baseline differences between experimental subjects, and use those differences to tailor its imputations to each individual subject. On the other hand, it has two serious disadvantages. First, the participants in the experiment are not necessarily drawn from the same population as the remnant, so there is no guarantee that the conditional distribution of $y(0)$ given $\boldsymbol{x}$ is the same in both groups. If the remnant is not representative of the experiment, so that $p(y(0)|\boldsymbol{x})$

differs between the two sets, $\hat{\tau}_{naive}$ may be biased for both $\bar{\tau}$ and $\mathbb{E}_{POP}[\tau]$. Second, even if the remnant is representative of the sample, there is typically no guarantee that the $\hat{y}^{REM}(0)(\cdot;\boldsymbol{\beta})$ is unbiased—in this case, the often erratic behavior of supervised learning algorithms in finite samples can also lead to bias.

2.3.3. Better Estimation using the Remnant

Both of these disadvantages can be corrected by relying on *both* randomization and supervised learning from the remnant. Specifically, the problems that cause internal bias in $\hat{\tau}_{naive}$ will also be present when comparing Y_i to $\hat{y}^r_i(0)$ for subjects in the control group, leading to the "remnant-based residualization" or "rebar" estimator (Sales et al., 2018b),

$$\hat{\tau}_{rebar} \equiv \hat{\tau}_{naive} - \overline{Y - \hat{y}^r}_{Z=0} = \overline{Y - \hat{y}^r}_{Z=1} - \overline{Y - \hat{y}^r}_{Z=0} = \hat{\tau}^{DM} - \overline{\hat{y}^r_i}_{Z=1} - \overline{\hat{y}^r_i}_{Z=0} \quad (1)$$

As (1) suggests, there are (at least) two ways to conceptualize the rebar estimator: first, it corrects the bias of $\hat{\tau}_{naive}$ by subtracting the analogous contrast in the $Z=0$ group, $\overline{Y - \hat{y}^r}_{Z=0}$, and second, it corrects for imprecision in $\hat{\tau}^{DM}$ by subtracting finite-sample difference in $\hat{y}^r$ between students in the two treatment conditions. $\hat{\tau}_{rebar}$ is precise if $\hat{y}^r$ are close to $y(0)$, on average, and is always unbiased for $\bar{\tau}$, due to the randomization of treatment assignment. Importantly, because the parameters β from the algorithm $\hat{y}^{REM}(0)(\cdot;\boldsymbol{\beta})$ are estimated using a separate sample, and $\boldsymbol{x}$ is fixed at baseline, $\hat{\tau}_{rebar}$ will be unbiased for $\bar{\tau}$ regardless of whether imputations $\hat{y}^r$ are themselves accurate or biased. This property is guaranteed by the randomization of treatment assignment.

The problem with $\hat{\tau}_{rebar}$ is that if the algorithm $\hat{y}^{REM}(0)(\cdot;\boldsymbol{\beta})$ performs poorly for subjects in RCT, then $\hat{\tau}_{rebar}$ will have high variance—sometimes even higher than $\hat{\tau}^{DM}$. A better solution is based on the fact that, in essence, $\hat{y}^r_i$ is itself a covariate, since it is a function of covariates $\boldsymbol{x}_i$ and parameters $\boldsymbol{\alpha}$ estimated using a separate sample. That being the case, it can be used as a covariate, perhaps along with others, in an existing covariate-adjusted estimator of $\overline{\tau}_{RCT}$.

For instance, a researcher could incorporate $\hat{y}^r$ into the ANCOVA estimator, the coefficient on Z of the regression of outcomes Y on an intercept, Z, $\hat{y}^r$, and, perhaps, a small number of other covariates. As discussed above, the ANCOVA estimator is consistent and only slightly biased in moderate to large samples.

Alternatively, Gagnon-Bartsch et al. (2021) suggests incorporating $\hat{y}^r$, perhaps alongside other covariates, into a flexible, internally-unbiased effect estimator that adjusts for baseline covariates using only RCT data (Wager et al., 2016b; Aronow and Middleton, 2013, for eg.). Like Gagnon-Bartsch et al. (2021), we will focus on the "LOOP" estimator (Wu and Gagnon-Bartsch, 2018b). Consider an A/B test with Bernoulli randomization—i.e. each subject is independently randomized—with $Pr(Z_i = 1) = p$ for all i. In this context, specify a 2nd algorithm $\widehat{y(z)}^{RCT}(\boldsymbol{x}, \hat{y}^r; \boldsymbol{\alpha})$ to impute potential outcomes $y(0)$ and $y(1)$ from remnant-based imputations $\hat{y}^r$, and (optionally) covariates $\boldsymbol{x}$, with parameters $\boldsymbol{\alpha}$. (Note that there are two separate algorithms predicting Y from $\boldsymbol{x}$: $\hat{y}^{REM}(0)(\boldsymbol{x};\boldsymbol{\beta})$ is fit using data from the remnant and produces imputations $\hat{y}^r_i$, while $\widehat{y(z)}^{RCT}(\boldsymbol{x}, \hat{y}^r; \boldsymbol{\alpha})$ is fit using RCT data.) For instance, (Gagnon-Bartsch et al., 2021) considers models

$$\widehat{y(z)}^{RCT}(\hat{y}^r\boldsymbol{\alpha})_{OLS} = \alpha^z_0 + \alpha^z_1\hat{y}^r \quad (2)$$

as well as a random forest predictor, $\widehat{y(z)}^{RCT}(\boldsymbol{x}, \hat{y}^r; \boldsymbol{\alpha})_{RF}$ incorporating covariates $\boldsymbol{x}$ alongside $\hat{y}^r$ as predictors, but ultimately recommends an ensemble of the two.

To estimate $\overline{\tau}_{RCT}$ without bias, it is essential that the predictions from $\widehat{y(z)}^{RCT}(\boldsymbol{x}, \hat{y}^r; \boldsymbol{\alpha})$ be statistically independent from the treatment assignment Z. The recommended estimators in (Gagnon-Bartsch et al., 2021) ensure that this is the case by using leave-one-out sample-splitting. For each subject in the experiment $i = 1, \ldots, n$, estimate $\boldsymbol{\alpha}$ as $\hat{\boldsymbol{\alpha}}_{(i)}$ using data from the other $n-1$ subjects, and impute missing potential outcomes using predictions $\widehat{y_i(0)}^{RCT}(\hat{y}^r, \boldsymbol{x}) = \widehat{y(0)}^{RCT}\left(\hat{y}^r_i, \boldsymbol{x}_i; \hat{\boldsymbol{\alpha}}_{(i)}\right)$ and $\widehat{y_i(1)}^{RCT}(\hat{y}^r, \boldsymbol{x}) = \widehat{y(1)}^{RCT}\left(\hat{y}^r_i, \boldsymbol{x}_i; \hat{\boldsymbol{\alpha}}_{(i)}\right)$.

Finally, estimate $\overline{\tau}_{RCT}$: first, let $\hat{m}_i = p\widehat{y_i(0)}^{RCT}(\hat{y}^r, \boldsymbol{x}) + (1-p)\widehat{y_i(1)}^{RCT}(\hat{y}^r, \boldsymbol{x})$, an imputation of i's expected counterfactual potential outcome. Then estimate $\overline{\tau}$ as:

$$\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x}) = \sum_{i:Z_i=1} \frac{Y_i - \hat{m}_i(\hat{y}^r, \boldsymbol{x})}{np} - \frac{1}{n} \sum_{i:Z_i=0} \frac{Y_i - \hat{m}_i(\hat{y}^r, \boldsymbol{x})}{n(1-p)} \tag{3}$$

where p, as above, is the probability of an individual participant being assigned to the $Z = 1$ condition. This is an inverse-probability-weighted estimate (also called Horvitz Thompson)—it is similar in form to $\hat{\tau}^{DM}$, except with the treatment and control sample sizes replaced with their expected values, np and $n(1-p)$. Aside from that difference, $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ with $\hat{m}_i = 0$ would correspond to $\hat{\tau}^{DM}$, and $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ with $\hat{m}_i = \hat{y}^r$ would be equivalent to $\hat{\tau}_{rebar}$. In general, $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ is much more flexible than either $\hat{\tau}^{DM}$ or $\hat{\tau}_{rebar}$, since it allows $\hat{y}^{REM}(')\,s$ role to vary depending on its prognostic value, and because it allows flexible incorporation of other baseline covariates.

Because parameters $\boldsymbol{\alpha}$ are estimated independently of i's outcome data, and $\boldsymbol{x}_i$ is fixed prior to treatment assignment, the sample splitting estimator is unbiased for the sample average treatment effect $\overline{\tau}$.

In (Gagnon-Bartsch et al., 2021), incorporating $\hat{y}^r_i$ into the LOOP estimator of (Wu and Gagnon-Bartsch, 2018b) in many cases led to substantial gains in precision compared to either $\hat{\tau}^{DM}$ or to the LOOP estimator with other covariates but not $\hat{y}^r_i$.

None of the methods considered here assumes that either imputation model, $\hat{y}^{REM}(0)(\cdot; \boldsymbol{\beta})$ or $\widehat{y(z)}^{RCT}(\boldsymbol{x}, \hat{y}^r; \boldsymbol{\alpha})$ is correct, unbiased, or consistent in any sense. Regardless of the quality of the imputation methods, randomization of treatment assignment ensures that effect estimates are unbiased.

2.3.4. Specific Estimators and Associated Terminology

Our two recommended estimators, which we term ReLOOP and ReLOOP+, combine ideas from $\hat{\tau}_{rebar}$ (1) and the leave-one-out covariate adjustment strategy LOOP (Wu and Gagnon-Bartsch, 2018b)—hence the name "ReLOOP." We will compare ReLOOP and ReLOOP+ to the T-Test estimator $\hat{\tau}^{DM}$, and a LOOP estimator that does not use remnant data. All told, we consider four different estimators:

- "T-Test": the difference-in-means estimator $\hat{\tau}^{DM}$, with no covariate adjustment
- "LOOP": $\hat{\tau}_{LOOP}(\boldsymbol{x})$ adjusts for covariates using a random forest imputation model fit to RCT data. It does not use any remnant data.
- "ReLOOP": $\hat{\tau}_{LOOP}(\hat{y}^r)$ adjusts only for $\hat{y}^r_i$, imputations from the model trained in the remnant, using LOOP with the OLS (2) RCT imputation model. It adjusts for no other covariates.

- "ReLOOP+": $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ uses an ensemble of OLS and random forests trained in RCT to adjust for both $\hat{y}^r_i$ and other covariates.

When an imputation model $\widehat{y(z)}^{RCT}(\boldsymbol{x}, \hat{y}^r; \boldsymbol{\alpha})$ is trained using RCT data, we refer to the associated covariate adjustment as "within-sample" or "within-RCT" adjustment. When an imputation model is trained in the remnant (i.e. $\hat{y}^{REM}(0)(\boldsymbol{x}; \boldsymbol{\beta})$), we refer to the associated covariate adjustment as "remnant-based." Comparing the two types of adjustment, within-sample adjustment as the advantage of hewing more closely to the actual RCT data on which it's trained, while remnant-based adjustment can rely on models fit using the remnant, which may boast a much larger sample size than the RCT. ReLOOP and ReLOOP+ make use of both types of adjustment.

2.3.5. Estimating Sampling Variance, p-values, and Confidence Intervals

The true sampling variances of $\hat{\tau}^{DM}$, $\hat{\tau}_{rebar}$, and $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$, as estimates of $\overline{\tau}_{RCT}$, depend on the correlation of $y(0)$ and $y(1)$, which is not identified without making further assumptions, since $y(0)$ and $y(1)$ are never observed simultaneously. However, it is possible to *conservatively* estimate the sampling variances of all three estimators. Specifically, for $z = 0, 1$, let

$$\hat{E}_z^2 = \frac{1}{n_z} \sum i : Z_i = 0 \left[\widehat{y(z)}^{RCT} \left(\hat{y}^r_i, \boldsymbol{x}_i; \hat{\boldsymbol{\alpha}}^z{}_{(i)} \right) - Y \right]^2$$

. Then estimate the sampling variance of $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ as:

$$\widehat{\mathbb{V}}\left(\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})\right) \frac{1}{n} \left[\frac{p}{1-p} \hat{E}_0^2 + \frac{1-p}{p} \hat{E}_1^2 + 2 \hat{E}_0 \hat{E}_1 \right]$$

As (Wu and Gagnon-Bartsch, 2018b) shows, $\mathbb{E}\left[\widehat{\mathbb{V}}\left(\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})\right)\right] \geq \mathbb{V}\left(\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})\right)$—that is, $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$'s estimated sampling variance is conservative in expectation.

Let the estimated standard error of $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ $\widehat{SE} = \widehat{\mathbb{V}}\left(\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})\right)^{1/2}$. The usual $1 - \alpha$ confidence interval has asymptotic coverage of at least $1 - \alpha$—i.e.

$$Pr(\bar{\tau} \in \hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x}) \pm ‡_{1-\alpha/2} SE) \to 1 - \tilde{\alpha} \geq 1 - \alpha$$

as $n \to \infty$, where $‡_{1-\alpha/2}$ is the $1 - \alpha/2$ quantile of the standard normal distribution. Similarly, a hypothesis test that rejects the null hypothesis of $\bar{\tau} = 0$ when $|\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})/SE| \geq ‡_{1-\alpha/2}$ will have a type-I error rate of at most α in large samples.

That is, the possible upward bias in these variance estimates will, if anything, cause confidence intervals to include the true parameter too often, or cause type-I error rates to be too low. While an unbiased sampling variance estimator would be preferable, conservative estimators are (arguably) the next best thing.

3. Data from 68 Educational A/B Tests

The remainder of the paper will discuss a set of illustrations and case-studies in using the ReLOOP and ReLOOP+ to estimate causal treatment effects from A/B tests run on an educational technology platform. This section describes the dataset—first the A/B tests themselves,

and then the remnant—and the following section describes $\hat{y}^{REM(0)}(\boldsymbol{x};\boldsymbol{\beta})$, the deep-learning model trained using remnant data. Subsequent sections will use data from the A/B tests and imputations from the model trained in the remnant to answer our research questions.

E-Trials is a platform that allows researchers to design educational experiments that will then be run within the ASSISTments online tutor. Education researchers can specify experimental conditions, including variation on how subject matter is portrayed, available hints, and feedback to students. Researchers also choose learning modules on which their experiments run. When teachers subsequently assign these modules to their students, the students are randomized between the conditions. After the period of the experiment has ended, the researcher is provided with a dataset, including classroom and student identifiers, log data from during the experiment, and outcome data such as which students completed the assignment and how many problems they worked. Students are randomized between conditions independently, one at a time; when there are only two conditions, this is Bernoulli randomization.

We gathered data from a set of 84 A/B tests run on E-Trials. Since our interest here is primarily methodological, with the goal of reducing standard errors, we focus on estimated standard errors as opposed to treatment effects. Our analyses will focus on assignment completion as a binary outcome.

We also gathered a set of nine student-level aggregated predictors, to be used for within-RCT covariate adjustment. These were the numbers of skill builders (mastery-based moduals in ASSISTments) and problems sets each student began and completed, as well as each student's prior median first response time when working ASSISTments problems, median time on task, overall correctness, completed problem count, and average attempt count.

Several experiments included multiple conditions, rather than only treatment and control. We assume that primary interest in these experiments focuses on head-to-head comparisons between conditions, and, as such, we analyze all unique pairs of conditions within randomized experiments separately. All in all, this includes 383 pairs. However, not every pair was amenable to analysis. Six pairwise contrasts were dropped because the outcome variance in one or both of the conditions was zero. Further exclusions were motivated by two factors: first, the LOOP estimator (which also underlies the ReLOOP and ReLOOP+ estimators) presumes that $p = P(Z_i)$ is known. In theory, $p = 1/2$ should hold in all pairwise comparisons. However, there were strong indications that that some subsets of experiments used a different randomization scheme that we did not have access to. Instead, we estimated p-values testing the null hypothesis that $p = 1/2$ for each comparison we considered; we dropped contrasts in which the p-value testing $p = 1/2$ was > 0.1. Secondly, there were some contrasts which included extremely small samples, with the smallest being $n = 16$. The LOOP estimators rely on OLS regression or more complex models, and cannot be expected to perform well when sample sizes are so small. In the main analyses, we dropped experiments in which the sample size in either condition was less than $5(k + 2) + 1$, where $k = 9$ is the number of predictors, which would allow for at least 5 observations per predictor in any model. In the subgroup analyses of Section 6, we analyzed subgroups with smaller sample sizes.

These exclusions left a total of 227 randomized contrasts—pairs of treatment conditions between which students were randomly assigned—drawn from 68 separate A/B tests.

3.1. Data Collection

The data was collected from ASSISTments in two sets, remnant data, and experiment data. Remnant data was used to train the imputation models, and experiment data was used to determine the outcomes of each experiment using the imputation models and ReLOOP. The skill builders started by the students in the remnant data were not the same skill builders as the experimental skill builders in the experiment data, nor is there any overlap in students between the two datasets. **No information from the students or skill builders in the experiment data was in the remnant data used to train the imputation models**.

For both the remnant and experiment data, the same information was collected. For each instance of a student starting a skill builder for the first time, data on whether they completed the skill builder, and if so, how many problems they had to complete before mastering the material was collected. The imputation models, discussed more in section 4 were trained to predict these two dependent measures. The data used to predict these dependent measures was aggregated from all of the previous work done by the student. Three different sets of data were collected for each sample in the datasets: prior student statistics, prior assignment statistics, and prior daily actions. Prior student statistics included the past performance of each student, for example, their prior percent correct, prior time on task, and prior assignment completion percentage. Prior assignment statistics were aggregated for each assignment the student started prior to the skill builder. Prior assignment statistics included things like the skill builders' unique identifier (or in the remnant data, the ID of the experimental version of a skill builder, if it existed), how many problems had to be completed in the assignment, students' percent correct on the assignment, and how many separate sessions students' used to complete the assignment. Prior daily actions contained the total number of times students performed each possible action in the ASSISTments Tutor for each day prior to the day they started the skill builder. The possible actions included things like starting a problem, completing an assignment, answering a problem, and requesting support. Complete lists of features included in prior student, assignment, and daily action datasets are included in Tables 5, 6, and 7 in the appendix. 193,218 sets of prior statistics on students, 837,409 sets of statistics on prior assignments, and 695,869 days of students' actions were aggregated for the remnant data, and 113,963 sets of prior statistics on students, 2,663,421 sets of statistics on prior assignments, and 926,486 days of students' actions were aggregated for the experiment data. The full dataset used in this work can be found at https://osf.io/k8ph9/?view_only=ca7495965ba047e5a9a478aaf4f3779e.

4. Remnant-Trained Imputation Models

4.1. Model Design

Each of the three types of data in the remnant dataset were used to predict both skill builder completion and number of problems completed for mastery. For each type of data: prior student statistics, prior assignment statistics, and prior daily actions, a separate neural network was trained. Additionally, a fourth neural network was trained using a combination of the previous three models. The prior student statistics model, shown in Figure 2 in red was a simple feed forward network with a single hidden layer of nodes using sigmoid activation and dropout. Both the prior assignment statistics model and the prior daily actions model, shown in Figure 2 in blue and yellow respectively, were recurrent neural networks with a single hidden layer of LSTM nodes (Gers et al., 2000) with both layer-to-layer and recurrent dropout. The prior

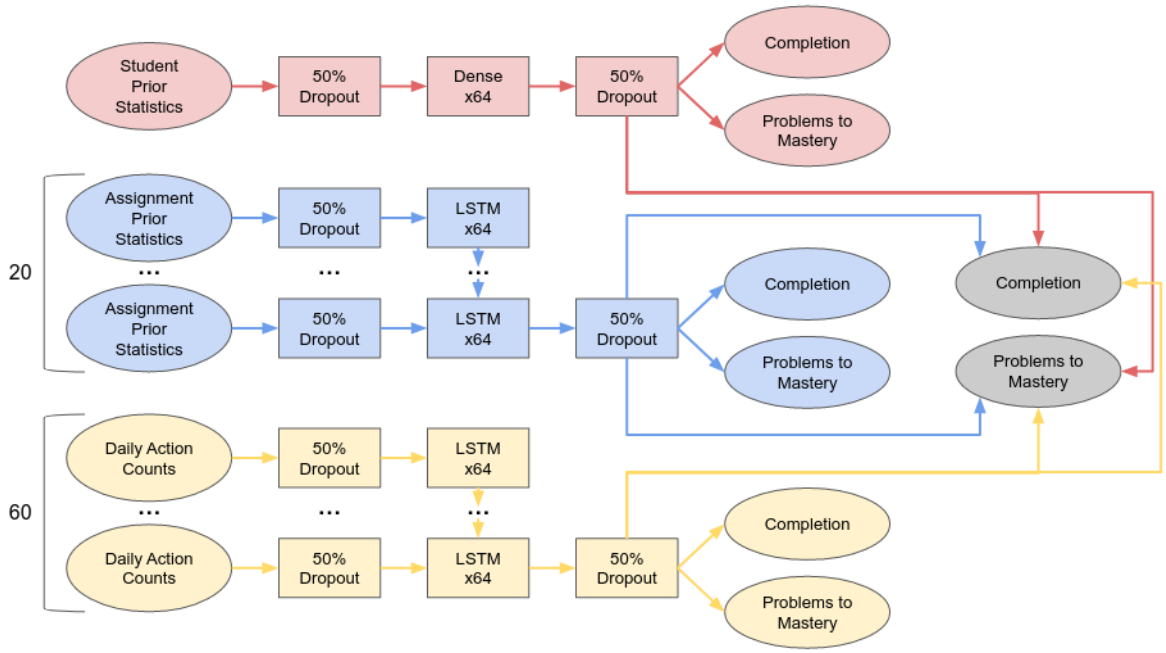


Figure 2: All four of the imputation models in one. The red model predicts performance using only prior statistics of the student, the blue model uses statistics on the last 20 assignments completed by the student to predict performance, and the yellow model uses the last 60 days of actions the student took in the tutor. The combined model, shown in grey, uses all three models to predict performance.

assignment statistics model used the last 20 started assignments as input, and the prior daily actions model used the last 60 days of actions as input. The combined model in Figure 2 takes the three models above and couples their predictions, such that the prediction is a function of all three models weights and the loss same loss is backpropigated through each model during training.

4.2. Model Training

To select the best model hyperparameters and to measure the quality of each imputation model, 5-fold cross validation was used to train and calculate various metrics for each model. For all training, the ADAM method (Kingma and Ba, 2014) was used during backpropigation, binary cross-entropy loss was used for predicting completion, and mean squared error loss was used for problems to mastery. The total loss for each model was the sum of the two individual losses. Because mean squared error and binary cross-entropy have different scales, a gain of 16 was applied to the binary cross-entropy loss, which brought the loss into the same range as the mean squared error loss for this particular dataset. Table 2 shows various metrics of the models' quality. Interestingly, even though all the models are bad at predicting problems to mastery, removing problems to mastery from the loss function reduced the models ability to predict completion.

Based on Table 2, statistics on prior assignments was the most predictive of students' assignment performance, followed by the students' overall prior performance statistics, and then

Table 2: Metrics Calculated from 5-Fold Cross Validation for each Model

Metric	Prior Student Statistics	Prior Assignment Statistics	Prior Daily Action Counts	Combined
Completion AUC	0.743	0.755	0.658	**0.770**
Completion Accuracy	0.761	0.767	0.743	**0.774**
Completion r^2	0.143	0.161	0.045	**0.184**
# of Problems MSE	8.489	8.505	8.719	**8.363**
# of Problems r^2	0.033	0.032	0.007	**0.048**

their daily action history, which was the least predictive of their performance on their next assignment. Combining these datasets together led to predictions of a higher quality than any individual dataset could achieve.

5. Research Question 1: Can Imputations from Remnant-Trained Models Improve Standard Errors for Average Effects?

To gauge the potential of remnant-based imputations to improve the precision of impact estimates, we compared estimated sampling variances from the four different treatment effect estimators listed in Section 2.3.4: T-Tests ($\hat{\tau}^{DM}$), which includes no covariate adjustment; LOOP, which uses random forests for within-sample covariate adjustment using only the 9 student-aggregated covariates in 3 but not the remnant; ReLOOP, which uses remnant-based imputations $\hat{y}_i^r$ in a within-sample OLS adjustment model; and ReLOOP+, which uses an ensemble algorithm with to adjust for both $\hat{y}_i^r$ and the nine student-aggregate covariates in LOOP. In this analysis, we used the "combined" model, including all available remnant data, to generate remnant-based imputations $\hat{y}_i^r$. We used these four estimators to estimate effects in each of the 227 randomized contrasts described above.

Figure 3 shows the ratios of estimated sampling variances from the four estimators. Since sampling variance scales as $1/n$, ratios of sampling variances can be thought of as "sample size multipliers"—that is, decreasing the variance by a factor of q is analogous to increasing the sample size by the same factor. The results in Figure 3 were previously reported in a conference poster (Sales et al., 2022).

The panel on the left of 3 compares $\hat{\tau}_{LOOP}(\hat{y}^r)$ to $\hat{\tau}^{DM}$, the t-test estimator. In nearly every case the estimator using remnant date substantially outperformed the t-test estimator. In the majority of cases, including remnant-based predictions was roughly equivalent to increasing the sample by between 15 and 60%. The middle panel of Figure 3 compares $\hat{\tau}^{DM}$ to $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$. Here the results are slightly more impressive than those of the left panel—the median improvement is equivalent to increasing the sample size by about 20%, and in the best case the improvement is equivalent to an 80% increase in sample size.

The rightmost panel of Figure 3 compares $\hat{\tau}_{LOOP}(\boldsymbol{x})$, which uses leave-one-out sample splitting and a random forest to adjust for covariates—but does not use the remnant—to $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ which does. In this case we see more modest gains, which is to be expected, since $\hat{\tau}_{LOOP}(\boldsymbol{x})$ can accomplish a good deal of covariate adjustment using only experimental data. Nevertheless, the contribution of the remnant is still significant—in roughly half of cases, including data from the remnant was equivalent to increasing the sample size by about 10–20%, and in a handful of

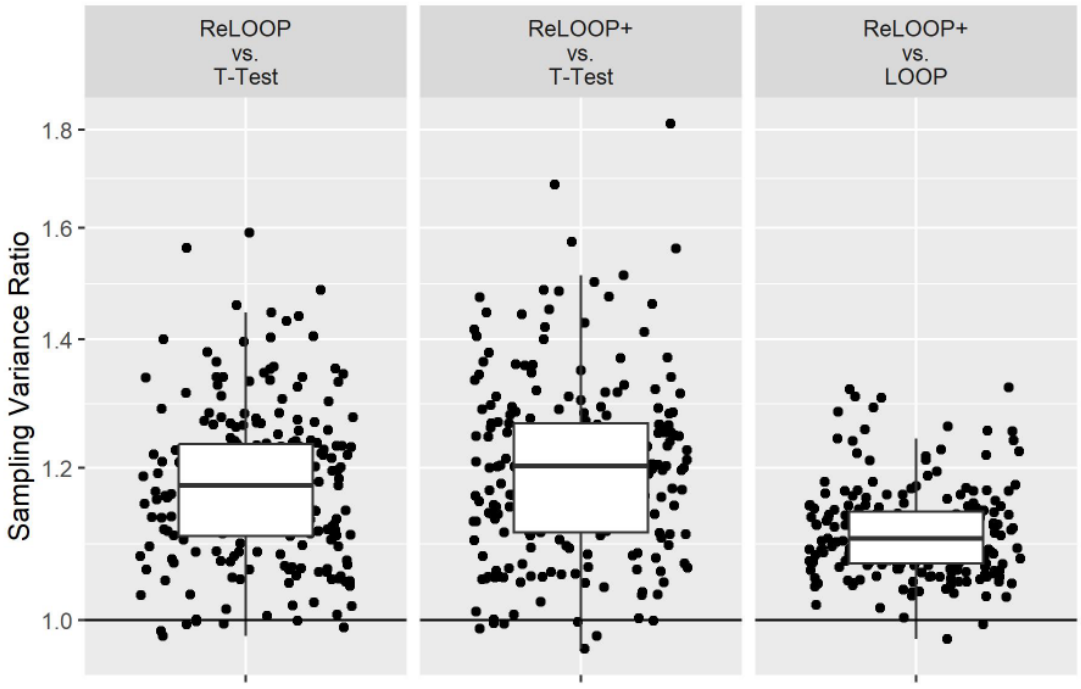


Figure 3: Boxplots and jittered scatter plots of the ratios of estimated sampling variances of $\hat{\tau}^{DM}$ (i.e. "T-Test," which includes no covariate adjustment), $\hat{\tau}_{LOOP}(\boldsymbol{x})$ ("LOOP", which adjusts for covariates within sample, but does not use the remnant), $\hat{\tau}_{LOOP}(\hat{y}^r)$ ("ReLOOP," which adjusts for remnant-based imputations but not within-sample covariates), and $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ ("ReLOOP+," which adjusts for both within-sample covariates and remnant-based imputations) 227 randomized contrasts. The Y-axis is on a logarithmic scale, so that, say, doubling the sample size appears as the same magnitude of an effect as halving the sample size.

cases the improvement was closer to 30%.

In summary, covariate adjustment can lead to substantial gains in precision, with the greatest improvement resulting from adjustment using both within-sample aggregated covariates and remnant-based imputations. In particular, estimators including remnant based imputations consistently outperformed those that used only within-sample covariate adjustment.

5.1. Did the Remnant Help Us Discover any Effects?

Researchers naturally want to know if our claim to increase the power of A/B tests to detect effects actually lead, in practice, to more effects detected. In other words, did covariate adjustment lead to any p-values dipping below the $\alpha = 0.05$ threshold? Counting significant p-values is a problematic approach to gauging the success of our method, since it depends on the size of the true effects. In particular, if the true $\overline{\tau}_{RCT}$ is equal to 0, then a p-value less than 0.05 would be a type-I error, but if the $\overline{\tau}_{RCT}$ is not equal to 0, a p-value less than 0.05 would be a true discovery. Since the ground truth is unknown, we cannot know if which one is the case.

	T-Test	LOOP	ReLOOP	ReLOOP+
Unadjusted	38	41	41	41
Benjamini-Hochberg	3	8	8	10
Benjamini-Yekutieli	2	2	2	2

Table 3: The number of p-values less than $\alpha = 0.05$ using each of the four estimators. The table counts significant p-values unadjusted for multiple comparisons, and adjusted with the Benjamini-Hochberg and Benjamini-Yekutieli procedures.

Nevertheless, we will press on. Table 3 gives the count of significant p-values using each of the four estimates. The first row gives a count of unadjusted p-values; if each pairwise comparison were considered in isolation, these would be the relevant counts. A researcher using T-Tests would report discoveries in 38 cases, while researchers using covariate adjustment via LOOP, ReLOOP, or ReLOOP+ would report an additional 3 discoveries. However, since there were 227 total hypothesis tests, even if the null hypothesis were true in every case we would expect around 11 significant p-values; in other words, since we are considering the p-values as a group a multiplicity adjustment is in order. We considered two adjustment methods, both designed to limit the "false discovery rate"—the proportion of the discoveries that are, in fact, type-I errors—to 5%. The second row of Table 3 counts p-values adjusted with the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995). This procedure is guaranteed to control the false discovery rate only if the tests are independent[3]. The pairwise comparisons we consider are not independent, since each A/B test may have contributed several pairwise comparisons, which share data. After Benjamini-Hochberg adjustment, a researcher using T-Tests would only discover 3 effects, while researchers using LOOP and ReLOOP would discover 8; combining within-sample and remnant-based adjustment with ReLOOP+ would lead to two additional discoveries, or 10 total. The third row of the table counts significant p-values adjusted by the more conservative Benjamini-Yekutieli procedure (Benjamini and Yekutieli, 2001), which controls the false discovery rate even under arbitrary dependence of tests. Researchers using any of the

[3]There are some types of dependence which are OK, too, but they are difficult to describe, much less to verify.

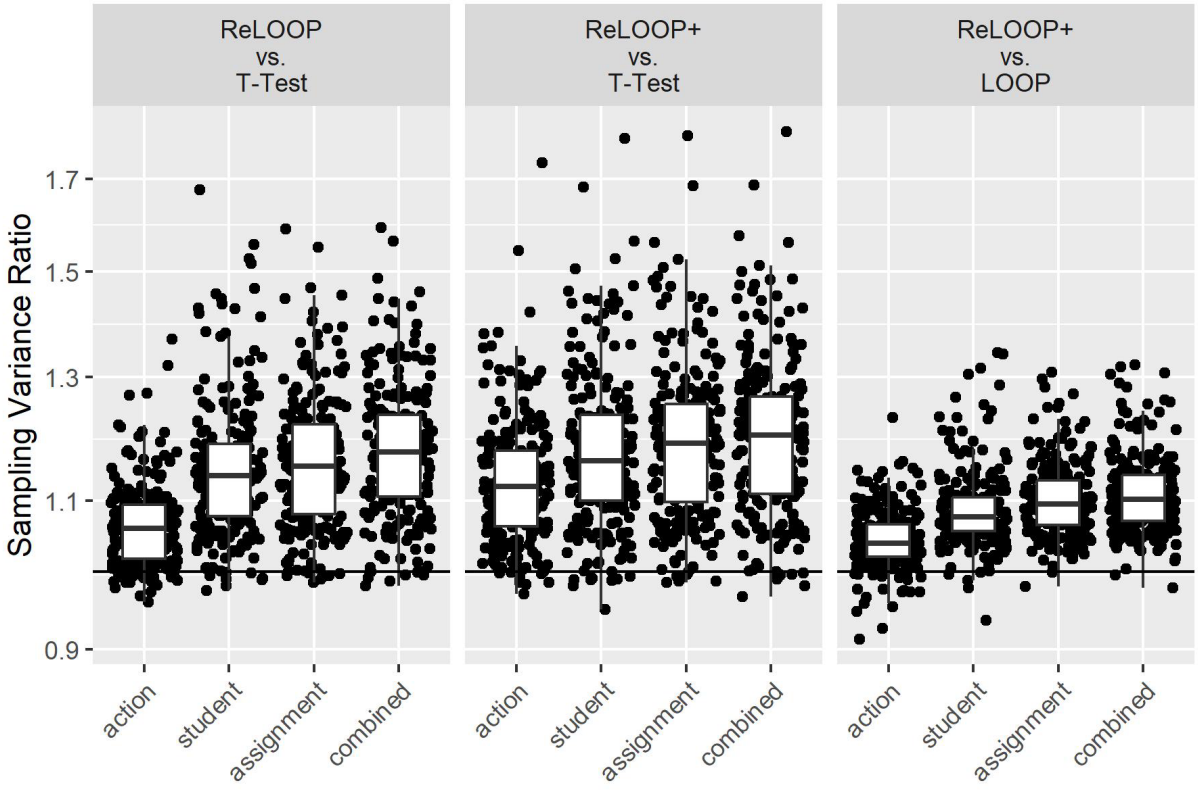


Figure 4: Boxplots and jittered scatter plots of the ratios of estimated sampling variances of $\hat{\tau}^{DM}$ (i.e. "T-Test," which includes no covariate adjustment), $\hat{\tau}_{LOOP}(\boldsymbol{x})$ ("LOOP", which adjusts for covariates within sample, but does not use the remnant), $\hat{\tau}_{LOOP}(\hat{y}^r)$ ("ReLOOP," which adjusts for remnant-based imputations but not within-sample covariates), and $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ ("ReLOOP+," which adjusts for both within-sample covariates and remnant-based imputations) 227 randomized contrasts. The Y-axis is on a logarithmic scale.

four estimators we've considered and adjusting with the Benjamini-Yekutieli procedure would all reject 2 null hypotheses among the 227 possibilities.

5.2. Which Remnant Data Helps the Most?

Figure 4 expands on figure 3 by contrasting the performance of ReLOOP and ReLOOP+, relative to T-Tests and LOOP, using remnant-based imputation models trained using different types of remnant data. As described above, the "action" model uses data on each student's daily actions in ASSISTments leading up to the A/B test, the "student" model used student-aggregated performance metrics prior to the beginning of the A/B test, and the "assignment" model used student performance metrics on previous assignments or skill-builders each student had worked on. Finally, the "combined" model—also shown above, in Figure 3—was an ensemble of the action, student, and assignment models. By examining the performance of each separate model, we can get a sense of the relative contribution of each type of remnant data to ReLOOP or ReLOOP+'s performance.

Comparing across models fit in the remnant, the action-level model performed the worst, while the combined model was responsible for the greatest decrease in sampling variance. Inter-

estingly, the assignment-level model performed nearly as well as the combined model, suggesting that action- and student-level data did not contribute substantially. This pattern is consistent across the three different comparisons shown, comparing ReLOOP and ReLOOP+ to T-Tests, and comparing ReLOOP+ to LOOP.

6. Research Question 2: ReLOOP for Subgroup Effects

To judge ReLOOP's potential for improving (or worsen) precision in subgroup effect estimates, we created subgroups using each of the 9 student-aggregated covariates available for each of the randomized comparisons we considered. Specifically, we first pooled each of the 9 covariates $\boldsymbol{x}_k$, $k = 1, \ldots, 9$ across all of the 227 pairwise comparisons, and calculated the 1/3 and 2/3 quantiles, $q_{1/3}(\boldsymbol{x}_k)$ and $q_{2/3}(\boldsymbol{x}_k)$. Then, for each contrast and each covariate x, we estimated effects for students with "low" ($x_{ik} < q_{1/3}(\boldsymbol{x}_k)$) or "high" ($x_{ik} > q_{2/3}(\boldsymbol{x}_k)$. Finally, using each of the four estimators described in the previous section, for each pairwise contrast and for each covariate, we estimated two effects: one for low students and one for high.

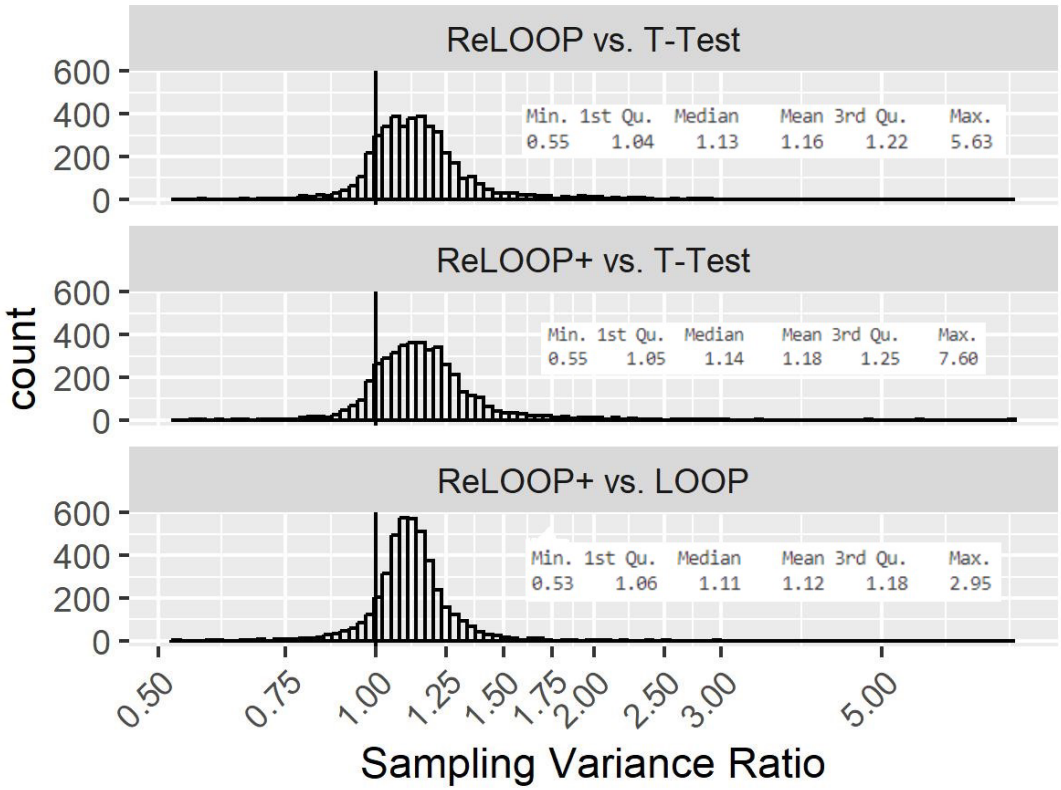
In addition to the 9 within-sample covariates, we also looked for effects in subgroups defined by the remnant-based imputations themselves—that is, students with a high or low probability of completing their assignment, using the remnant-based model.

All told, this should have resulted in $227 \times 10 \times 2 = 4,540$ estimates for each of the four estimators. In practice, we did not estimate effects if either treatment arm within a subgroup had fewer than 10 subjects, which excluded 210 of these comparisons, and we encountered other estimation problems (such as the lack of variance in outcomes) in 19 others, leaving a total of 4,311 random comparisons to consider. Now, these 4,311 comparisons are by no means independent—they represent different ways to slice the data from the original 68 A/B tests. Nevertheless, by considering them all we may be able to discern some patterns in ReLOOP's effectiveness in improving precision.

First, though, Figure 6 shows sampling variance ratios pooled across all A/B tests, pairwise comparisons, and subgroups. For the first time, we see some cases of covariate adjustment substantially harming the precision of effect estimates—ReLOOP gave larger standard errors than T-Tests in about 14% of cases, ReLOOP+ gave larger standard errors than T-Tests in around 12.5% of cases and ReLOOP+ gave larger standard errors than LOOP in about 11% of cases. In the vast majority of these cases the effect was comparable to decreasing the sample size by less than 10%, but about 3% of cases using ReLOOP was equivalent to decreasing the sample size by 10% or more, and in a handful of cases the decrease was even larger, up to about 50%.

Still, in the majority of cases remnant-based covariate adjustment improved the precision of impact estimates, sometimes by dramatic amounts. For all three comparisons shown in the figure, the median sampling variance ratio was greater than 1.1, meaning that ReLOOP or ReLOOP+ was equivalent to increasing the sample size by more than 10% at least half the time. Much more dramatic improvements were also common: in 25% of cases, ReLOOP outperformed the T-Test by 22% or more, ReLOOP+ outperformed the T-Test by 25% or more, and ReLOOP+ outperformed LOOP by at least 18%. In some extreme cases the improvement due to ReLOOP was equivalent to doubling or tripling the sample size, and in a few cases it was equivalent to multiplying the sample size by 5 or even 7.

Echoing the analysis in Section 5.1, Table 4 shows the number of discoveries—i.e. $p < 0.05$—a researcher would make using each of the three estimators. If p-values are not adjusted for multiple comparisons, a researcher using ReLOOP or ReLOOP+ would reject 10 more null



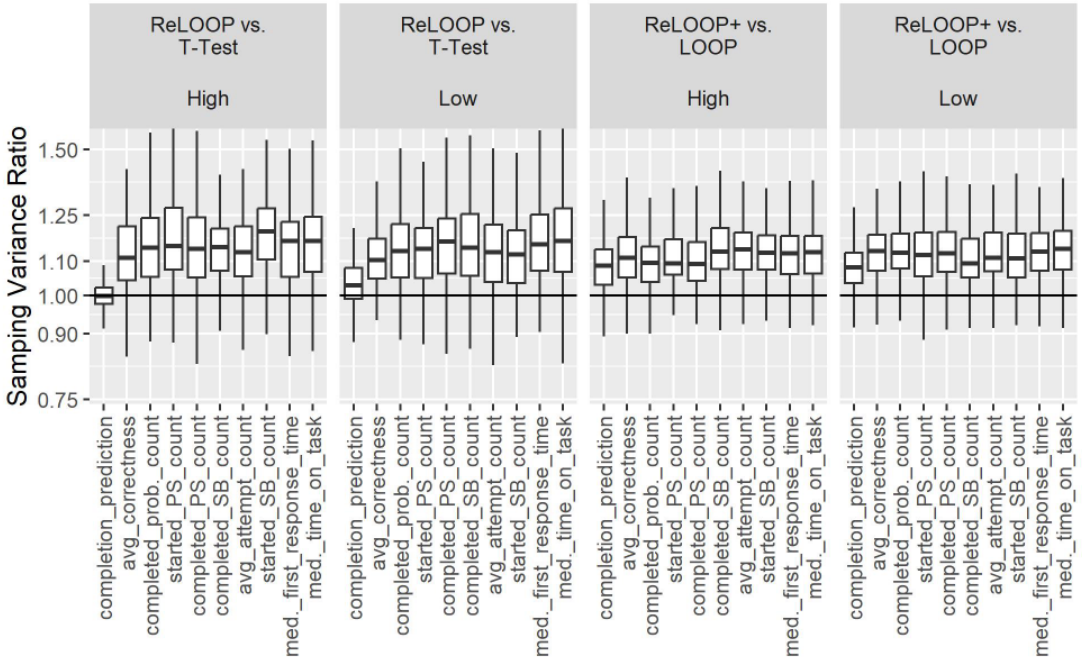
Histograms of the ratios of sampling variances of $\hat{\tau}^{DM}$ (T-Tests), $\hat{\tau}_{LOOP}(\boldsymbol{x})$ (LOOP), $\hat{\tau}_{LOOP}(\hat{y}^r)$ (ReLOOP), and $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ (ReLOOP+) for 4,311 estimated subgroup effects. Sample statistics of the distributions of ratios are also shown. The X-axis is on logarithmic scale.

hypotheses than a researcher using LOOP, and 43 more than a researcher using T-Tests. If p-values are adjusted with the Bejamini-Hochberg procedure, a researcher using T-Tests would fail to reject every one of the 4,311 null hypotheses, while one using LOOP would reject 23, one using ReLOOP would reject 22, and a researcher using ReLOOP+ would reject 28, ensuring tenure and grant funding. After adjusting with the Benjamini-Yekutieli procedure, only researchers using ReLOOP or ReLOOP+ would reject any hypotheses—7 in both cases.

	T-Test	LOOP	ReLOOP	ReLOOP+
Unadjusted	370	403	413	413
Benjamini-Hochberg	0	23	22	28
Benjamini-Yekutieli	0	0	7	7

Table 4: The number of p-values less than $\alpha = 0.05$ using each of the four estimators. The table counts significant p-values unadjusted for multiple comparisons, and adjusted with the Benjamini-Hochberg and Benjamini-Yekutieli procedures.

The following two subsections dig deeper into these varying effects by looking at subgroup effects broken down by subgroup, and as a function of sample size.



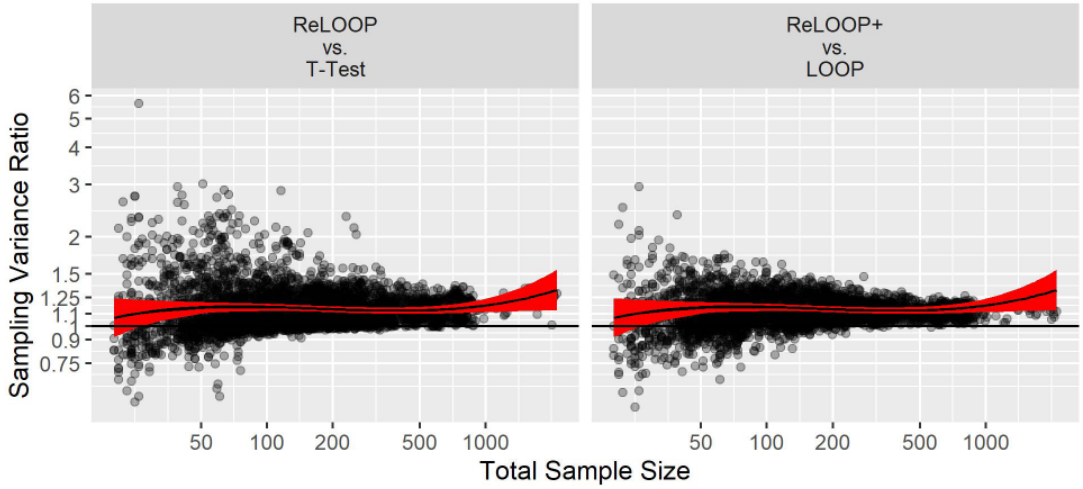
Boxplots of the ratios of sampling variances of $\hat{\tau}^{DM}$ (T-Tests), $\hat{\tau}_{LOOP}(\boldsymbol{x})$ (LOOP), $\hat{\tau}_{LOOP}(\hat{y}^r)$ (ReLOOP), and $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ (ReLOOP+) for each subgroup considered. Outliers are omitted. The Y-axis is on a logarithmic scale.

6.1. Subgroup Effect Standard Errors by Covariate

Figure 6.1 shows boxplots of sampling variance ratios comparing ReLOOP to T-Tests and ReLOOP+ to LOOP for each subgroup we considered. A few features are apparent. First, ReLOOP performs no better than T-Tests for the high `completion_prediction` subgroup, and little better than T-Tests for the low `completion_prediction` subgroup. These are the subgroups defined based on $\hat{y}_i^r$; since the variance of $\hat{y}_i^r$ is, by definition, lower in these subgroups than in the sample as a whole, there is less opportunity to use it for variance reduction.

Aside from those defined based on `completion_prediction`, there was little difference in ReLOOP's effectiveness between subgroups. In every case the lower quartile was greater than 1, though the lower tail reached below 1. For comparisons between ReLOOP and T-Tests, the median ratio was between 1.1 and 1.25, while for ReLOOP+/LOOP comparisons, the medians were somewhat lower.

Figure 6.1 plots the sampling variance ratios comparing ReLOOP to T-Tests and ReLOOP+ to LOOP against each subgroup's sample size. A semi-parametric regression fit (the natural logarithm of the sample size ratio regressed on a b-spline of the log of sample size with four degrees of freedom) is plotted over the points. The standard error shown is adjusted for the correlation of ratios from the same experiment. There is little evidence of a trend in the mean improvement due to ReLOOP—instead, it appears fairly constant as sample size varies. On the other hand, the range and spread of possible ratio decreases markedly as sample size increases. Every case in which ReLOOP hurt the precision relative to T-Tests was in a subgroup with $n < 100$, as were all but one of the cases when ReLOOP adjustment was equivalent to multiplying the sample size by 2.5 or higher, relative to T-Tests. Apparently ReLOOP's greatest potential for radically improving statistical precision occurs in relatively small samples. On the other hand,



Boxplots of the ratios of sampling variances of $\hat{\tau}^{DM}$ (T-Tests), $\hat{\tau}_{LOOP}(\boldsymbol{x})$ (LOOP), $\hat{\tau}_{LOOP}(\hat{y}^r)$ (ReLOOP), and $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ (ReLOOP+) for each subgroup considered. Outliers are omitted. The Y-axis is on a logarithmic scale.

in relatively small samples the asymptotic guarantee that ReLOOP cannot increase sampling variance apparently does not hold consistently.

7. Research Question 3: ReLOOP with an Unrepresentative Remnant

Previous sections illustrated the potential for a model fit in the remnant to improve the precision of treatment effect estimates in A/B tests, without assuming that both datasets were drawn from the same population. However, in previous examples it was not always entirely clear in what way the data from the remnant may or may not have been representative of RCT data. In this section, we examine a case where the remnant is primarily composed of one demographic subgroup, while the RCT is a mix of subgroups.

In particular, we describe an experiment in which we intentionally designed the remnant to differ from the RCT, in order to investigate the impact remnant unrepresentativeness may have on ReLOOP or ReLOOP+'s ability to improve statistical precision.

The experiment builds on the analyses of previous sections. However, to illustrate the effects of a remnant that is not representative of the RCT, we re-trained $\hat{y}^{REM(0)}(\cdot; \boldsymbol{\beta})$ using a subset composed disproportionally (though not entirely) of white and Asian males, and examined the estimated sampling variance of the $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ estimator for the entire RCT, for a similarly-composed subset, and for that subset's complement.

7.1. "Inferred Gender"

To help maintain students' privacy, ASSISTments does not gather data on student demographics. However, the ASSISTments foundation gathers (but does not publish) students' names, to facilitate classroom instruction (teachers need to know which student's assignment they are grading). For some analyses on ASSISTments data, analysts will attempt to guess a student's gender

identification based on that student's name. To do so, the Python package "gender-guesser"[4] was given each student's first name. The gender-guesser package uses a library of names and a script released by the German tech magazine, Heise, to determine which gender a name is associated with based on input from native speakers of various European and Asian languages. The script categorizes a name as being male, female, mostly male, mostly female, androgynous, or unknown if the name is not in the library. Clearly, this process is faulty and inexact. That being said, there is good reason to believe that most students who are inferred to be male or mostly-male are male, and most inferred to be female or mostly-female are female.

There is also reason to believe that the "unknown" category has a higher proportion of non-Asian racial or ethnic minorities or immigrants than the inferred male or female categories. This claim follows from the assumption that names that are not in the library are uncommon, and that uncommon names are probably most common among populations with non-European or non-Asian language traditions (including immigrants and native speakers with non-European or non-Asian cultural traditions) and African Americans, since there is a long tradition of distinctive naming in the African American community (Cook et al., 2014).

It follows that while the set of students labeled "Male" or "mostly-male" includes students with diverse genders, ethnicities, and linguistic traditions, it includes a disproportionate number of white and Asian males. In this way, this set of students follows an unfortunate, though common, pattern of disproportionately white male training sets for machine learning algorithms (Denton et al., 2020).

To demonstrate the ability of the $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ estimator to estimate internally-unbiased causal effects, even when the remnant reflects common biases in training datasets, we artificially limited the remnant to students labeled "Male" or "mostly-male". Then, we estimated three sets of effects: one in which the RCT was limited in the same way as the remnant—i.e. to students labeled as male—another in which only the students who would be excluded from the remnant—those not labeled male—and the complete RCT data.

7.1.1. Results

Using the predictions from the model described above, we estimated $\overline{\tau}_{RCT}$ for each experimental contrast in four ways: with the difference-in-means estimator $\hat{\tau}^{DM}$, with $\hat{\tau}_{LOOP}(\boldsymbol{x})$, a LOOP estimator using aggregated student covariates but without using any information from the remnant, with $\hat{\tau}_{LOOP}(\hat{y}^r)$, a LOOP estimator using *only* predictions from the remnant, and with $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$, which uses both aggregated student-level covariates and the predictions from the remnant.

Figure 5 shows the results comparing estimators that use imputations from the remnant to those that do not. Both estimators $\hat{\tau}_{LOOP}(\hat{y}^r)$ and $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ are almost always more precise than the difference in means estimator $\hat{\tau}^{DM}$. The only exception is a handful of cases in the Male RCTs in which including remnant imputations is equivalent to decreasing the sample size by 10% or less. This is mostly due to very small samples in some RCTs. On the other hand, in roughly half of the RCTs the improvement was 10% or more, and in many it was upwards of 30%. Comparing $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ to an estimator that uses other covariate adjustment, $\hat{\tau}_{LOOP}(\boldsymbol{x})$, produced somewhat more modest gains, but still impressive. Most surprisingly, the estimators performed as well or better in the non-Male sets and the full RCTs.

[4] https://pypi.org/project/gender-guesser/

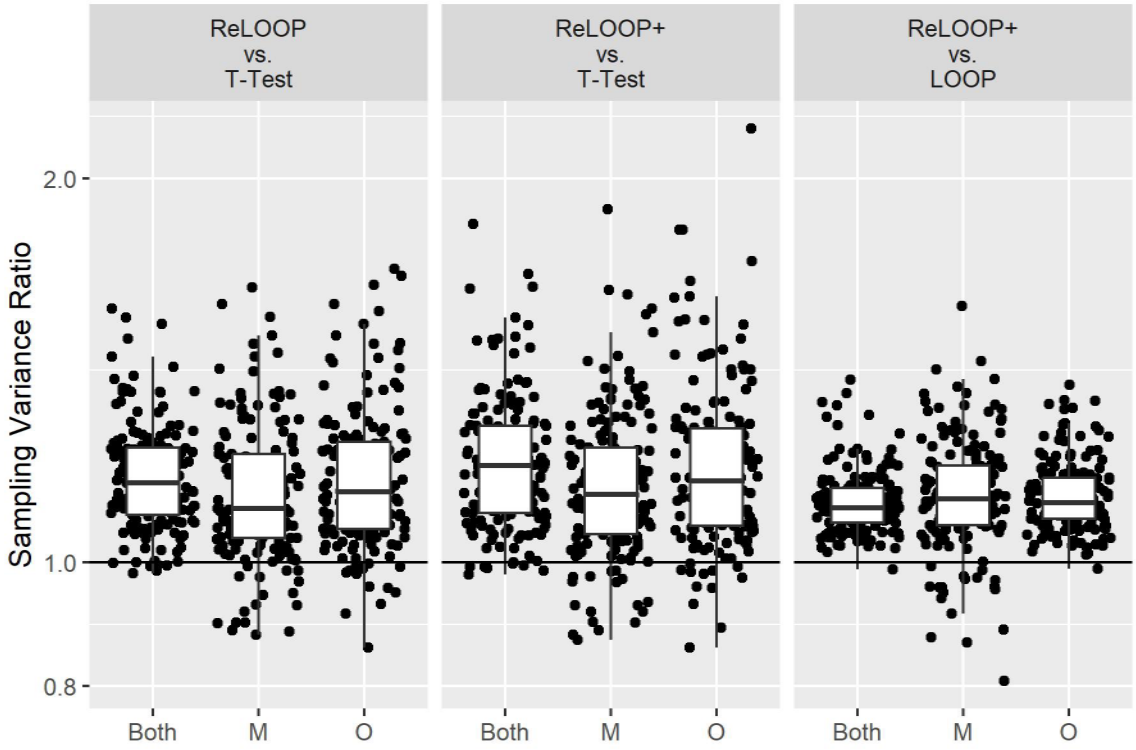


Figure 5: Results comparing estimators using imputations from the remnants $\hat{\tau}_{LOOP}(\hat{y}^r)$ or $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ (with or without other covariates) to estimators that do not, $\hat{\tau}^{DM}$ and $\hat{\tau}_{LOOP}(\boldsymbol{x})$. For all analyses, the remnant was composed of only students whose inferred gender was male; imputations from a model trained on the male remnant were used to analyze A/B tests including all participants ("Both"), or just inferred male ("M") or inferred non-male ("O").

8. Research Question 4: ReLOOP for Population Average Effects

Previous sections have focused on estimating $\overline{\tau}_{RCT}$, the average effect of a treatment for subjects in an RCT. However, often researchers are interested in $\mathbb{E}_{POP}[\tau]$, average effects across a wider population, POP. This section describes, first, how an unbiased estimate of $\overline{\tau}_{RCT}$ may still be biased for $\mathbb{E}_{POP}[\tau]$, and then describes a method for reducing some of this bias, and illustrates a way in which ReLOOP and ReLOOP+ can improve $\mathbb{E}_{POP}[\tau]$ estimation.

8.1. Estimating $\mathbb{E}_{POP}[\tau]$

An unbiased estimator of $\overline{\tau}_{RCT}$ may still be biased for $\mathbb{E}_{POP}[\tau]$, depending on the population of interest POP. For instance, consider a stylized example in which G encoded income level: poor $G = 1$ versus rich $G = 2$, and that the effect of an intervention differs by income level—say $\mathbb{E}_{POP}[\tau \mid G = 1] < \mathbb{E}_{POP}[\tau \mid G = 2]$—and that sample proportions $p_1 < p_2$ while population proportions $\pi_1 > \pi_2$, so the experiment was conducted among subjects who were wealthier, on average, than the population of interest. Finally, say that within income groups G, the experimental subjects are representative of the corresponding subgroups in the population, so that $\mathbb{E}[\overline{\tau}_{G=k}] = \mathbb{E}_{POP}[\tau \mid G = k]$. Let $\hat{\tau}$ be an unbiased estimator of $\overline{\tau}_{RCT}$. As an estimate of the population average effect $\mathbb{E}_{POP}[\tau]$, $\hat{\tau}$ will be biased:

$$\begin{aligned}
&\mathbb{E}[\hat{\tau}] - \mathbb{E}_{POP}[\tau] = \mathbb{E}[\overline{\tau}_{RCT}] - \mathbb{E}_{POP}[\tau] \\
=&p_1 \mathbb{E}\left[\overline{\tau}_{G=1}\right] + (1-p_1)\mathbb{E}\left[\overline{\tau}_{G=2}\right] - \pi_1 \mathbb{E}_{POP}[\tau \mid G=1] - (1-\pi_1)\mathbb{E}_{POP}[\tau \mid G=2] \\
=&(p_1 - \pi_1)\mathbb{E}_{POP}[\tau \mid G=1] + (\pi_1 - p_1)\mathbb{E}_{POP}[\tau \mid G=2] \\
=&(p_1 - \pi_1)(\mathbb{E}_{POP}[\tau \mid G=1] - \mathbb{E}_{POP}[\tau \mid G=2]) > 0
\end{aligned} \tag{4}$$

since $p_2 = 1 - p_1$ and $\pi_2 = 1 - \pi_1$. It is clear from (4) that if either $p_1 = \pi_1$, so that the subjects in the experiment are representative of POP, or if $\mathbb{E}_{POP}[\tau \mid G = 1] = \mathbb{E}_{POP}[\tau \mid G = 2]$, so that the average effect of the treatment doesn't vary with G, that $\hat{\tau}$ will be unbiased. In general, for an estimate to be externally biased, there must be at least one (observed or unobserved) characteristic in which the subjects in the experiment do not represent the population, *and* which predicts variation in the treatment effect. If the ways in which the experimental sample is unrepresentative are unrelated to treatment effect variation, then there will be no external bias.

Since, in the example above, $\hat{\tau}$ was unbiased for $\overline{\tau}_{RCT}$, the bias of (4) is purely external bias. However, if internal bias is also present, then the two biases add, so that

$$\mathbb{E}\left[\hat{\tau}\right] - \mathbb{E}_{POP}[\tau] = \text{internal bias} + \text{external bias} \tag{5}$$

Note, however, that if internal and external bias have opposite signs, they may (partially) cancel each other out—that said, it is hard to know when this fortunate situation may or may not hold.

8.1.1. Subgroup Effects and Bias

If $\hat{\tau}_{G=k}$ is an estimator of $\overline{\tau}_{G=k}$, it may be subject to its own internal bias, and if it is an estimator of $\mathbb{E}_{POP}[\tau \mid G = k]$, it may be subject to both internal and external bias, just like estimates of the full sample or population average effects.

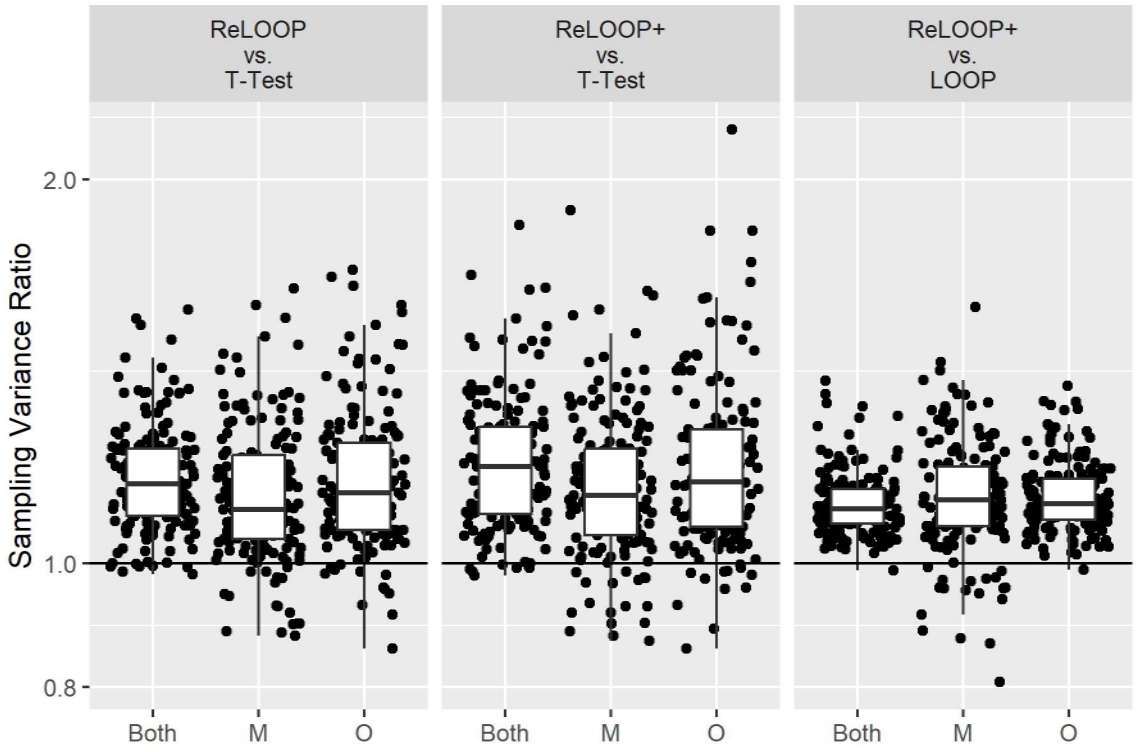


Figure 6: Results comparing post-stratification estimators using imputations from the remnants $\hat{\tau}_{LOOP}(\hat{y}^r)$ or $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ (with or without other covariates) to estimators that do not, $\hat{\tau}^{DM}$ and $\hat{\tau}_{LOOP}(\boldsymbol{x})$

On the other hand, if $\mathbb{E}[\overline{\tau}_{G=k}] \approx \mathbb{E}_{POP}[\tau \mid G = k]$ as in the example of § 8.1, and if population proportions π_k are known, then estimated subgroup effects can reduce external bias, via post-stratification (Miratrix et al., 2013). Let $\hat{\tau}_k$ be unbiased estimates of $\overline{\tau}_{G=k}$; then,

$$\mathbb{E}\left[\sum_k \pi_k \hat{\tau}_k\right] = \sum_k \pi_k \mathbb{E}[\hat{\tau}_k] \approx \sum_k \pi_k \mathbb{E}_{POP}[\tau \mid G = k] = \mathbb{E}_{POP}[\tau] \tag{6}$$

Hence, accurate estimation of subgroup effects can reduce external bias of overall population effects.

8.2. Post-Stratification for Estimating $\mathbb{E}_{POP}[\tau]$

To attempt to estimate $\mathbb{E}_{POP}[\tau]$, we conducted a post-stratification estimator (6) using the guessed gender predictor. While we do not observe the true distribution of guessed gender among all middle school ASSISTments users, we may estimate it from the remnant. When we do so, we find that roughly a third are labeled "Male."

We calculated four post-stratified estimators for each treatment contrast, using the four sets of $\overline{\tau}_{RCT}GG = Male$ estimates. Then, as in $\overline{\tau}_{RCT}$ estimation, we gauged wither $\hat{\tau}_{LOOP}(\hat{y}^r, \boldsymbol{x})$ improves the statistical precision of $\hat{\tau}^{DM}$ or $\hat{\tau}_{LOOP}(\boldsymbol{x})$.

Figure 6 shows similar results for post-stratification. Indeed, including imputations from the remnant improves the precision of these estimators greatly.

9. Conclusion

Using remnant-trained models to predict A/B test outcomes, then using those predictions to estimate effects, has the potential to boost the precision of average effect estimators in education research. For typical analysis of A/B testing results, the use of remnant-based imputations could be equivalent to increasing the sample size by as much as 40-50% relative to t-tests and as much as 30% relative to state-of-the art unbiased, covariate adjusted effect estimators. Further, in the A/B tests we analyzed, incorporating remnant-based imputations never noticeably harmed precision.

The benefits of remnant-based predictions were even more pronounced in estimating subgroup effects, and could be roughly equivalent to increasing the sample size by factors of 2, 3, or more. On the other hand, for subgroups with fewer than 100 students, there was a small risk that incorporating remnant-based predictions could harm precision instead of improving it.

The benefits of using the remnant appear to extend to cases in which the remnant does not resemble data from A/B tests on demographic characteristics. In fact, counterintuitively, we found greater benefits in the subgroup that was least represented in the remnant.

Finally, we found that incorporating remnant-based predictions into a post-stratification model can substantially improve post-stratified estimates, and hence help researchers generalize their findings to broader populations.

10. Acknowledgements

This work was funded by IES grant #R305D210031.

References

Aronow, P. M. and Middleton, J. A. 2013. A class of unbiased estimators of the average treatment effect in randomized experiments. *Journal of Causal Inference 1,* 1, 135–154.

Benjamini, Y. and Hochberg, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological) 57,* 1, 289–300.

Benjamini, Y. and Yekutieli, D. 2001. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, 1165–1188.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. 2018. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal 21,* 1, C1–C68.

Cook, L. D., Logan, T. D., and Parman, J. M. 2014. Distinctively black names in the american past. *Explorations in Economic History 53*, 64–82.

Denton, E., Hanna, A., Amironesei, R., Smart, A., Nicole, H., and Scheuerman, M. K. 2020. Bringing the people back in: Contesting benchmark machine learning datasets. *arXiv preprint arXiv:2007.07399*.

Ding, P., Li, X., and Miratrix, L. W. 2017. Bridging finite and super population causal inference. *Journal of Causal Inference 5,* 2.

Fisher, R. A. 1935. *Design of experiments*. Oliver and Boyd, Edinburgh.

FREEDMAN, D. A. 2008. On regression adjustments to experimental data. *Advances in Applied Mathematics 40,* 2, 180–193.

GAGNON-BARTSCH, J. A., SALES, A. C., WU, E., BOTELHO, A. F., ERICKSON, J. A., MIRATRIX, L. W., AND HEFFERNAN, N. T. 2021. Precise unbiased estimation in randomized experiments using auxiliary observational data. *arXiv preprint arXiv:2105.03529*.

GERS, F. A., SCHMIDHUBER, J., AND CUMMINS, F. 2000. Learning to forget: Continual prediction with lstm. *Neural computation 12,* 10, 2451–2471.

HARRISON, A., SMITH, H., HULSE, T., AND OTTMAR, E. R. 2020. Spacing out! manipulating spatial features in mathematical expressions affects performance. *Journal of Numerical Cognition 6,* 2, 186–203.

HEFFERNAN, N. T. AND HEFFERNAN, C. L. 2014. The assistments ecosystem: building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education 24,* 4, 470–497.

HELLER, R., ROSENBAUM, P. R., AND SMALL, D. S. 2009. Split samples and design sensitivity in observational studies. *Journal of the American Statistical Association 104,* 487, 1090–1101.

IMBENS, G. W. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and statistics 86,* 1, 4–29.

KINGMA, D. P. AND BA, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

MCDERMOTT, R. 2011. Internal and external validity. *Cambridge handbook of experimental political science*, 27–40.

MIRATRIX, L. W., SEKHON, J. S., AND YU, B. 2013. Adjusting treatment effect estimates by post-stratification in randomized experiments. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 75,* 2, 369–396.

NEYMAN, J. 1923. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science 5*, 463–480. 1990; transl. by D.M. Dabrowska and T.P. Speed.

OSTROW, K. S., SELENT, D., WANG, Y., VAN INWEGEN, E. G., HEFFERNAN, N. T., AND WILLIAMS, J. J. 2016. The assessment of learning infrastructure (ali): the theory, practice, and scalability of automated assessment. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*. ACM, 279–288.

RUBIN, D. B. 1978. Bayesian inference for causal effects: The role of randomization. *6*, 34–58.

SALES, A., BOTELHO, A., PATIKORN, T., AND HEFFERNAN, N. T. 2018a. Using big data to sharpen design-based inference in a/b tests. In *Proceedings of the Eleventh International Conference on Educational Data Mining*.

SALES, A. C., BOTELHO, A., PATIKORN, T. M., AND HEFFERNAN, N. T. 2018b. Using big data to sharpen design-based inference in a/b tests. In *Proceedings of the 11th International Conference on Educational Data Mining. International Educational Data Mining Society*. 479–486.

SALES, A. C., PRIHAR, E., GAGNON-BARTSCH, J., GURUNG, A., AND HEFFERNAN, N. T. 2022. More powerful a/b testing using auxiliary data and deep learning. In *International Conference on Artificial Intelligence in Education*. Springer, 524–527.

SCHOCHET, P. Z. 2015. Statistical theory for the RCT-YES software: Design-based causal inference for RCTs. NCEE 2015-4011. *National Center for Education Evaluation and Regional Assistance*.

VAN DER LAAN, M. J. AND ROSE, S. 2011. *Targeted learning: causal inference for observational and experimental data*. Springer Science & Business Media.

WAGER, S., DU, W., TAYLOR, J., AND TIBSHIRANI, R. J. 2016a. High-dimensional regression adjustments in randomized experiments. *Proceedings of the National Academy of Sciences 113,* 45, 12673–12678.

WAGER, S., DU, W., TAYLOR, J., AND TIBSHIRANI, R. J. 2016b. High-dimensional regression adjustments in randomized experiments. *Proceedings of the National Academy of Sciences 113,* 45, 12673–12678.

WU, E. AND GAGNON-BARTSCH, J. A. 2018a. The loop estimator: Adjusting for covariates in randomized experiments. *Evaluation review 42,* 4, 458–488.

WU, E. AND GAGNON-BARTSCH, J. A. 2018b. The LOOP estimator: Adjusting for covariates in randomized experiments. *Evaluation Review 42,* 4, 458–488.

A. VARIABLES USED IN REMNANT IMPUTATION MODEL

Table 5: Prior Student Statistics Features

Name	Description
target_sequence	The ID of the experimental skill builder
has_due_date	Whether the skill builder had a due date
assignments_started	The number of assignments previously started by the student
assignments_percent_completed	The number of assignments previously completed by the student
median_ln_assignment_time_on_task	The median of the log of the time between starting and finishing an assignment for all the students completed prior assignments
average_problems_per_assignment	The average number of problems completed by the student across all their previous assignments
median_ln_problem_time_on_task	The median of the log of the time the student took between starting and finished all their completed prior problems
median_ln_problem_first_response_time	The median of the log of the time the student took to submit their first answer or request tutoring across all their completed prior problems
average_problem_correctness	The fraction of previously completed problems the student got correct on their first attempt without tutoring
average_problem_attempt_count	The average number of attempts for all problems previously completed by the student
average_answer_first	The fraction of times the student submitted an answer before requesting tutoring for all problems previously completed by the student
average_problem_hint_count	The average number of hints requested for all problems previously completed by the student
skill_average_problems_per_assignment skill_median_ln_problem_time_on_task skill_median_ln_problem_first_response_time skill_average_problem_correctness skill_average_problem_attempt_count skill_average_answer_first skill_average_problem_hint_count	These features are the same as the features above with a similar name, but only calculate statistics across problems with the same skills as the problems in the experimental skill builder

Table 6: Prior Assignment Statistics Features

Name	Description
id	The ID of the student
assignment_start_time	The UNIX time of when the assignment was started
directory_1	The highest level directory of the assignment location, usually an indication of curriculum
directory_2	The second level directory of the assignment location, usually an indication of grade level
directory_3	The third level directory of the assignment location, usually an indication of unit
sequence_id	The unique ID of the skill builder assignment, or the corresponding normal skill builder ID for experiments
is_skill_builder	Boolean flag for whether or not this assignment is a skill builder or a normal problem set
has_due_date	Boolean flag for if the assignment has a due date
assignment_completed	Boolean flag for if the student completed the assignment
time_since_last_assignment_start	The time between the student starting this assignment and starting their prior assignment
All Following Features	In addition to the raw value, a value z-scored across all students who completed the assignment previously, and a percentile across students in the same class who completed the assignment previously was included in the model as well.
session_count	How many times the student left and rejoined the assignment
day_count	How many days the student worked on the assignment for
completed_problem_count	How many problems the student completed in the assignment
median_ln_problem_time_on_task	The median of the log of the time between the student starting and finishing problems in the assignment
median_ln_problem_first_response	The median of the log of the time it took for the student to submit their first answer or request tutoring on the problems they started in the assignment
average_problem_attempt_count	The average number of attempts the student made on the problems in the assignment
average_problem_answer_first	The fraction of times the student made an attempt before requesting tutoring on all the problems in the assignment
average_problem_correctness	The fraction of times the student got the problem correct on their first try on all the problems in the assignment
average_problem_hint_count	The average number of hints used by the student on all the problems in the assignment
average_problem_answer_given	The fraction of times the student was given the answer on all the problems in the assignment

Table 7: Prior Daily Actions Features

Name	Description
id	The ID of the student
timestamp	The UNIX time at 00:00:00 of the day the action counts apply to
ln_action_1_count	Log of the count of assignment started actions taken
ln_action_2_count	Log of the count of assignment resumed actions taken
ln_action_3_count	Log of the count of assignment finished actions taken
ln_action_4_count	Log of the count of problem set started actions taken
ln_action_5_count	Log of the count of problem set resumed actions taken
ln_action_6_count	Log of the count of problem set finished actions taken
ln_action_7_count	Log of the count of problem set mastered actions taken
ln_action_8_count	Log of the count of problem set exhausted actions taken
ln_action_9_count	Log of the count of problem limit exceeded actions taken
ln_action_10_count	Log of the count of problem started actions taken
ln_action_11_count	Log of the count of problem resumed actions taken
ln_action_12_count	Log of the count of problem finished actions taken
ln_action_13_count	Log of the count of tutoring set started actions taken
ln_action_15_count	Log of the count of tutoring set finished actions taken
ln_action_16_count	Log of the count of hint requested actions taken
ln_action_17_count	Log of the count of scaffolding requested actions taken
ln_action_19_count	Log of the count of explanation requested actions taken
ln_action_20a_count	Log of the count of student correct response actions taken
ln_action_20b_count	Log of the count of student incorrect response actions taken
ln_action_21_count	Log of the count of open response submission actions taken
ln_action_25_count	Log of the count of answer requested actions taken
ln_action_26_count	Log of the count of continue selected actions taken
ln_action_30_count	Log of the count of help requested actions taken
ln_action_31_count	Log of the count of timer started actions taken
ln_action_32_count	Log of the count of timer resumed actions taken
ln_action_33_count	Log of the count of timer paused actions taken
ln_action_34_count	Log of the count of timer finished actions taken
ln_action_35_count	Log of the count of live tutoring requested actions taken
Other Actions	Artifacts of the database, always 0

Chapter 1.7

Effective Evaluation of Online Learning Interventions with Surrogate Measures

Effective Evaluation of Online Learning Interventions with Surrogate Measures

Ethan Prihar
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA, USA
ebprihar@wpi.edu

Kirk Vanacore
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA, USA
kpvanacore@wpi.edu

Adam Sales
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA, USA
asales@wpi.edu

Neil Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA, USA
nth@wpi.edu

ABSTRACT

There is a growing need to empirically evaluate the quality of online instructional interventions at scale. In response, some online learning platforms have begun to implement rapid A/B testing of instructional interventions. In these scenarios, students participate in series of randomized experiments that evaluate problem-level interventions in quick succession, which makes it difficult to discern the effect of any particular intervention on their learning. Therefore, distal measures of learning such as posttests may not provide a clear understanding of which interventions are effective, which can lead to slow adoption of new instructional methods. To help discern the effectiveness of instructional interventions, this work uses data from 26,060 clickstream sequences of students across 31 different online educational experiments exploring 51 different research questions and the students' posttest scores to create and analyze different proximal surrogate measures of learning that can be used at the problem level. Through feature engineering and deep learning approaches, next problem correctness was determined to be the best surrogate measure. As more data from online educational experiments are collected, model based surrogate measures can be improved, but for now, next problem correctness is an empirically effective proximal surrogate measure of learning for analyzing rapid problem-level experiments.

Keywords

Surrogate Measures, Measures of Learning, A/B Testing, Educational Experiments

1. INTRODUCTION

There is a growing need to empirically evaluate the quality of online instructional interventions at scale. This is in part motivated by the lack of empirical evidence for many existing interventions, especially in mathematics. According to Evidence for ESSA, a website that tracks empirical research on educational practices created by the Center for Research and Reform in Education at Johns Hopkins University School of Education, only four technology based interventions have strong evidence for improving students' mathematics skills [5]. In response, more and more online learning platforms are creating infrastructure to run randomized controlled experiments within their platforms [21, 12, 20] in order to increase the impact of the their programs on student learning and facilitate research in the field. This infrastructure allows for rapid A/B testing of different instructional interventions. In an A/B testing scenario, students assigned to particular assignments or problems within these online learning platforms will be automatically randomized to one of multiple experimental conditions in which different instructional interventions will be provided to them. While this paradigm allows for rapid testing of many hypotheses, this rapid testing environment makes statistical analysis difficult. In some cases, students participate in many randomized controlled experiments in parallel or in quick succession. For example, in ASSISTments, an online learning platform in which students complete pre-college level mathematics assignments [9], students can be randomized between different instructional interventions for each mathematics problem in their assignment. In these scenarios, it is important to evaluate the effect of the interventions as quickly as possible. If one were to wait until the end of a section of the curriculum, or even the end of the current assignment before evaluating students' mastery of the subject matter, then the effect of an intervention for a single problem near the beginning of the assignment would be obfuscated by the effects of all the following interventions. For this reason, prior work has only used students' behavior on the problem they attempted after receiving an intervention but before receiving another intervention to evaluate the effectiveness of the first intervention [13, 17]. However, the measures used in prior work were chosen based on theory, without any empirical evidence that they are in fact an effective surrogate measure of learning.

To address the lack of empirical evidence for these proximal surrogate measures of learning, the first goal of this work was to create a variety of surrogate measures from students' clickstream data on the problem they attempted after receiving an experimental intervention. These measures were created through feature engineering, discussed in Section 3, and model fitting, discussed in Sections 4.1 and 4.2.

After creating surrogate measures, The second goal of this work was to evaluate how effective these measures were at estimating the treatment effects between pairs of conditions in online experiments. To achieve this goal, data was collected to compare 51 different pairs of conditions from 31 assignment-level online experiments with posttests in which students were exposed to the same intervention multiple times within the same assignment, but were not exposed to any other interventions. By determining the extent to which each measure was a surrogate for students' posttest scores, discussed more in Sections 2.3 and 4.4, the surrogate measures could be compared to each other.

After determining which surrogate measure was most suited for use in rapid online experiments, the third goal of this work was to explore the effects of using the chosen surrogate to analyze the results of online education experiments compared to using posttest scores to analyze the results, discussed more in Section 4.5.

To summarise, this work strives to answer the following three research questions:

1. What surrogate measures can be created from short sequences of students' clickstream data?
2. Which of these surrogate measures is the best surrogate for posttest score?
3. How does using this surrogate measure to analyze online educational experiments effect their results?

2. BACKGROUND

2.1 Rapid Online Educational Experimentation

Experimentation is a cornerstone of formative improvement of online instructional interventions [20, 1]. When making decisions about implementing changes to online learning programs, designers must understand which features will have the greatest impact on student learning. A/B testing, i.e., comparing students' performance when they are randomly exposed to different variants of feature, allows researchers to estimate the causal effect of a specific feature. This causal estimate can be used to determine which variant of a feature should be scaled system wide and inform design decisions for future product development.

Systems like ASSISTments E-TRIALS were established to allow researchers to test learning theories and feature ideas through experiments within online mathematics assignments [12]. Using systems like E-TRIALS, students are randomized between different assignment-level interventions and complete a posttest at the end of their assignment to evaluate their learning. Experiments in E-TRIALS have shown that providing explanations, hints, or scaffolding questions to students tends improves their performance more than simply providing them with the answer after an incorrect attempt [18]. Experiments in E-TRIALS have evaluated more than just instructional intervention based experiments. For example, experiments have shown that students' learning was negatively impacted by interjecting motivational messages into their mathematics assignments [18].

Although assignment-level experiments provide some relevant information to online program designers, these designers are faced with a nearly infinite number of decisions about what features to build and how to build them. Since only one causal inference can be estimated from each manipulation [10], designing assignment-level experiments for each potential impactful variant of a feature is often infeasible. Rapid online educational experimentation provides a more efficient alternative more traditional assignment-level experiments by assigning students to a condition at each problem and instead of requiring students to complete a posttest, using the student's performance on the subsequent problem as the outcome.

One example of rapid online educational experimentation is the TeacherASSIST system, which randomizes students between crowdsourced educator generated hints and explanations. In this system, there were over 7,000 support messages produced by 11 educators. These support messages consisted of hint messages or worked explanations in both text and video form. These educator created problem-level support messages produced an average positive effect on student performance [13, 17] and more work is being done to understand the nuanced effects of each tutoring message [15]. This system has allowed for a much more efficient deployment of experiments and evaluation of feature nuances.

2.2 Unconfounded Outcomes For Rapid Online Experiments

In order for rapid online experimentation to increase the number of casual inferences made, we must identify outcomes that are unconfounded by the other experimental manipulations to which a student was exposed. Distal outcomes, such as end-of-unit or assignment-level posttest scores, do not allow a researcher to determine which of the treatments the student was exposed to during the experiment produced the effect. An alternative, used by [13, 17] to evaluate TeacherASSIST, is to use data from the problem students completed directly after the experimental condition, i.e., next problem measures.

Although individual students' behaviors and performance may be influenced by the aggregate of experimental manipulations within an assignment, the average difference in next problem measures is unconfounded due to the random assignment at the problem level. Next problem measures are unconfounded by either the prior experimental conditions or next problem experimental conditions because the assignment to each condition is independently random and therefore the effects of the prior and post-conditions are zero. Therefore, the remaining difference in the next problem measures between treatment and control is an unconfounded measure of the treatment effect.

2.3 Surrogate Measures

Although measures taken during the next problem after the experiment, such as next-problem correctness, are unconfounded by other experiments within the problem set, it is not yet known whether these measure are good estimates of distal outcomes. In assignment-level A/B testing, a researcher creates a posttest designed to measure the expected effect of the treatment condition compared to the control condition, but within online instructional interventions, the next problem was designed for pedagogical purposes, not to evaluate the effects of the intervention. Therefore, to use next problem measures to validate the impact of a condition, we must validate whether these measures assess researchers' outcomes of concern.

One way to think about these next problem measures is as surrogate measures. Surrogate measures are used in medical experiments when the outcome is either difficult to assess or distal [19]. Surrogates can either have causal or correlation relations to the outcome [11]. Validating causal surrogates requires a causal path from the treatment to the surrogate and subsequently to the outcome, such that the indirect path through the surrogate has a larger effect than the direct path through from the treatment to the outcome. Alternatively, an associative surrogate is valid when the following three criteria are met [11]:

1. There is a monotonic relationship between the treatment effect on the surrogate and the treatment effect on the outcome across experiments.
2. When the treatment effect on the surrogate is zero, the treatment effect on the outcome is also zero.
3. The treatment effect on the surrogate predicts the treatment effect on the outcome.

In this work, various next problem measures are evaluated for their effectiveness as an associative surrogate measure of posttest scores.

3. DATA COLLECTION AND PREPARATION

3.1 Data Source

The data used in this work comes from ASSISTments, an online learning platform that focuses on pre-college mathematics curricula. Within ASSISTments, external researchers can run experiments at scale that compare different instructional interventions. In July, 2022 ASSISTments released a dataset of 88 randomized controlled experiments that were conducted within the platform since 2018 [18]. These experiments compared various assignment-level and problem-level interventions. For example, Fig. 1 shows the two conditions of an ASSISTments experiment in which students were randomized between receiving either open response problems, or multiple choice problems.

In this work, only the experimental assignments from ASSISTments that had posttests were used. This ensured that any learning measures derived from a student's clickstream data on the problem immediately after receiving an intervention for the first time could be directly compared to their posttest score. A students posttest score is the fraction of

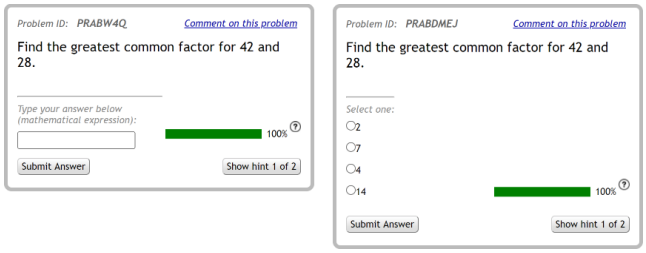


Figure 1: An example of two experimental conditions. In the first condition (left), students are given open response versions of mathematics problems. In the second condition (right), students are given multiple choice versions of the same problems.

problems they answered correctly on their posttest. To avoid bias from missing posttest scores, only data from experiments in which there was no statistically significant difference in students' completion rates between conditions were used, and students that did not complete the posttest were excluded from the analysis. In some contexts it would be better to impute missing posttest scores as the minimum score. However, the purpose of this work was to create a surrogate measure for posttest score in situations where it is infeasible to require students to complete a posttest, and therefore it seems more appropriate to remove missing posttest scores to ensure that the surrogate measures students' posttest scores, not their propensity to complete an assignment. This additional filtering step removed only one of the ASSISTments experiments from the analysis. Additionally, the data used in this work is limited to students who participated in the experiments prior to July 23rd, 2021. On July 23rd, 2021 all unlisted YouTube videos created prior to 2017 were made private [7]. Many of the experiments included YouTube videos uploaded prior to 2017, which were made private, ruining the experiments that contained them.

These experiments provided a rare opportunity to fairly compare next problem measures to posttest score because typically, when next problem measures are used as a dependent measure, it is because many different types of interventions are being given to a student in quick succession. However, in these experiments students are given the same intervention for each problem in the experimental assignment. Therefore, in these experiments, next problem measures measure the student's propensity to learn the material after seeing the experimental intervention for the first time, and posttest score measures the student's propensity to learn the material after seeing the same intervention multiple times, but both are evaluating the effectiveness of the same intervention. In total, 26,060 clickstream sequences of a student completing a problem and their corresponding posttest score were collected for model training and analysis across 51 different research questions within 31 different experimental assignments. These sequences and the code used to evaluate them has been made publicly available and can be found at https://osf.io/uj48v/.

3.2 Expert Features

As established by prior work, i.e. ([13], [17], [15]), collecting data to evaluate the effectiveness of an intervention is often

limited to data from the next problem in a student's assignment, before they received another intervention. While next problem correctness was used in prior work, this work extracted four additional expert features from students' clickstream data on their next problem that have been useful predictors of student behavior in prior work [22, 23]. Table 1 describes the expert features used in this work.

3.3 Clickstream Data

In addition to expert features, this work used deep learning to create surrogate measures of learning from students' clickstream data. The clickstream data consisted of the action sequences of students within the ASSISTments tutor from the time they start the problem after they received an experimental intervention to the time they either receive another intervention or complete the problem. This short window of time is not confounded by other experimental interventions and is likely to give the clearest insight into the impact of experimental interventions being tested in quick succession.

The students' clickstream data was broken down into a series of one-hot encoded actions followed by the time since taking the last action. The first action was always "problem_started", therefore this action was dropped from students' clickstreams prior to being given to a deep learning model. The time since taking the last action was log-transformed in order to weight the difference between short time periods more than long time periods and to reduce the impact of large outliers, which are due to students walking away from their computers during assignments and returning later. Additionally, the log-transformed times are scaled within the range [0, 1]. Scaling the time within the same range as the one-hot encoded actions helps the model balance the importance of the different features. Each action sequence was equal in length to the longest action sequence, which was 12 actions. When students took less than the maximum number of actions, their action sequences were zero padded from the start of the sequence. Table 2 provides an example sequence of a student's clickstream data in which a student unsuccessfully attempted to get a problem correct twice, then took a break, then returned to their assignment, got the problem incorrect again, and then on their fourth attempt, got the problem correct. The first six columns contain all zeros because the student only took a total of six actions. This representation of students' clickstream action sequences was chosen because of previous work's success with this representation for various prediction tasks [22, 16, 23].

4. METHODOLOGY

4.1 Expert Feature-Based Models

To derive a surrogate measure of learning from the expert features, three approaches were taken. The first approach was to simply use each expert feature as a surrogate measure of learning. If an expert feature could be used as an effective surrogate measure, it would make it much easier for researchers and online learning platforms to adopt this measure, as no model fitting would be required. The second approach was to fit a linear regression on posttest score using the expert features as input. Equation 1 shows the model fit for approach two, where n is the number of students, f is the number of features, Y is an n by 1 matrix of students' posttest scores, X is an n by f matrix of students' feature values, and β is an f by 1 matrix of coefficients learned during model fitting.

$$Y = X\beta \tag{1}$$

The third approach was to fit a linear regression on the treatment effect on posttest score using the treatment effects on each expert feature as input. The third approach was included because if the goal is to predict the treatment effect on posttest score, than it might be more effective to fit a model that combines the treatment effects on different expert features into the treatment effect on posttest score than to simply predict posttest score. This would be advantageous in a scenario where there was information in the expert features that was predictive of a student's propensity to learn independent of the intervention they were given. In that scenario, a model trained to predict posttest score might learn to rely on that information, which would lead the model to predict more similar posttest scores between different experimental conditions than were actually observed. By directly predicting the treatment effect on posttest score, the model must learn to use the features that are predictive of the effect of the experimental conditions. The downside of this approach is that each research question's data is reduced to a single sample in the regression. Therefore, while the second approach had the full 26,060 samples of student data to fit on, the third approach only had 51 samples to fit on; one for each research question. Equation 2 shows the model fit for the third approach, where n is the number of students, f is the number of features, Y is an n by 1 matrix of students' posttest scores, X is an n by f matrix of students' feature values, Z is an array of conditions where 1 indicates the student was placed in the treatment condition, and 0 indicates the student was placed in the control condition, and β is an f by 1 matrix of coefficients learned during model fitting.

$$y_t = \frac{\sum_{i=1}^{n} Y_i \times Z_i}{\sum_{i=1}^{n} Z_i}, \qquad y_c = \frac{\sum_{i=1}^{n} Y_i \times (1 - Z_i)}{\sum_{i=1}^{n} 1 - Z_i}$$

$$x_t = \frac{\sum_{i=1}^{n} X_i \times Z_i}{\sum_{i=1}^{n} Z_i}, \qquad x_c = \frac{\sum_{i=1}^{n} X_i \times (1 - Z_i)}{\sum_{i=1}^{n} 1 - Z_i} \tag{2}$$

$$y_t - y_c = (x_t - x_c)\beta$$

4.2 Deep Learning Models

Two deep learning approaches were used to create a surrogate measure of learning from students' clickstream data. Both approaches trained a recurrent neural network to predict students' posttest scores given their clickstream data using Bidirectional LSTM layers [24, 6], which read the clickstream data both forward and backward to learn the relationship between students' actions and their posttest scores. Following the same intuition as the previous section, while the first model used the mean squared error of its posttest score predictions as its loss function, the second model used the squared error of the treatment effect calculated from its posttest score predictions as its loss function.

Table 1: Expert Features

Feature Name	Description
Correctness	A binary indicator of whether or not the student answered the problem correctly on their first try without tutoring of any kind.
Tutoring Requested	A binary indicator of whether or not the student requested tutoring of any kind.
No Attempts Taken	A binary indicator of whether or not the student did not make any attempts to answer the problem.
Attempt Count	The number of attempts made by the student to answer the problem.
First Response Time	The natural log of the total seconds from when the problem was started to when the student submitted an answer or requested tutoring of any kind for the first time.

Table 2: A Student's Clickstream Data Sequence After Processing

Feature Name	Clickstream Data Sequence											
problem_resumed	0	0	0	0	0	0	0	0	1	0	0	0
tutoring_requested	0	0	0	0	0	0	0	0	0	0	0	0
wrong_response	0	0	0	0	0	0	1	1	0	1	0	0
correct_response	0	0	0	0	0	0	0	0	0	0	1	0
problem_finished	0	0	0	0	0	0	0	0	0	0	0	1
time_since_last_action	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.51	6.39	0.12	0.38	0.01

Essentially, the first model was trained to predict accurate posttest scores, and the second model was trained to predict posttest scores that would lead to the same treatment effect estimates as the actual posttest scores. For context, Equation 3 formalizes the mean squared error loss function of the first approach using the same notation as Equation 4, which formalizes the custom loss function for the second approach, where Y is an array of students' posttest scores, $\hat{Y}$, is an array of predicted posttest scores, Z is an array of conditions where 1 indicates the student was placed in the treatment condition, and 0 indicates the student was placed in the control condition, n is the number of students in the array, and τ and $\hat{\tau}$ are the treatment effects of the research question calculated using posttest and the surrogate measure respectively.

$$\text{Mean Squared Error Loss} = \frac{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}{n} \tag{3}$$

$$y_t = \frac{\sum_{i=1}^{n} Y_i \times Z_i}{\sum_{i=1}^{n} Z_i}, \qquad y_c = \frac{\sum_{i=1}^{n} Y_i \times (1 - Z_i)}{\sum_{i=1}^{n} 1 - Z_i}$$

$$\hat{y}_t = \frac{\sum_{i=1}^{n} \hat{Y}_i \times Z_i}{\sum_{i=1}^{n} Z_i}, \qquad \hat{y}_c = \frac{\sum_{i=1}^{n} \hat{Y}_i \times (1 - Z_i)}{\sum_{i=1}^{n} 1 - Z_i} \tag{4}$$

$$\tau = y_t - y_c, \qquad \hat{\tau} = \hat{y}_t - \hat{y}_c$$

$$\text{Treatment Effect Squared Error Loss} = (\hat{\tau} - \tau)^2$$

4.3 Model Training

To fairly evaluate the surrogate measures of learning, each model was trained and evaluated using leave-one-out cross-validation partitioned by the experimental assignment, and only the surrogate measures of learning calculated for the held out data were used to determine the surrogate measures' effectiveness. In each experimental assignment, multiple research questions are evaluated, but there is overlap in the data used to answer each of these research questions. For example, one experimental assignment evaluated the effectiveness of both video-based and text-based encouraging messages during an assignment. Both of these conditions shared the same control condition in which students did not receive encouraging messages. While there are two research questions being evaluated, if we trained a model using the data from all but one of these research questions, the data from the control condition of the held out research question would have been used to train the model. This would have given the model an unfair advantage. Therefore, when using leave-one-out cross-validation to train and evaluate the models, the data was partitioned by experimental assignment, and all the research questions in the held-out experimental assignment were evaluated using the model trained on all the other experimental assignments. This ensures that no data is shared between the training data and the held-out data.

For the expert feature-based models, an ablation study was performed to identify which combination of features lead to the highest correlation between surrogate measure and posttest treatment effects. In this ablation study, the models were trained first using all of the expert features as input, and then models were trained using all but one of the features. If any of the all-but-one-feature models out-performed the model with all the features, then that model became the best model so far, and more models were trained using all but one of the features in the new best model. Eventually, the best model will not have improved from removing any of its features, denoting that this model has the optimal set of features as input.

For the deep learning models, the models were initialized, trained, and evaluated ten times, and the average of all these evaluations was used to determine the quality of the deep

learning models predictions as a surrogate measure. Unlike linear regressions, neural networks cannot be solved for the optimal value of their coefficients. Instead, a neural network's weights, which are akin to a linear regression's coefficients, are randomly initialized, and then gradient descent is used to optimize them. These random initializations can lead to more or less optimal weights at the end of training. Therefore, by training the model multiple times using different random initializations and averaging the results, the evaluation of the model's surrogate measure is more reliable.

Additionally, deep learning models are highly nonlinear and are prone to over-fitting on the data, which leads to worse predictive accuracy on the held-out data. To address this, only half of the data used for training the model were used to optimize the weights for the first approach. The other half of the data was used as a validation set. The prediction error on the validation set was calculated each time the model's weights were updated. Once the prediction error on the validation set began to increase, training was stopped, because any further reduction in prediction error on the training data would be due to over-fitting on the training data, as opposed to learning the underlying relationship between students' clickstream data and their posttest scores. For the second approach, the treatment effect loss function made it more difficult for the model to learn the relationships in the data because all predictions for a single experiment were reduced to a single loss value, making it more difficult to properly attribute blame for predictive error to the weights in the model. Therefore, none of the data was used for validation during the second approach. This provided the neural network with as much information as possible. Instead, over-fitting was prevented by training the model used in the second approach for about the same number of training steps taken by the model trained for the first approach before it began to over-fit.

4.4 Evaluation of Surrogate Measures

To reiterate from Section 2.3, a surrogate measure must meet three criteria [11]:

1. There is a monotonic relationship between the treatment effect on the surrogate and the treatment effect on the outcome across experiments.

2. When the treatment effect on the surrogate is zero, the treatment effect on the outcome is also zero.

3. The treatment effect on the surrogate predicts the treatment effect on the outcome.

Criteria 1 and 3 can be simultaneously evaluated by looking at the Pearson correlation between the treatment effect on the surrogate measures and the treatment effect on posttest score because a high Pearson correlation between two measures indicates that there is a monotonic linear relationship between them [2], and the linearity implies predictability. The higher the Pearson correlation between treatment effects across all research questions, the more effective the surrogate measure is. Using the same terminology from Equation 4, the goal is to maximize $corr(\tau, \hat{\tau})$.

To evaluate Criteria 2, after the surrogate measures were used to determine the treatment effects for the different research questions, a linear regression was fit to predict the treatment effect on posttest given the treatment effect on one of the surrogate measures and an intercept. If the coefficient of the intercept is small and statistically insignificant, then there is no evidence that Criteria 2 was violated. Therefore, the best surrogate measure was determined to be the measure with the highest Pearson correlation between its treatment effects and the posttest treatment effects across all the research questions (Criteria 1 and 3), as long as the measure did not have a significant intercept when its treatment effects were used to predict the posttest treatment effects (Criteria 2).

4.5 Experiment Analysis

It is not only important to identify the best surrogate measure, but also to understand the impact that using this measure of learning would have on analyzing A/B tests and educational experiments. Therefore, after each surrogate measure of learning was evaluated, the treatment effect on both posttest score and the best surrogate measure along with the 95% confidence interval of these treatment effects were calculated for each research question using a simple difference in means between the treatment and control groups in each research question [25]. The treatment effects on the surrogate measure were then compared to the treatment effects on posttest score.

5. RESULTS

5.1 Evaluation of Surrogate Measures

The treatment effect of each research question was calculated using each surrogate measure described in Sections 4.1 and 4.2. To evaluate whether the surrogate measures met Criteria 1 and 3 from Section 4.4, the treatment effects on each surrogate measure across all the research questions were correlated with the treatment effects on posttest score. Table 3 reports the different surrogate measures, the Pearson correlation [2] of their treatment effects, and the statistical significance of these correlations.

Of all the expert features, correctness and tutoring requested were the only two features whose treatment effects were statistically significantly correlated with the treatment effect on students' posttest scores. Correctness had a positive correlation with posttest score, indicating that students that got the next problem correct on their first try without any support tended to have higher posttest scores than those who did not, and tutoring requested had a negative correlation with posttest score, indicating that students that requested tutoring on the next problem tended to have lower posttest scores than those who did not. The direction of these correlations makes intuitive sense, as one would expect students who struggle to answer mathematics problems correctly during their assignment to have difficulty on their posttest as well.

When performing the ablation study to identify the optimal set of expert features for the linear regression used to predict posttest score (Section 4.1, Approach 2), the highest performing model used only correctness. Interestingly, no other feature could be used in combination with correctness

Table 3: The Correlations between Surrogate Measure and Posttest Score Treatment Effects

Surrogate Measure	Treatment Effect Correlation with Posttest Score	Correlation p-value
Expert Features as a Surrogate Measure (Section 4.1, Approach 1)		
Correctness	**0.62**	**<0.001**
Tutoring Requested	-0.59	<0.001
No Attempts Taken	-0.01	0.935
Attempt Count	-0.16	0.264
First Response Time	0.04	0.784
Expert Features Used to Predict Posttest Score (Section 4.1, Approach 2)		
Posttest Prediction	0.62	<0.001
Expert Feature Treatment Effects Used to Predict Treatment Effect on Posttest (Section 4.1, Approach 3)		
Treatment Effect Prediction	0.50	<0.001
Deep Learning Posttest Prediction with Mean Squared Error Loss (Section 4.2, Approach 1)		
Posttest Prediction	0.60	<0.001
Deep Learning Posttest Prediction with Treatment Effect Squared Error Loss (Section 4.2, Approach 2)		
Posttest Prediction	0.49	<0.001

to improve the model's predictions. Therefore, using this linear regression to predict posttest is an equivalent surrogate measure to just using correctness as a surrogate measure itself.

When performing the ablation study to identify the optimal set of expert features for the linear regression used to predict treatment effect on posttest (Section 4.1, Approach 3), the highest performing model used tutoring requested and attempt count. Interestingly, correctness, while being the best and only feature used to predict posttest score, was not as effective at directly predicting treatment effect. Ultimately, this approach was inferior to the other approaches at identifying surrogate measures using expert features.

To evaluate Criteria 2 from Section 4.4, a linear regression was fit for each surrogate measure using data from all the research questions to predict the treatment effect on posttest given the treatment effect on the surrogate measure and an intercept. Table 4 reports the different surrogate measures, the coefficients of their linear regressions' intercepts, and and the statistical significance of these coefficients.

There was little evidence that any of the surrogate measures violated Criteria 2. Only the deep learning model with treatment effect squared error loss had an intercept coefficient that was close to statistically significant, but the p-value of 0.050 is rounded down, and that model was not a contender for best model based on the results in Table 3. Therefore, the best surrogate measure was simply next problem correctness, because the treatment effect on no other feature nor any model prediction was more correlated with the treatment effect on posttest than treatment effect on next problem correctness.

5.2 Experiment Analysis

After identifying next problem correctness as the best surrogate measure of learning, the treatment effects on posttest and on next problem correctness were calculated for each research question along with their confidence intervals. Figure 2 plots the treatment effect and confidence interval using both measures for each research question, sorted from largest to smallest posttest confidence interval. Figure 2 shows that while next problem correctness tends to lead to wider confidence intervals, it also tends to lead to larger treatment effects.

Additionally, Figure 3 shows a confusion matrix comparing the significant findings when using both measures. Only five of the 51 research questions had significant findings when using posttest score as a measure of learning. Using next problem correctness as a measure of learning resulted in six significant findings, but only one of these findings is found when using both measures to perform the analysis. However, the lack of common significant findings should not be discouraging. There is typically a sparsity of significant findings in online educational experiments, and the most important result is that the two learning measures never disagreed on which condition is better when they both identified a statistically significant difference between conditions.

6. DISCUSSION

Ultimately, next problem correctness was the best surrogate measure of learning. The treatment effect on next problem correctness had the highest Pearson correlation with the treatment effect on posttest, and there was no evidence that the treatment effect on next problem correctness was not zero when the treatment effect on posttest was zero, which satisfies all three criteria discussed in Section 2.3. It was not expected that one of the simplest of the surrogate measures, which had been used previously despite no empirical evidence to support that choice, would be the best surrogate. One possible reason for why the predictive models did not perform well is that the behavior of students within an experiment could be highly dependent on the material in the assignment. For example, geometry problems might on average take more time to answer than algebra problems, which would make students first response time less informative of their learning because it is in part dependent on the subject matter. Methods like Knowledge Tracing and Performance Factor Analysis, which measure students' mastery of mathematics concepts, take into account the knowledge components of the students' assignments when predicting student performance to compensate for this dependence [4].

Table 4: The Correlations between Surrogate Measure and Posttest Score Treatment Effects

Surrogate Measure	Intercept Coefficient	Intercept Significance p-value
Expert Features as a Surrogate Measure (Section 4.1, Approach 1)		
Correctness	-0.0084	0.133
Tutoring Requested	-0.0059	0.293
No Attempts Taken	-0.0066	0.340
Attempt Count	-0.0080	0.293
First Response Time	-0.0073	0.177
Expert Features Used to Predict Posttest Score (Section 4.1, Approach 2)		
Posttest Prediction	-0.0085	0.131
Expert Feature Treatment Effects Used to Predict Treatment Effect on Posttest (Section 4.1, Approach 3)		
Treatment Effect Prediction	-0.0098	0.152
Deep Learning Posttest Prediction with Mean Squared Error Loss (Section 4.2, Approach 1)		
Posttest Prediction	-0.0073	0.198
Deep Learning Posttest Prediction with Treatment Effect Squared Error Loss (Section 4.2, Approach 2)		
Posttest Prediction	1.94	0.050

14. By providing the models with more nuanced information about student behavior, it is possible they were picking up on behavioral trends that were not generalizable across experiments. Additionally, the sample size of the data was fairly low. Only 51 research questions were used in this analysis, and it is likely that data from more experiments testing a greater variety of interventions would help the models learn to differentiate between generalizable trends and trends specific to subsets of experiments.

These reasons help to explain what may have caused the models to underperform, but from a different perspective, what caused next problem correctness to perform so well? It seems likely that next problem correctness was a strong surrogate because posttest score is simply a different measure of problem correctness. In other words, next problem correctness is a measure of whether the student got the problem immediately following the intervention correct, and posttest score is a measure of whether the student got a few problems ahead of the intervention correct. It makes sense that two measures that revolve around a student's propensity to answer problems correctly would correlate. This leads to the question: is correctness what matters? If the goal of education is ultimately to give students better, more fulfilling lives, then perhaps test scores are not what a surrogate should measure. There is plenty of evidence of test scores falling short when attempting to correlate them with things like college and career success. For example, studies have found that SAT scores do not explain any additional variance in college GPA for non-freshman college students after taking into account social/personality and cognitive/learning factors [8]. Additionally, these test scores can be biased against minority groups. For example, studies have found that SAT scores are more predictive of white students' college GPA than they are for Black or Hispanic students [26]. While these are important factors to consider, one could argue that these impacts are less relevant in the context of this work, where the goal is simply to use short patterns in students' behavior to analyze the difference in the impact of various problem-level interventions meant to help students learn how to correctly answer the following problems in their assignments. However, one should always be cognizant of the potential bias a surrogate measure could introduce.

When using next problem correctness and posttest scores to analyze the results of the 51 research questions, only six and five of the 51 research questions had significant differences between conditions respectively, but only one of these significant findings was identified by both measures. While it would be better if the two measures found more similar significant findings, as long as the two measures do not disagree on which condition is most effective when they both find something statistically significant, then there is no concern that using next problem correctness could lead a researcher to the wrong conclusion. Next problem correctness, on average, had wider confidence intervals than posttest score, but also had larger treatment effects. This may be explained simply by the more extreme nature of the next problem correctness values. To gain some intuition on why this might be the case, consider that posttest is essentially the average of many next problem correctness measures. If we think of whether a student gets a problem correct as a random variable, then one can see how the average of many random variables will tend to be closer to the expected value than a single random sample. The variance of students' posttest scores can therefore be expected to be lower than the variance of students' next problem correctness, which would cause the confidence interval of the treatment effect on posttest to be smaller as well.

6.1 Limitations and Future Work

While in this work next problem correctness was found the be the best proximal surrogate measure for posttest score, there are some factors that could limit the generalizability of these findings. Firstly, this work uses data entirely from ASSISTments Skill Builder assignments. In these assignments, students are given a series of mathematics problems on the same skill, and are given immediate feedback on each problem as they complete it. Next problem correctness could be especially relevant in this context because the next problem is guaranteed to evaluate the same knowledge components as the previous problem. In assignments where interleaving [3] is used, the problem following an intervention could be only tangentially related to the problem for which the interven-

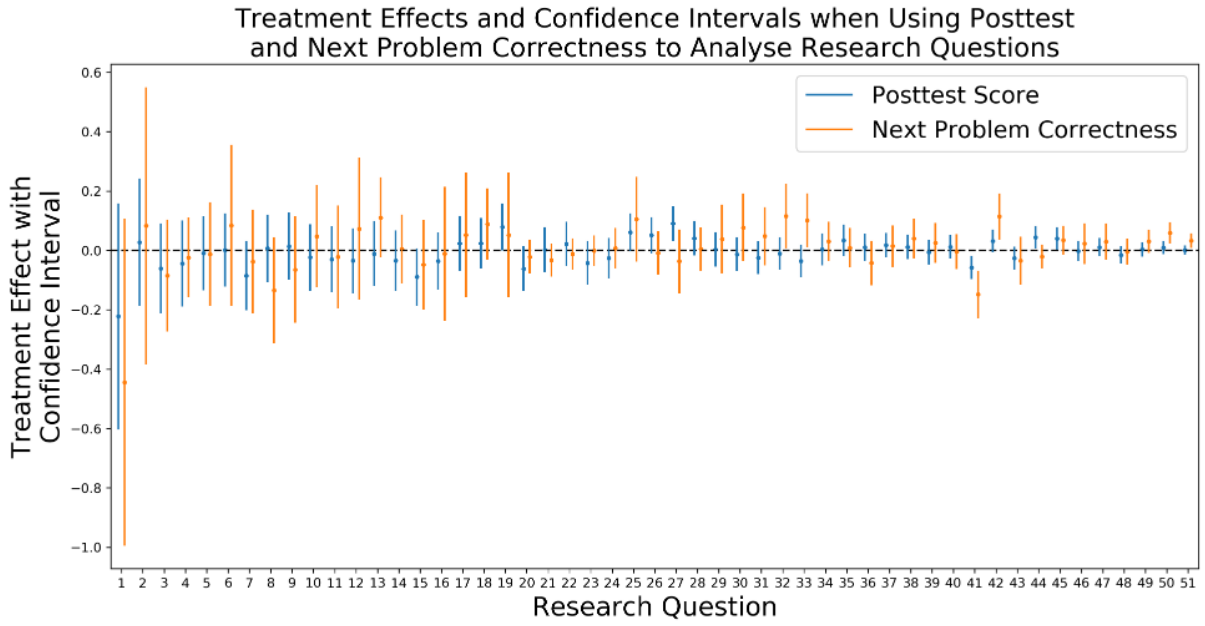


Figure 2: A plot of the treatment effect and confidence interval determined using posttest score and next problem correctness for each research question.

tion was provided, and thus a student's performance on the next problem would not be a good measure of the effectiveness of the intervention. In the future, using next problem correctness as a surrogate measure should be evaluated in other kinds of online learning environments, perhaps in contexts where the content students see is chosen adaptively. In this scenario, students will see different problems following an intervention, and combining the next problem correctness of multiple problems could have positive or negative effects on next problem correctness's value as a surrogate measure of learning.

Additionally, in this work, only 51 different research questions were used to evaluate the quality of different measures, with a total of 26,060 samples. It is possible that some of the model based attempts at creating a surrogate measure of learning would be more successful if given more data from a wider variety of situations in which A/B testing was performed. Having a larger and more diverse dataset to train the models from also opens up the possibility to train multiple specific models for different subgroups of users or experiments. With the limited data in this work, it was unlikely that splitting the data into subgroups would have helped any of the models. However, with more data it could be the case that a model trained on students from a specific socioeconomic background would be more effective at interpreting behaviors specific to those students. It could also be the case that training a model for a specific type of experiment, for example, experiments that alter the way in which students must answer the question as opposed to experiments that alter the support messages students receive, could improve the model's ability to pick up on different student behaviors associated with these different experiments. In the future, if more data becomes available, models trained on subgroups should be explored.

7. CONCLUSION

In this work, we attempted to derive and validate an effective surrogate measure of learning for use in online learning platforms where rapid A/B testing is used to compare problem-level instructional interventions at scale. To accomplish this, a variety of proximal surrogate measures for posttest score were created through feature engineering, regression, and deep learning. After evaluating each surrogate measure by ensuring it met the criteria for an associative surrogate as described in [11], students' next problem correctness was determined to be the best surrogate. When comparing the treatment effect on posttest score to the treatment effect on next problem correctness across 51 different research questions, both measures determined that approximately 10% of the research questions had statistically significant treatment effects, but both of the measures shared only one statistically significant finding. Although there was not much overlap in these significant findings, both measures agreed on which condition was most effective when they both found a significant treatment effect. Additionally, using next problem correctness as a measure lead to larger treatment effects with wider confidence intervals than using posttest score.

Follow-up work should be done to validate next problem correctness as a measure of learning in different domains and for different learning environments. Moving forward, using next problem correctness as a measure of learning within online learning platforms could be an effective way to evaluate students' progress and compare problem-level interventions to each other. We hope this work can help support the learning analytics community by providing a way to rapidly evaluate new instructional methods and interventions.

8. ACKNOWLEDGEMENTS

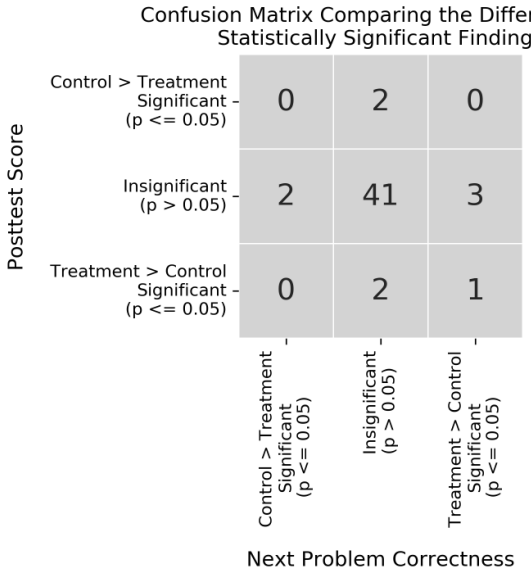


Figure 3: A confusion matrix comparing the differences in statistically significant findings when using posttest score and next problem correctness as measures of learning.

We would like to thank Anthony Botelho and Ben Hansen for their thoughtful advice and feedback on the early stages of this work. We would also like to thank NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, R305A120125), GAANN (e.g., P200A180088 P200A150306), EIR (U411B190024 S411B210024), ONR (N00014-18-1-2768), NHI (R44GM146483), and Schmidt Futures. None of the opinions expressed here are that of the funders.

9. REFERENCES

[1] R. S. Baker, N. Nasiar, W. Gong, and C. Porter. The impacts of learning analytics and a/b testing research: a case study in differential scientometrics. *International Journal of STEM Education*, 9(1):1–10, 2022.

[2] J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[3] S. K. Carpenter. Spacing and interleaving of study and practice. *Applying the science of learning in education: Infusing psychological science into the curriculum*, pages 131–141, 2014.

[4] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[5] C. for Research and J. H. U. Reform in Education. Evidence for essa, 2022.

[6] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[7] Google. Older unlisted content, 2022.

[8] B. Hannon. Predicting college success: The relative contributions of five social/personality factors, five cognitive/learning factors, and sat scores. *Journal of Education and Training Studies*, 2(4):46, 2014.

[9] N. T. Heffernan and C. L. Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[10] G. W. Imbens and D. B. Rubin. Rubin causal model. In *Microeconometrics*, pages 229–241. Springer, 2010.

[11] . G. T. Joffe, M. M. Related causal frameworks for surrogate outcomes. *Biometrics*, 65(2):530–538, 2009.

[12] K. S. Ostrow, D. Selent, Y. Wang, E. G. Van Inwegen, N. T. Heffernan, and J. J. Williams. The assessment of learning infrastructure (ali) the theory, practice, and scalability of automated assessment. In *Proceedings of the sixth international conference on learning analytics & knowledge*, pages 279–288, 2016.

[13] T. Patikorn and N. T. Heffernan. Effectiveness of crowd-sourcing on-demand assistance from teachers in online learning platforms. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pages 115–124, 2020.

[14] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*, 2009.

[15] E. Prihar, A. Haim, A. Sales, and N. Heffernan. Automatic interpretable personalized learning. In *Proceedings of the Ninth ACM Conference on Learning@ Scale*, pages 1–11, 2022.

[16] E. Prihar, A. Moore, and N. Heffernan. Identifying struggling students by comparing online tutor clickstreams. In *International Conference on Artificial Intelligence in Education*, pages 290–295. Springer, 2021.

[17] E. Prihar, T. Patikorn, A. Botelho, A. Sales, and N. Heffernan. Toward personalizing students' education with crowdsourced tutoring. In *Proceedings of the Eighth ACM Conference on Learning@ Scale*, pages 37–45, 2021.

[18] E. Prihar, M. Syed, K. Ostrow, S. Shaw, A. Sales, and N. Heffernan. Exploring common trends in online educational experiments. 2022.

[19] P. R. L. Surrogate endpoints in clinical trials: definition and operational criteria. *Stat Med*, 1989.

[20] J. Renz, D. Hoffmann, T. Staubitz, and C. Meinel. Using a/b testing in mooc environments. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 304–313, 2016.

[21] S. Ritter, A. Murphy, S. E. Fancsali, V. Fitkariwala, N. Patel, and J. D. Lomas. Upgrade: an open source tool to support a/b testing in educational software. In *Proceedings of the First Workshop on Educational A/B Testing at Scale (at Learning@ Scale 2020)*, 2020.

[22] A. Sales, A. Botelho, T. Patikorn, and N. T. Heffernan. Using big data to sharpen design-based inference in a/b tests. In *Proceedings of the Eleventh International Conference on Educational Data Mining*, 2018.

[23] A. C. Sales, E. Prihar, J. Gagnon-Bartsch, A. Gurung,

and N. T. Heffernan. More powerful a/b testing using auxiliary data and deep learning. In *International Conference on Artificial Intelligence in Education*, pages 524–527. Springer, 2022.

[24] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[25] B. L. Welch. The generalization of 'student's'problem when several different population varlances are involved. *Biometrika*, 34(1-2):28–35, 1947.

[26] R. Zwick and J. C. Sklar. Predicting college grades and degree completion using high school grades and sat scores: The role of student ethnicity and first language. *American Educational Research Journal*, 42(3):439–464, 2005.

Chapter 2
Tutoring Creation, Collection, and Analysis

All of the following papers have to do with analyzing the data in ASSISTments to discover where there are opportunities to collect more tutoring, creating tutoring, and determining what types of tutoring are most effective. This knowledge will help personalize learning by giving students the supports most effective for them.

The first paper, "**Toward Personalizing Students' Education with Crowdsourced Tutoring**", Evaluated the quality of crowdsourced student supports in ASSISTments. In this work I extracted the data from ASSISTments and performed the statistical analysis related to which content creators support is more effective. Identifying which content creators were most effective at helping students learn can be used to influence which creators are solicited for more content and to explore for specific features of the support that makes them more effective. This full paper was published at L@S 2021.

The second paper, "**A Novel Algorithm for Aggregating Crowdsourced Opinions**", provides a novel algorithm for ASSISTments to use when crowdsourcing opinions from teachers. In ASSISTments, teacher feedback is used to evaluate how similar different problems and supports are to each other. Effectively aggregating these opinions allows ASSISTments to establish an accurate understanding of content similarity, which can be used to share support between content, making more supports available within the platform. In this work I designed and evaluated the novel algorithm. A short version of this paper was accepted at EDM 2021.

The third paper "**Exploring Common Trends in Online Educational Experiments**", aggregates the data from over 50 experiments conducted within ASSISTments and reports on the common trends. These trends reveal what kinds of support are most effective at increasing students' learning, and can be used to direct the creation of future student supports. Additionally, the data collected for this work can be explored for opportunities to personalize learning. For this work, I analyzed the experiments, collected and aggregated their data, and performed the statistical analysis. This full paper was published at EDM 2022.

The fourth paper, "**Comparing Different Approaches to Generating Mathematics Explanations Using Large Language Models**", generated explanations of mathematics problems using different approaches that leveraged large language models, specifically GPT-3. Explanations were generated using few-shot learning with existing explanations, and by summarizing conversations between a tutor and a student. Ultimately, the explanations generated were not integrated into ASSISTments because they were not of high enough quality. However, the methodology in this work can be used with more advanced large language models in the future to create higher quality content. This content can then be used to personalize students' learning. This paper has been submitted to AIED 2023.

Chapter 2.1

Toward Personalizing Students' Education with Crowdsourced Tutoring

Toward Personalizing Students' Education with Crowdsourced Tutoring

Ethan Prihar
Worcester Polytechnic Institute
ebprihar@wpi.edu

Thanaporn Patikorn
Worcester Polytechnic Institute
tpatikorn@wpi.edu

Anthony Botelho
Worcester Polytechnic Institute
abotelho@wpi.edu

Adam Sales
Worcester Polytechnic Institute
asales@wpi.edu

Neil Heffernan
Worcester Polytechnic Institute
nth@wpi.edu

ABSTRACT

As more educators integrate their curricula with online learning, it is easier to crowdsource content from them. Crowdsourced tutoring has been proven to reliably increase students' next problem correctness. In this work, we confirmed the findings of a previous study in this area, with stronger confidence margins than previously, and revealed that only a portion of crowdsourced content creators had a reliable benefit to students. Furthermore, this work provides a method to rank content creators relative to each other, which was used to determine which content creators were most effective overall, and which content creators were most effective for specific groups of students. When exploring data from TeacherASSIST, a feature within the ASSISTments learning platform that crowdsources tutoring from teachers, we found that while overall this program provides a benefit to students, some teachers created more effective content than others. Despite this finding, we did not find evidence that the effectiveness of content reliably varied by student knowledge-level, suggesting that the content is unlikely suitable for personalizing instruction based on student knowledge alone. These findings are promising for the future of crowdsourced tutoring as they help provide a foundation for assessing the quality of crowdsourced content and investigating content for opportunities to personalize students' education.

Author Keywords

Online Tutoring; Crowd Sourcing; Statistical Analysis; Personalized Education

L@S'21, June 22–25, 2021, Virtual Event, Germany.

ACM ISBN 978-1-4503-8215-1/21/06.
https://doi.org/10.1145/3430895.3460130

CCS Concepts

•**Human-centered computing** → **Human computer interaction (HCI); Empirical studies in HCI;**

INTRODUCTION

The need for crowdsourcing within online learning platforms is growing as the user base of these platforms continues to expand and diversify [18, 7]. Crowdsourcing can be used effectively to generate new teaching materials [22] and new tutoring for students [18]. As more platforms integrate crowdsourcing, methods to evaluate and maintain the quality of crowdsourced materials need to be developed to ensure students receive a high quality education and effective support.

In the 2017-2018 academic year, ASSISTments, an online learning platform [10], deployed TeacherASSIST. TeacherASSIST allowed teachers to create tutoring in the form of hints and explanations for problems they assigned to their students. TeacherASSIST then redistributed teachers' tutoring to students outside of the their class. At L@S 2020, ASSISTments reported that teachers created about 40,000 new instances of tutoring for about 26,000 different problems. Through two large-scale randomized controlled experiments, it was determined that there was statistically significant improvement on the next problem correctness of students who received crowdsourced tutoring. Since the publication of these findings, ASSISTments has scaled up the distribution of crowdsourced content within the platform. The first part of this study uses new data, collected from the 2019-2020 and 2020-2021 school years to re-evaluate the findings of the original study and confirm that crowdsourced tutoring continues to benefit students overall.

The second part of this study investigated if there was a significant difference between the quality of different teachers' tutoring. The methodology used in this paper could be used in the future to determine which teacher's content should have priority when distributing tutoring to students in other classes.

Lastly, this study determined if there were any qualitative interactions between the teachers who created tutoring and

students grouped by their knowledge-level. Personalized learning requires qualitative interactions, defined as one group of students benefiting more from one type of instruction, while a different group of students benefited more from an alternative type of instruction. The learning science community has spent a considerable amount of time investigating the impact of personalized learning on students. While personalized tutoring based on prior knowledge has shown some evidence of a qualitative interaction [20], other methods for personalization, such as learning styles, have rarely shown conclusive evidence of a qualitative interaction [17]. The method used in this study can be used to search experimental data for qualitative interactions without using a randomized controlled trial to directly evaluate the presence of a particular qualitative interaction.

Specifically, this work seeks to address the following research questions:

1. Do the findings of the previous TeacherASSIST study still hold when tested on new data?
2. How did the effectiveness of teachers' tutoring compare to each other?
3. Was there any potential to personalize the tutoring students received based on their knowledge-level?

BACKGROUND

The Value of Crowdsourcing

The growing popularity of online learning platforms has created a greater opportunity and a greater need for educational materials of all levels. With a greater diversity of students, there arises the need to provide instruction to students of varying skill levels. Crowdsourcing can help diversify the available tutoring and assist in personalizing lesson plans for students [25, 3]. Crowdsourcing offers a mechanism to obtain the breadth of educational content required to meet the growing demand of online tutoring, but poses some challenges as well [25]. The biggest risk from using crowdsourced materials is the potential for low quality, or misleading material to negatively impact students [25]. Even if the information is high-quality, overly detailed tutoring, or tutoring from highly different sources can also have a negative impact of students' learning [23, 13, 12]. Ways to mitigate these risks include algorithmically evaluating the quality of crowdsourced content creators [21], or simply crowdsourcing content only from people that have been deemed qualified [16, 4, 24, 5].

Even with these risks, crowdsourcing has been a viable method for obtaining information on the knowledge components of different math problems [15], assisting students learning computer programming [2], and collecting videos explaining how to solve mathematics problems [26, 27]. Most directly, in the study preceding this work, tutoring messages created by teachers, for students completing work in ASSISTments, had an overall positive effect on students' learning [18]. Although crowdsourcing has shown promising results in many situations, there is a need to continue to evaluate the methods through which crowdsourced content is collected and validated so that as more educational platforms begin to incorporate crowdsourcing, they can do so efficiently, effectively, and without risk to students.

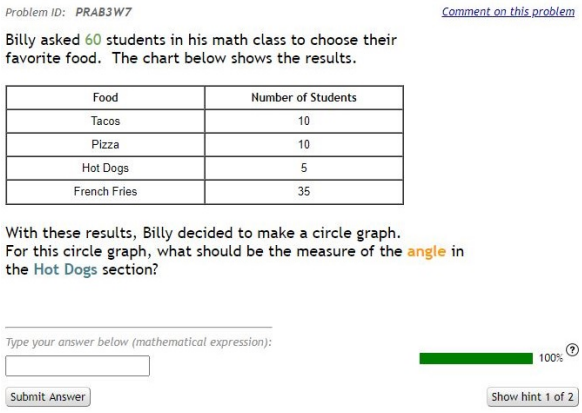


Figure 1. The ASSISTments tutor, as scene by a student solving a mathematics problem.

ASSISTments

The data used in this study comes from ASSISTments. ASSISTments[1] [11] is an online learning platform focused on empowering teachers via automating laborious tasks such as grading and record keeping of students, and providing insight to teachers on their class's common wrong answers and miss conceptions on assignments [11]. ASSISTments provides K-12 mathematics problems and assignments from multiple open source curricula for teachers to choose from and assign to their students. After an assignment has been assigned to students, students complete the assignment in the ASSISTments tutor, shown in Figure 1 [18]. In the tutor, students receive immediate feedback when they submit a response to a problem, which informs them if they are correct [9]. For some problems, students can request tutoring, which is available to them at any point during their completion of the problem, regardless of whether or not they have already attempted the problem. Tutoring comes in the form of hints, explaining how to solve parts of the problem, [11, 20], examples of how to solve similar problems [8, 14], examples of incorrect responses to problems with explanations of the error [14, 1], and full solutions to problems [27, 26]. Two examples of tutoring in ASSISTments are shown in 2 [18].

Recently ASSISTments began a program called TeacherASSIST, in which tutoring was crowdsourced from teachers in the form of written and video-recorded hints and explanations for solving middle-school math problems. ASSISTments collected tutoring created by teachers who had already used the platform for their own classrooms, and then provided the crowdsourced hints and explanations to students. Distributing these hints and explanations lead to a positive impact on students' learning [18]. In this study, the data released from the TeacherASSIST study [19], new data from TeacherASSIST collected since the publication of the previous study, and information on students' knowledge-level collected from the ASSISTments platform were used to investigate if any content creators' tutoring significantly out-performed other content creator's tutoring, as well as determine if there were any qualitative interactions between content creators and students.

[1] https://www.ASSISTments.org/

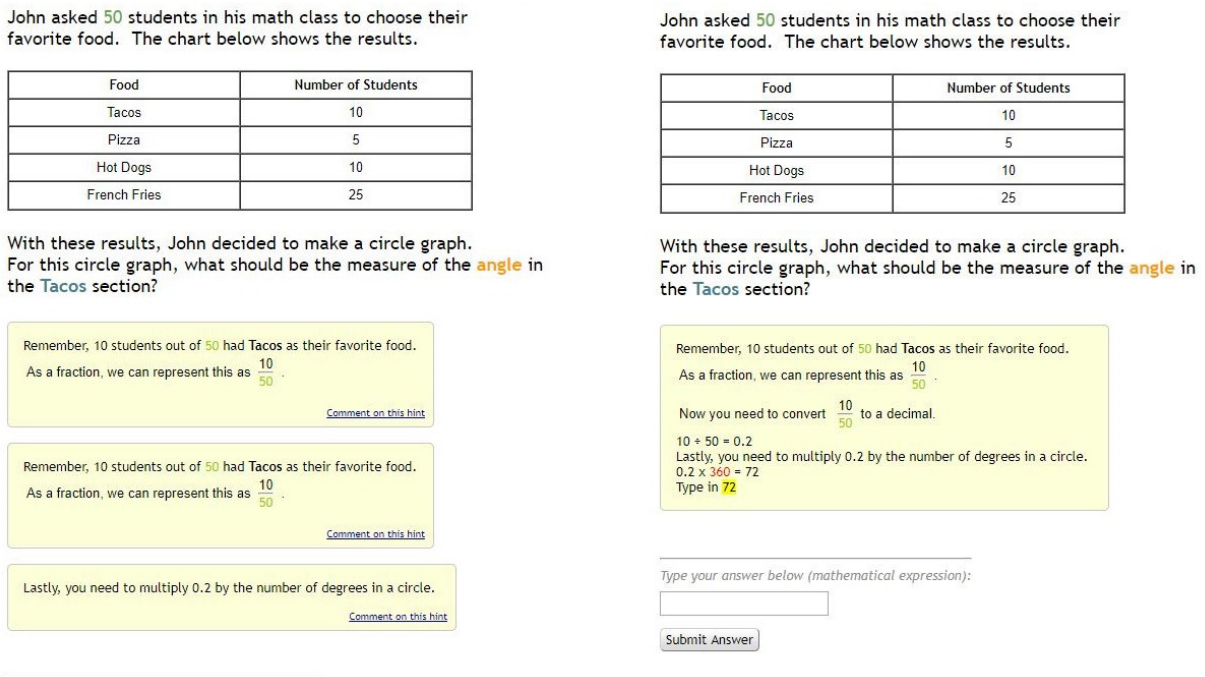


Figure 2. Two instances of tutoring in ASSISTments. On the left is a series of hints. On the right is a full explanation of how to solve the problem.

METHODOLOGY

Confirming the Previous Study's Findings

The same analysis performed in the original study [18] was repeated using the exact same code from the previous study made available by the Open Science Foundation [19]. New data, collected since the completion of the previous study up until February 2, 2021, was used to determine if the previously reported positive impact of TeacherASSIST was still present in a new academic year. The new dataset contained 6,774 unique problems, 7,059 unique tutoring messages, 18,420 unique students, and 500,900 answered problems. 50,426 of the answered problems were answered by students in the control condition, where they were not given the option to request tutoring, and 450,474 of the problems were answered by students in the intent-to-treat condition, in which they had the option to, but did not necessarily request tutoring. A majority of students were placed in the treatment condition because the previous study found the treatment condition to have a reliable positive effect, and ASSISTments did not want to prevent half the students from receiving beneficial crowdsourced tutoring. Of all the students in the new dataset, only 7.92% of them appeared in the initial study's data as well.

In order to gain more insight into how reliable the findings of the initial study were, a problem-level and student-level intent-to-treat analysis, in which the students were considered to be in the treatment condition if they were given the option to receive crowdsourced tutoring, regardless of whether or not they received it, and a treated analysis, where a student was considered to be in the treatment condition only if they received crowdsourced tutoring, were performed. For all of these analyses, which were all performed in the initial study, the Benjamini-Hochberg procedure was used to control the false discovery rate [6].

Measuring the Effectiveness of Teachers

To determine the effectiveness of each teacher, the data from the previous study and this study were combined and filtered such that only the instances where a student received no tutoring, or crowdsourced tutoring **for the first time**, and then immediately answered another problem remained. This step was necessary to remove compounding and extended exposure effects that would occur if students' next problem correctness was used to evaluate the quality of teacher's tutoring after students had seen tutoring from multiple teachers. Furthermore, any teachers whose tutoring was only seen by fewer than 30 students was excluded, as there was insufficient data to measure the effectiveness of these teachers. After data processing, 31,616 instances of a student getting one of 1,026 problems wrong, receiving tutoring from one of 11 different teachers, and then answering one of 1,308 different problems were used in the following analysis.

The filtered data was used to fit a regression which predicted next problem correctness based on the student, the problem the student got wrong, the teacher who wrote the tutoring that the student saw upon getting the problem wrong, and the next problem used to evaluate the quality of the tutoring. In addition to accounting for compounding and extended exposure effects, the students, and the problems they completed, were abstracted into sets of representative features. The features for students are shown in Table 1, and the features for problems are shown in Table 2. These features were used in the model instead of unique identifiers for each student and problem for two reasons. Primarily, using features to represent students and

Student Features
Total number of problems answered
Mean correctness on all completed problems
Mean time until first response on all completed problems
Mean time on task per problem
Mean number of attempts per problem

Table 1. Features used to abstract students while measuring the effectiveness of teacher's tutoring.

Problem Features
Type of problem, e.g., multiple choice, algebraic response
Mean correctness of all answers submitted for the problem
Mean time until first response for all students that answered the problem
Mean time on task of all answers submitted for the problem
Mean number of attempts of all answers submitted for the problem

Table 2. Features used to abstract problems while measuring the effectiveness of teacher's tutoring.

problems makes it easier to generalize this procedure to other data from different educational platforms. Secondly, given the large number of unique students and problems, a model trained to predict next problem correctness would likely over-fit and obtain very high accuracy by recognizing unique combinations of students and problems, rather than estimating correctness based on the teacher who created the tutoring given to the student, as intended.

Unlike the students and problems, teachers were not abstracted into representative features, as the goal of this process was to evaluate the effectiveness of the individual teachers, not the effectiveness of the different qualities of teachers. Teacher's unique identifiers were one-hot encoded for use in the model. In cases from the control condition, where students did not receive tutoring, all of the one-hot encoded teacher covariates equaled zero. By structuring the model's inputs this way, the coefficient of each teacher covariate measured how much more or less likely a student was to get the next problem correct after receiving tutoring from the corresponding teacher, and the probability of the null hypothesis for the covariate was the probability that receiving tutoring from the corresponding teacher was not better than receiving no tutoring at all. The probability of the null hypothesis was adjusted using the Benjamini-Hochberg procedure for controlling the false discovery rate [6] because each determination of the effectiveness of a teacher's tutoring was treated as a separate hypothesis. This model was used to determine which teachers' tutoring was statistically significantly better for students than receiving no tutoring.

Comparing the Effectiveness of Different Teachers

In addition to using the model from the previous section to evaluate the overall effectiveness of each teacher's tutoring, the model can also be used to compare teachers to each other. Comparing the coefficient of each teacher to determine which teacher's tutoring has a larger treatment effect is, alone, not enough to confirm that one teacher's tutoring is truly more effective that another teacher's tutoring, as the standard deviation of the difference between the teachers' effectiveness could be so large that the difference between the teachers' coefficients is statistically insignificant. However, using the variance-covariance matrix, the standard deviation of the difference between two teachers' coefficients can be calculated using Equation 1, where $var(T_x)$ is the variance of teacher x's coefficient, $var(T_y)$ is the variance of teacher y's coefficient, $cov(T_x, T_y)$ is the covariance of teacher x's and y's coefficient from the variance-covariance matrix, and δ is the standard deviation of the difference between teacher x's and y's coefficients. Then, if the difference in coefficients falls outside the 95% confidence interval, calculated using δ, it can be concluded that the teacher with a higher model coefficient created more effective tutoring than the teacher with a lower coefficient. This technique was used to create a map of teacher effectiveness, which could be used in the future to determine which teacher's tutoring should be given to struggling students.

$$\delta = \sqrt{var(T_x) + var(T_y) - cov(T_x, T_y)} \tag{1}$$

Measuring the Potential for Personalized Tutoring

The method described previously for comparing the effectiveness of different teacher's tutoring was also used to explore the data for opportunities for personalized tutoring. Personalizing the tutoring different groups of students receive based on the teacher that created the tutoring would only be justifiable, in this context, if three criteria are met:

1. One teacher's tutoring is more effective than another teacher's tutoring for one group of students. This can be determined using the method described in Section 3.3, using a model trained on only data from the students in the group.

2. The other teacher's tutoring is more effective for a separate group of students. This can also be determined using the method described in Section 3.3, using a model trained on only data from the other group of students.

3. Each teachers' tutoring is more effective than the control condition of receiving no tutoring for students in the group that benefits the most from the corresponding teacher. This can be determined using the method described in Section 3.2 on the data from only students in one group.

These criteria qualify the core assumption of personalized education, which is that in order for all students to attain the highest level of achievement they are capable of, different groups of students need to be provided with different content. If the above criteria are met, then in the future, personalizing student's educational content based on which teacher created the content would be justified. Otherwise, it would be more beneficial to give all students educational content from the teacher whose content led to the highest improvement in next problem correctness compared to the control condition. This work explored personalizing which teacher's tutoring a student received based on the knowledge-level of the student, determined by the students' average correctness.

Dependent Measure	Control Mean	Experiment Mean	t-Stat	p-Value
Correct First Try	0.65	0.66	-1.66	0.10
Requested Tutoring	0.20	0.19	2.61	**0.01**
Stop Out	0.01	0.01	1.08	0.28
Attempt Count	1.54	1.54	-0.74	0.46

Table 3. Problem-level paired t-test intention-to-treat analysis on student next-problem dependent variables. The number of unique problems = 5079.

Dependent Measure	Control Mean	Experiment Mean	t-Stat	p-Value
Correct First Try	0.63	0.64	-2.43	**0.02**
Requested Tutoring	0.20	0.20	3.22	**< 0.01**
Stop Out	0.01	0.01	-0.26	0.79
Attempt Count	1.59	1.59	0.52	0.60

Table 4. Student-level paired t-test intention-to-treat analysis on student next problem dependent variables. The number of unique students = 10340.

Dependent Measure	Control Mean	Experiment Mean	t-Stat	p-Value
Correct First Try	0.33	0.35	-3.09	**<0.01**
Requested Tutoring	0.55	0.51	5.10	**< 0.01**
Stop Out	0.02	0.02	-0.49	0.62
Attempt Count	1.85	1.86	-0.23	0.82

Table 5. Problem-level paired t-test treated analysis on student next problem dependent variables. The number of unique problems = 2524.

Dependent Measure	Control Mean	Experiment Mean	t-Stat	p-Value
Correct First Try	0.36	0.40	-4.27	**<0.01**
Requested Tutoring	0.51	0.46	5.70	**< 0.01**
Stop Out	0.02	0.02	-0.94	0.35
Attempt Count	1.93	1.86	2.54	**0.01**

Table 6. Student-level paired t-test treated analysis on student next problem dependent variables. The number of unique students = 3547.

RESULTS

The Effectiveness of Crowdsourcing

The results of this replication of the previous study showed the same positive findings as the previous study, but with better confidence. Specifically, students who received TeacherASSIST tutoring were more likely to be able to solve the next problem correctly on their first try than students in the control condition. When students who received tutoring did not succeed on their first attempt, they were not more likely to give up or submit many more wrong answers, and they were more likely to be able to eventually solve the problem without requesting more tutoring. With this new, larger dataset, the effect on the treated is large enough to be detected with significance in the intention-to-treat analysis. Tables 3 and 4 show the results of the problem-level and student-level intention-to-treat analysis respectively, and tables 5 and 6 show the results of the problem-level and student-level treated analysis respectively. Correct first try measures the difference in next problem correctness, requested tutoring measures the difference in how much tutoring students' requested on the next problem after receiving tutoring from TeacherASSIST, Stop Out measures the difference in students' completion of the next problem, and Attempt Count measures the difference in how many attempts students' took to answer the next problem following the tutoring they received from TeacherASSIST. The bold p-values are the significant values after correcting for multiple hypothesis testing with the Benjamini-Hochberg procedure [6]. These findings confirm the previous study's conclusion that TeacherASSIST has an overall positive effect on students' learning.

Measuring the Effectiveness of Teachers

Using the method described in Section 3.2. The next problem correctness of students after receiving a teacher's tutoring was compared to receiving no tutoring. A coefficient measuring the impact of each teacher's tutoring on students' next problem correctness, a p-value denoting the probability that this coefficient is statistically equivalent to a null treatment effect, and the total number of students who viewed the tutoring from each teacher were calculated and are shown in Table 7. If a teacher's row is bold, this indicates that their tutoring had a statistically significant impact on next problem correctness after adjusting for multiple hypothesis testing.

Interestingly, even though receiving crowdsourced tutoring had an overall positive effect on students' next problem correctness, only four of the 11 teachers' tutoring had a statistically significant positive effect. Additionally, one teacher's tutoring had a statistically significantly negative impact on student's next problem correctness. This demonstrates a potential benefit to evaluating the quality of each content creator's tutoring as it is not necessarily the case that when crowdsourced content is overall beneficial, each content creator by themselves is providing a benefit. In the future of TeacherASSIST, and in other crowdsourcing endeavors, only distributing content from teachers whose tutoring has a reliable positive effect, and tutoring from teachers whose tutoring is still of ambiguous benefit, would likely lead to higher next problem correctness for students.

Teacher ID	View Count	Coefficient	p-Value
No Tutoring	2,289		
A	95	0.0629	0.112
B	222	**-0.0724**	**0.044**
C	11,202	0.0147	0.118
D	5,340	**0.0301**	**0.005**
E	76	0.0573	0.189
F	3,671	**0.0449**	**< 0.001**
G	5,763	**0.0271**	**0.008**
H	911	**0.0396**	**0.007**
I	1,452	-0.0184	0.197
J	544	0.0046	0.819
K	51	-0.0061	0.914

Table 7. The impact, statistical significance, and view count of each teacher's tutoring on students' next problem correctness.

This evaluation of teachers' effectiveness could also be used as professional development for the teachers themselves. If a teacher's tutoring is not leading to a statistically significant increase in students' next problem correctness, the crowdsourcing platform could alert these teachers that their tutoring could use improvement and provide them with examples of other teacher's tutoring that had been shown to be effective. Then, after the teacher updates their tutoring, the platform could re-evaluate their effectiveness and report back to the teacher. This interaction with teachers could also encourage teachers that are creating highly effective tutoring to create more tutoring by reporting how many students have received their tutoring, and to what extent their tutoring has helped students beyond their classroom.

Comparing the Effectiveness of Different Teachers

Using the method described in Section 3.3, the effectiveness of each teacher's tutoring was compared to every other teacher's tutoring. Figure 3 Shows the instances, in green, when the tutoring from the teacher labeled on the row, was more effective than the tutoring from the teacher labeled on the column. A grey cell indicates that the row teacher did not create more effective tutoring than the column teacher. For clarity, the teachers were sorted by how many other teachers their tutoring was more effective than. If all the teachers could be put in order from most to least effective tutoring, then Figure 3 would have entirely green cells above the diagonal. However, this is clearly not the case. Due to the variance in the effectiveness of teachers' tutoring, no teacher's tutoring is significantly better or worse than every other teachers' tutoring.

Figure 3 shows some clear examples of teachers whose tutoring is more effective than some of the other teachers' tutoring, for example, teacher F, and teachers whose tutoring is less effective than most other teachers' tutoring, for example, teacher B. Figure 3 also shows examples of teacher's whose variance in the effectiveness of their tutoring is very high, for example, teacher K. This high variance results in no teacher significantly outperforming teacher K's tutoring, and teacher K's tutoring not significantly outperforming any other teacher's tutoring. Teacher K demonstrates the need to take into account the variance of the difference between teachers' effectiveness. One cannot assert that one teacher's tutoring is more effective than another teacher's tutoring using the model coefficients alone.

Comparing teacher's tutoring can be used to choose between potential tutoring for students when more than one option is available, but care must be taken, if implementing this at scale, to not ignore tutoring from content creators with high variance in the effectiveness of their tutoring. It could be that these content creators are new to the platform, and have either created only a few instances of tutoring, or their tutoring has not had a lot of exposure yet. Content creators with high variance should be given the benefit of the doubt, and only when a teacher's tutoring is statistically significantly better than another teacher's tutoring should the more effective tutoring be chosen for the student. When using this model to select which tutoring to give the student, the student's next problem correctness should not be included in any statistical analysis that relies on random sampling.

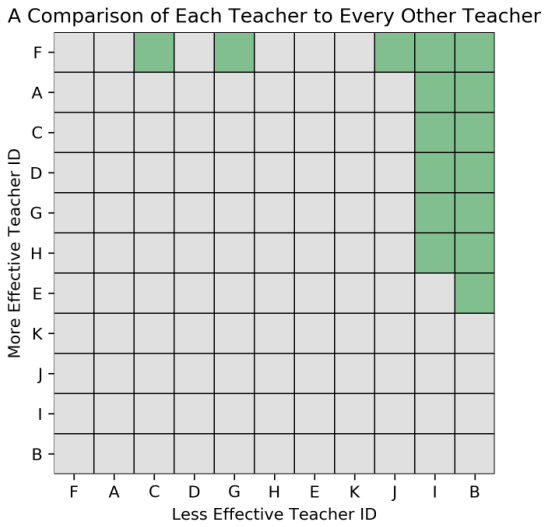


Figure 3. A map comparing the effectiveness of different teachers' tutoring.

Teacher's could also benefit from a platform that compares their effectiveness to other teachers. For professional development, teachers could be paired with a mentor and mentee. The mentor would be a teacher with statistically better tutoring than them, and the mentee would be a teacher with statistically worse tutoring than them. This would give teachers the opportunity to learn and teach others, and garner community support for the platform. Top performers could be rewarded with notoriety within the platform, and encouraged to continue to make content. Considering how heavily crowdsourcing relies on user engagement, working the analysis of teachers' effectiveness into different methods of engaging existing users and drawing in new users is an important step in the crowdsourcing process.

Measuring the Potential for Personalized Tutoring

Lastly, using the method described in Section 3.4, it was investigated if personalizing which teacher's tutoring students received based on students' knowledge-levels would likely have had a positive impact on students' next problem correctness. To group students by knowledge-level, the data was split into two datasets, The high-knowledge student data contained 18,139 instances of students whose average correctness was above average and the low-knowledge student data contained 13,475 instances of students whose average correctness was below average. To determine which teachers met Criteria 1 and 2 from Section 3.4: one teacher's tutoring is more effective than another teacher's tutoring for one group of students, and, the other teacher's tutoring is more effective for a separate group of students, the same method used in Section 3.3 was used on each group of students. The results of these comparisons are shown in Figure 4. Figure 4 shows that there is no evidence to support the claim that personalizing the tutoring students received would have led to an increase in next problem correctness. While some teachers, like teacher E, were very effective for low-knowledge students, and some teacher,

like teacher B, were particularly ineffective for high knowledge students, there were no teachers that met Criteria 2 and 3, in other words, the same teacher's tutoring was likely to have the highest positive impact on all students' next problem correctness regardless of the student's knowledge-level.

This rigorous process used to determine if there is truly a benefit to personalized tutoring could be used for more than just determining if student's tutoring can be personalized based on their knowledge-level and who created the tutoring. This process could be used on a per-problem basis. For each problem, an analysis could be performed to evaluate which of the available crowdsourced tutoring messages would be most likely to positively impact students' next problem correctness based on traits of the students. Doing this analysis on a per-problem basis would require much more data, but as platforms expand and curricula increase their integration with online learning, this may become a viable option. Additionally, if socioeconomic and demographic information on students is available, then this process could be used to personalize tutoring for students based on their gender or race. It is particularly important to pay attention to how personalization effects minority students. If the effectiveness of whatever intervention being deployed is being measured by how it effects all students on average, then in the same way that this study found that crowdsourced tutoring was overall beneficial, but some teacher's tutoring had a negative impact on next problem correctness, an intervention may be beneficial overall, but also be detrimental to minority students. Being aware of how each group of students is effected by an intervention will allow researchers to maintain fair interventions that help all students achieve their full potential.

LIMITATIONS AND FUTURE WORK

Although the results of this study are promising, there are limitations to this work. In order to compare teachers' tutoring, students and problems had to be represented with features. While these features adequately modeled students and problems well enough to account for the variations in problem difficulty and student performance, these features are not necessarily the best features to use. The features used in our models could only predict next problem correctness with an ROC AUC of 0.71. It is unlikely that the features we had available captured 100% of the variance in problems and students, and therefore including more, or different features for problems and students could increase the reliability in the measurements of the effectiveness of teacher's tutoring by increasing the model's accuracy.

In addition to potential improvements to the student and problem features, features for teachers could also be used to group teachers similar to how students were grouped in Section 4.4. Features of teachers could be used to investigate if certain groups of teachers tend to outperform other groups and could be used for personalization similarly to how individual teachers were compared in this work. Additionally, if certain features were indicative of a teacher's ability to create particularly effective tutoring, this information could be used to advise teachers and other content creators.

In this work, statistical analysis was used to determine which teachers' tutoring was most effective. While this method could be used to select which tutoring to provide to students based on which teacher is overall most effective, an online learning platform could also use reinforcement learning to select which of multiple instances of tutoring to provide to a student based on the same features of problems and students used in this work. Contextual bandit algorithms [28] use context, which in this case are features of students and problems, to take one of multiple actions, which in this case are the actions of providing one of many different instances of tutoring to a student. Then they receive a reward, which in this case would be the student's next problem correctness, and adjust their decision making process to take the action that is most likely to lead to the highest reward. While using a contextual bandit algorithm prevents one from doing the same kind of experimental analysis performed in this work, it provides a method to algorithmically determine and offer the best tutoring available to students.

Although no conclusive evidence of qualitative interactions between teachers' tutoring and students knowledge were found in this work, the potential for personalized learning should continue to be explored. More specific or alternative student features could be created evaluated for qualitative interactions the same way that knowledge-level was used in this work. It is possible that even within the dataset used in this work, there are qualitative interactions between groups of students that were not able to be considered. For example, this work had no knowledge of students' state test scores, home environments, demographic information, or socioeconomic status. All of these factors could influence what tutoring is most effective for each student and reveal the opportunity to personalize students' education.

CONCLUSION

In this follow up study, providing tutoring through TeacherASSIST continued to reliably increase students' next problem correctness, an indication that crowdsourced tutoring within the ASSISTments platform has a positive impact on students' learning. Due to many schools' recent transition to partially or fully remote learning, more data was available this year than in previous years, which allowed this study to find a reliably positive effect on students' learning even in an intent-to-treat analysis, where not every student chose to view the tutoring available to them. Furthermore, when investigating the impact of each teacher's tutoring separately, only four of the 11 teachers had a reliably positive impact on students, and one teachers' tutoring had a reliably negative impact. This finding could be used in the future to select which teacher's tutoring to provide to students based on how reliable a teachers' tutoring has been in the past. As online tutoring platforms grow and continue to incorporate crowdsourcing techniques, it will be important to include metrics for evaluating the quality of crowdsourced materials and the means to algorithmically select the most effective content. As the corpus of crowdsourced tutoring grows, the most effective content can also be explored for similarities to each other. Empirically evaluating what makes tutoring effective has the potential to improve current methods for creating tutoring, and enhance existing pedagogy.

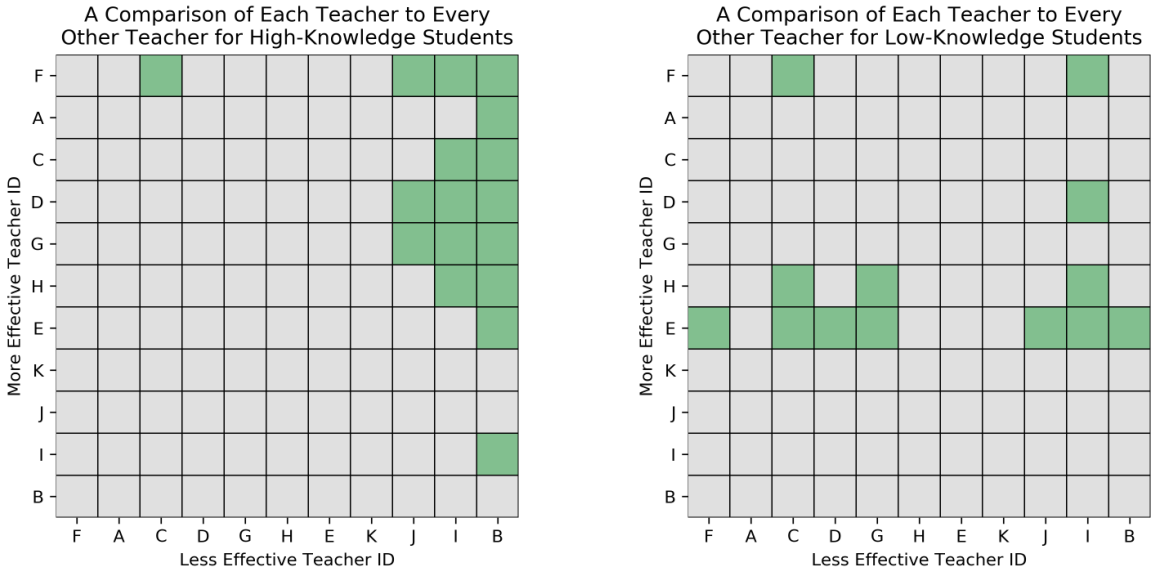


Figure 4. A map comparing the effectiveness of different teachers' tutoring separately for high and low knowledge students.

Although no evidence of the benefit of personalized education was found in this study, there is still the potential for other qualities of tutoring and the students that receive the tutoring to have an impact on what kind of tutoring is most effective. Future work can explore for more opportunities to personalize students' education using the same method in this study, or look to contextual bandit algorithms to find opportunities for personalization. Through continued efforts, crowdsourcing has the potential to advance pedagogy and provide students with a more equitable education.

ACKNOWLEDGMENTS

We would like to thank multiple NSF grants (e.g., 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, 1535428, 1440753, 1316736, 1252297, 1109483, & DRL-1031398), as well as the US Department of Education for three different funding lines; a) the Institute for Education Sciences (e.g., IES R305A170137, R305A170243, R305A180401, R305A120125, R305A180401, & R305C100024), b) the Graduate Assistance in Areas of National Need program (e.g., P200A180088 & P200A150306), and c) the EIR. We also thank the Office of Naval Research (N00014-18-1-2768), Schmidt Futures, and an anonymous philanthropic foundation.

REFERENCES

[1] Deanne M Adams, Bruce M McLaren, Kelley Durkin, Richard E Mayer, Bethany Rittle-Johnson, Seiji Isotani, and Martin Van Velsen. 2014. Using erroneous examples to improve mathematics learning with a web-based tutoring system. *Computers in Human Behavior* 36 (2014), 401–411.

[2] Dhiya Al-Jumeily, Abir Hussain, Mohammed Alghamdi, Chelsea Dobbins, and Jan Lunn. 2015. Educational crowdsourcing to support the learning of computer programming. *Research and practice in technology enhanced learning* 10, 1 (2015), 1–15.

[3] Carlos Eduardo Barbosa, Vanessa Janni Epelbaum, Marcio Antelio, Jonice Oliveira, and Jano Moreira de Souza. 2013. Crowdsourcing environments in E-learning scenario: A classification based on educational and collaboration criteria. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 687–692.

[4] Tiffany Barnes and John Stamper. 2008. Toward automatic hint generation for logic proof tutoring using historical student data. In *International Conference on Intelligent Tutoring Systems*. Springer, 373–382.

[5] Tiffany Barnes and John Stamper. 2010. Automatic hint generation for logic proof tutoring using historical data. *Journal of Educational Technology & Society* 13, 1 (2010), 3.

[6] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57, 1 (1995), 289–300.

[7] Anthony F Botelho and Neil T Heffernan. 2019. Crowdsourcing Feedback to Support Teachers and Students. *Design Recommendations for Intelligent Tutoring Systems* 7 (2019), 101–108.

[8] Tessa HS Eysink, Ton de Jong, Kirsten Berthold, Bas Kolloffel, Maria Opfermann, and Pieter Wouters. 2009. Learner performance in multimedia learning arrangements: An analysis across instructional approaches. (2009).

[9] Mingyu Feng and Neil T Heffernan. 2006. Informing teachers live about student learning: Reporting in the assistment system. *Technology Instruction Cognition and Learning* 3, 1/2 (2006), 63.

[10] Neil T Heffernan and Cristina Lindquist Heffernan. 2014a. The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education* 24, 4 (2014), 470–497.

[11] Neil T Heffernan and Cristina Lindquist Heffernan. 2014b. The ASSISTments Ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education* 24, 4 (2014), 470–497.

[12] Raymond W Kulhavy, Mary T White, Bruce W Topp, Ann L Chan, and James Adams. 1985. Feedback complexity and corrective efficiency. *Contemporary educational psychology* 10, 3 (1985), 285–291.

[13] Mary Ellen Lepionka. 2008. *Writing and developing your college textbook: a comprehensive guide to textbook authorship and higher education publishing*. Atlantic Path Publishing.

[14] Bruce M McLaren, Tamara van Gog, Craig Ganoe, Michael Karabinos, and David Yaron. 2016. The efficiency of worked examples compared to erroneous examples, tutored problem solving, and problem solving in computer-based learning environments. *Computers in Human Behavior* 55 (2016), 87–99.

[15] Steven Moore, Huy A Nguyen, and John Stamper. 2020. Evaluating Crowdsourcing and Topic Modeling in Generating Knowledge Components from Explanations. In *International Conference on Artificial Intelligence in Education*. Springer, 398–410.

[16] Korinn Ostrow and Neil Heffernan. 2014. Testing the multimedia principle in the real world: a comparison of video vs. Text feedback in authentic middle school math assignments. In *Educational Data Mining 2014*.

[17] Harold Pashler, Mark McDaniel, Doug Rohrer, and Robert Bjork. 2008. Learning styles: Concepts and evidence. *Psychological science in the public interest* 9, 3 (2008), 105–119.

[18] Thanaporn Patikorn and Neil T Heffernan. 2020a. Effectiveness of Crowd-Sourcing On-Demand Assistance from Teachers in Online Learning Platforms. In *Proceedings of the Seventh ACM Conference on Learning at Scale*. 115–124.

[19] Thanaporn Patikorn and Neil T. Heffernan. 2020b. Release of TeacherASSIST Dataset #1. (2020). DOI: http://dx.doi.org/10.17605/OSF.IO/EGP5F Accessed: 2020-05-15.

[20] Leena M Razzaq and Neil T Heffernan. 2009. To Tutor or Not to Tutor: That is the Question.. In *AIED*. 457–464.

[21] Paul Ruvolo, Jacob Whitehill, and Javier R Movellan. 2013. Exploiting commonality and interaction effects in crowdsourcing tasks using latent factor models. In *Neural Information Processing Systems. Workshop on Crowdsourcing: Theory, Algorithms and Applications*. Citeseer.

[22] Catharyn C Shelton and Leanna M Archambault. 2019. Who are online teacherpreneurs and what do they do? A survey of content creators on TeachersPayTeachers. com. *Journal of Research on Technology in Education* 51, 4 (2019), 398–414.

[23] Valerie J Shute. 2008. Focus on formative feedback. *Review of educational research* 78, 1 (2008), 153–189.

[24] John Stamper, Michael Eagle, Tiffany Barnes, and Marvin Croy. 2013. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education* 22, 1-2 (2013), 3–17.

[25] Daniel S Weld, Eytan Adar, Lydia B Chilton, Raphael Hoffmann, Eric Horvitz, Mitchell Koch, James A Landay, Christopher H Lin, and Mausam Mausam. 2012. Personalized Online Education-A Crowdsourcing Challenge.. In *HCOMP@ AAAI*. Citeseer.

[26] Jacob Whitehill and Margo Seltzer. 2017. A Crowdsourcing Approach to Collecting Tutorial Videos–Toward Personalized Learning-at-Scale. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*. ACM, 157–160.

[27] Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. 2016. Axis: Generating explanations at scale with learnersourcing and machine learning. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*. ACM, 379–388.

[28] Li Zhou. 2015. A survey on contextual multi-armed bandits. *arXiv preprint arXiv:1508.03326* (2015).

Chapter 2.2

A Novel Algorithm for Aggregating Crowdsourced Opinions

A Novel Algorithm for Aggregating Crowdsourced Opinions

Ethan Prihar, Neil Heffernan
Worcester Polytechnic Institute
ebprihar@wpi.edu, nth@wpi.edu

ABSTRACT

Similar content has tremendous utility in classroom and online learning environments. For example, similar content can be used to combat cheating, track students' learning over time, and model students' latent knowledge. These different use cases for similar content all rely on different notions of similarity, which make it difficult to determine contents' similarities. Crowdsourcing is an effective way to identify similar content in a variety of situations by providing workers with guidelines on how to identify similar content for a particular use case. However, crowdsourced opinions are rarely homogeneous and therefore must be aggregated into what is most likely the truth. This work presents the Dynamically Weighted Majority Vote (DWMV) method. A novel algorithm that combines aggregating workers' crowdsourced opinions with estimating the reliability of each worker. The DWMV method was compared to the traditional majority vote method in both a simulation study and an empirical study, in which opinions on seventh grade mathematics problems' similarity were crowdsourced from middle school math teachers and college students. In both the simulation and the empirical study the DWMV method outperformed the traditional majority vote method, suggesting that DWMV should be used instead of majority vote in future crowdsourcing endeavors.

Keywords

Crowdsourcing, Similarity, Community Detection, Hierarchical Clustering

1. INTRODUCTION

Within online learning platforms and intelligent tutoring systems there is a tremendous opportunity to utilize knowledge of content similarity. Similar problems can help prevent cheating during exams by randomly selecting from multiple similar problems when students receive the exam, measure students' learning gains by spreading out similar problems between assignments, and measure the effects of instructional interventions by comparing a student's scores on similar problems before and after the intervention. Similar instructional material can be used to offer students choices in which instructional material they receive, which has been shown to increase engagement and achievement [22]. While it is possible to implement these methods with general knowledge of content similarity, such as similarity in prerequisite knowledge or difficulty, if a more informed definition of content similarity is used, the success of these methods is likely to grow.

Although there is a lot of value in knowing what content is similar to other content, what content should be considered similar is highly dependent on use case. This makes it a challenge for content creators to define the similarity in the content, as they don't necessarily know what their content will be used for. While some content is obviously similar, for example, two mathematics problems that are identical except for the numbers used in the problems, in other situations it is much more difficult, especially when content is being aggregated from multiple sources that may not even use the same metrics for prerequisite knowledge or difficulty.

Crowdsourcing offers a way to derive which content is similar to other content for specific use cases. Crowdsourced opinions on similar content can be gathered each time a new use case for similar content arises. By informing the workers, whose opinions are being crowdsourced, of the specific use case and requirements for similarity, the methods that rely on content being similar are more likely to be successful. However, crowdsourcing opinions on similar content poses some challenges as well. Before an online learning platform or intelligent tutoring system uses crowdsourced assertions of similarity, steps must be taken to assess the trustworthiness of workers whose opinions are being crowdsourced and ensure the truthfulness of the final assertions of similarity.

In this work we present a novel algorithm that both measures the reliability of the workers whose opinions are being crowdsourced, and determines, from these individual's opinions, what content is most likely to be similar to other content. To evaluate this method, we first simulated a wide range of conditions in which assertions of similarity were made, and compared the performance of our algorithm to the traditional alternative. We then performed a case study where teachers and college students were told to identify middle-school mathematics problems that evaluated a similar skill set. The assertions of similarity collected from the

case study were used to identify groups of similar problems and measure the reliability of each worker's assertions.

Ultimately, this work seeks to answer the following three research questions:

1. Can we exploit properties of community detection to more accurately form groups of content from crowdsourced opinions?

2. How does the resulting algorithm perform in a simulation study compared to the more traditional method?

3. How does the resulting algorithm perform in a case study using workers of various expertise to determine which mathematics problems are similar to each other?

2. BACKGROUND

2.1 The Value of Similar Content

Similar content has been used in online and in-person learning environments to increase the engagement and learning gains of students through many different means. For some students, giving them the option to choose what assignment they want to complete from a set of similar assignments led to an increase in their assignment completion rate and learning gains [10, 22]. Furthermore, having similar content available allows students' education to be distributed over time. Spacing out the time between when students are tested on similar material has been shown to have a positive effect on students' learning and retention of the content [1, 4, 13, 14, 24]. Educators and online platforms have also used similar problems to prevent cheating. Using similar, but different problems on both online and in person exams is a well established method for preventing students from cheating off of each others work, or off of easily accessible online explanations [5, 7].

In addition to increasing student engagement, similar content can be used to estimate students' knowledge. Knowledge Tracing and Performance Factor Analysis both use students' correctness on problems to estimate their latent knowledge [2, 19]. Knowledge Tracing and Performance Factor Analysis have been used in online learning platforms where the problems are tagged by which skills are required to solve them, providing a metric by which to measure similarity [18, 20]. Using skill tags as the only metric for similarity poses some issues for knowledge tracing, which assumes each problem can be represented by a single knowledge component [8, 19]. If some skill tags encompass multiple knowledge components, these methods could misrepresent students' latent knowledge. These models of student knowledge could benefit from a more refined notion of similarity for their specific use case.

2.2 Ensembling Crowdsourced Opinions

Identifying the truth from crowdsourced opinions is not a new problem. Most of the techniques employed to ensure the accuracy of crowdsourced opinions rely on ensuring that workers have sufficient knowledge of the subject matter. This can be done through testing workers before giving them tasks, tailoring tasks specific to their skill sets, recruiting high quality workers, and educating workers before assigning them tasks. This can also be done through encouragement with extrinsic motivators like money, promotions, or prizes, or intrinsic motivators like a sense of purpose, or by gamifying the crowdsourcing tasks [3].

While there are many methods to encourage individuals whose opinions are being crowdsourced to be accurate, this work is focused on how to validate the quality of individuals' opinions after their task is complete. Current methods for accomplishing this place the burden of validation back onto the workers. Having workers rank the quality of other workers assertions is one method of validation. Another common method for validation is to have multiple workers perform the same task and merge the output of each worker, either as an average or as a majority vote [3].

There are also more advanced ways of algorithmically validating crowdsourced opinions. Item response theory and latent factor analysis based models have out-performed majority voting based validation methods on tasks related to identifying facial expressions and answering questions about geography [21, 26]. These models also determine the quality of individuals whose opinions are being crowdsourced, which can be used to refine the pool of individuals used for future crowdsourcing tasks [21, 26]. The novel algorithm in this work also aggregates crowdsourced opinions while evaluating the quality of each worker.

2.3 Community Detection

The field of community detection is focused around determining groups of similar items from a network of connected items. This has many applications throughout mathematics, physics, biology, computer science, and social sciences. Many things can represented as a network, for example, interstellar objects, neurons, city streets, and social media can all be represented as networks of interconnected items [6, 9, 15]. Finding similar educational content can be framed as a community detection problem by representing educational content as a network in which items are connected by topic, difficulty, language, prerequisite knowledge, or, in the case of this work, opinions on similarity. Structuring the task of identifying similar educational content as a community detection problem allows for the use of various well-established community detection algorithms.

2.3.1 Hierarchical Clustering

A common community detection algorithm, and the one used in this study, is the hierarchical clustering method. In hierarchical clustering, each item begins in it's own cluster. Then, clusters are merged based on the merge strategy and distance between clusters [12, 17]. Both the merge strategy and distance metric are hyper-parameters of the model that must be chosen. In order to provide an understanding of how a merge strategy works, Figure 1 helps illustrate the difference between two merge strategies: Complete Linkage and Single Linkage. Complete Linkage calculates the distance between the furthest points of each cluster. In Figure 1, lines A, B, and C represent these distances. Single Linkage calculates the distance between the closes points of each cluster. In Figure 1, lines D, E, and F represent these distances. After these distances are calculated, both methods merge the clusters with the shortest distance between points.

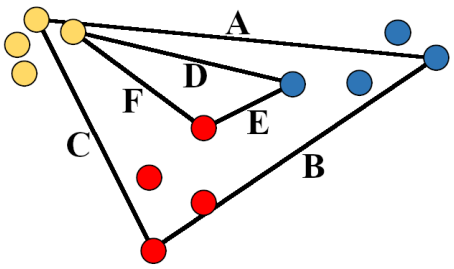


Figure 1: An example of Single Linkage and Complete Linkage merge strategies for hierarchical clustering.

In Figure 1, the red and yellow clusters would be merged if Complete Linkage was used, and the red and blue clusters would be merged if Single Linkage was used. Calculating distances and merging clusters is repeated until either a desired number of clusters is reached or the distances between clusters are all above a specified threshold [23]. The former stopping criteria requires the number of clusters to be specified, while latter stopping criteria requires the maximum distance between mergeable clusters to be specified. In Figure 1, euclidean distance was used to illustrate the different merge strategies, but this does not have to be the case. Any similarity metric, for example, Euclidean Distance or Manhattan Distance, or if the data is binary, Jaccard Distance or Dice Distance, can be used by hierarchical clustering methods.

3. METHODOLOGY

3.1 Dynamically Weighted Majority Vote

The Dynamically Weighted Majority Vote (DWMV) method is our alternative to the traditional majority vote method for combining multiple crowdsourced opinions. The DWMV method calculates the weighted majority opinion for each task, then determines the weight of each worker by how closely their opinion agreed with the majority opinion. The closeness of a worker's opinion to the majority opinion can be determined with something such as Precision or Recall [23] for tasks with binary output, or Mahalanobis Distance [16] for tasks with continuous outputs. DWMV initializes all workers' weights to be equal at the beginning of the algorithm, and iteratively updates these weights until the weighted majority vote does not change between iterations. Once the weighted majority vote remains constant from one iteration to the next, the weights of the workers can be interpreted as a measure of confidence in each worker, and the final weighted majority vote can be used downstream in the same way the traditional majority vote would have been used.

Using a simulated example to illustrate how DWMV works, imagine that 40 workers with random error rates were each asked to answer 900 questions in which their answer could only be 1 or 0. Figure 2 shows the true answers to each question in blue. The orange line is the weighted average of each simulated worker's responses during the first iteration of DWMV, while all workers have equal weight. The

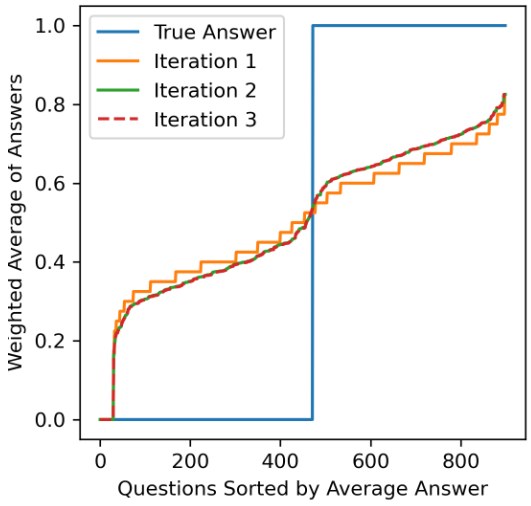


Figure 2: Progressive iterations of DWMV converging toward the truth.

green line is the weighted average of each worker's responses during the second iteration of DWMV. The first iteration determined the weight of each worker as the accuracy of their responses compared to the majority vote for each question, and as a result, the new weighted average answer for each question more closely approximates the truth. The red line is the weighted average of each worker's responses during the third iteration of DWMV. Because the weighted majority vote for each question remained constant between the second and third iteration, no more iterations are performed. Figure 3 shows the actual error rate of each worker compared to their weight as determined by DWMV after the DWMV method converged. Workers with higher error rates had lower weights, therefore their opinions mattered less when determining the majority vote. In this example, traditional majority vote achieved 98.4% accuracy, using DWMV achieved 100% accuracy.

3.2 Simulation Study

To determine if DWMV had a positive impact on forming groups from crowdsourced opinion, a simulation study was performed to compare the DWMV method to the traditional majority vote method in a variety of conditions. Figure 4 illustrates the simulation process. In the simulation study, hierarchical clustering was used to form groups from simulated workers' opinions of item similarity aggregated using both the majority vote method and the DWMV method. Table 1 lists the different initial parameters and their values used in the simulation. Five trials of every possible combination of the values in Table 1 were simulated for a total of 37,500 simulation runs.

The simulation began by randomly placing i items into g groups, where i and g are initial parameters of the simulation. Then the simulation crated ten workers. Each worker had a false positive rate and a false negative rate. These values were calculated separately to make the simulation more true to real life. In real life, it is not often that a

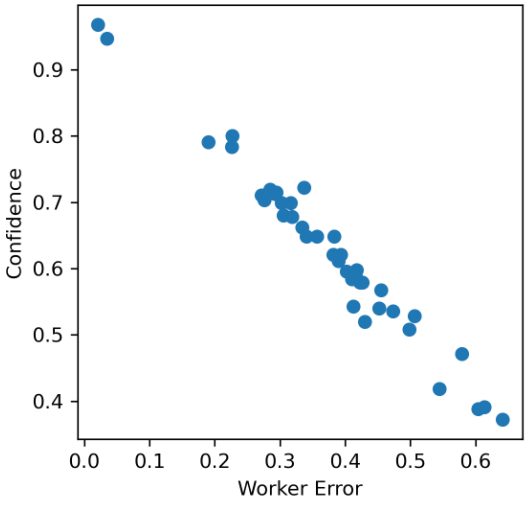


Figure 3: DWMV's confidence in each worker after the DWMV method converged.

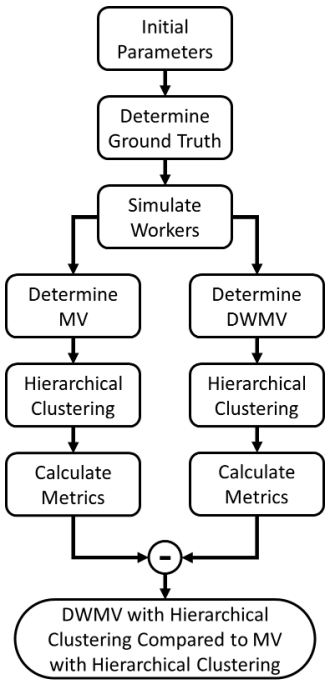


Figure 4: A flowchart of the simulation process, DWMV and majority vote were compared to each other through their use in community detection through hierarchical clustering.

Table 1: Simulation Parameters and Simulated Values

Parameter	Values
i	50, 100, 150, 200
g	5, 10, 15, 20, 25
w_{fp}	0.1, 0.2, 0.3, 0.4, 0.5
w_{fn}	0.1, 0.2, 0.3, 0.4, 0.5
p	20, 40, 60, 80, 100
d	0.25, 0.5, 0.75

worker would have an equal chance of incorrectly asserting that two items are or are not similar. The more likely case is that some workers think there is more similarity and other workers think there is less similarity between items than the actual similarity of items. The false positive and false negative rates of the workers were sampled separately for each worker from a uniform distribution in the range $[0, w_{fp}]$ and $[0, w_{fn}]$ respectively, where w_{fp} and w_{fn} are initial parameters of the simulation. Once the items were randomly placed in groups, and the error rates of the workers were randomly determined, a random p percent of all pairs of items were given to each worker, where p is an initial parameter of the simulation. Each worker then determined whether or not the items in each pair they received were similar to each other, taking into account their error rates.

Once all workers asserted whether or not each item pair they were given contained similar items, the majority vote and DWMV for the similarity of each item pair was calculated. The majority votes and DWMVs of item similarity were then used to form a network of item similarity, where each item is connected to every other item it was voted to be similar to. The majority vote network and DWMV network were both used to form groups through hierarchical clustering with Jaccard Index as the distance metric. Jaccard Index was used as the distance metric because Jaccard Index does not take into account true negatives [23]. Most items are not similar to each other, so a metric that takes into account true negatives would be over-inflated and not as informative in this context. After forming groups from the majority vote and DWMV similarity networks, the difference in accuracy, precision, and recall between the groups formed from the majority vote and DWMV similarity networks were used to determine if the DWMV method improved upon tradition majority vote.

3.3 Empirical Study: Similar Problems

In addition to a simulation, an empirical study was performed to compare DWMV to majority vote on a real crowdsorcing task. In this study, middle school mathematics teachers and college students were given 50 seventh grade mathematics problems from the Engage New York[1], Illustrative Mathematics[2], and Utah Middle School Math Project[3] curricula. Two examples of problems, as they are shown in ASSISTments[4], an online learning platform [11], are shown in Figure 5. Each worker was told to identify problems that evaluate similar mathematics skills. The workers' crowdsourced opinions of similarity were aggregated using both DWMV and majority vote, and then grouped using hierarchical clustering, with Jaccard Index as the distance metric with a threshold of 0.75. The resulting groups were then compared to a ground truth, provided by ASSISTments in the form of Common Core State Standards Mathematics Skill Codes[5], which each problem was tagged with. These ground truth skill tags were determined by trained experts and the designers of the above stated curricula. The difference in accuracy, precision, and recall between groups

[1]https://www.engageny.org/
[2]https://illustrativemathematics.org/
[3]http://utahmiddleschoolmath.org/
[4]https://new.assistments.org/
[5]http://www.corestandards.org/

formed with hierarchical clustering from DWMV and majority vote were again used to evaluate the quality of the DWMV algorithm.

4. RESULTS

4.1 Simulation Study

To compare the DWMV method to the traditional majority vote method, the difference in accuracy, precision, and recall as a function of w_{fp}, w_{fn}, i, g, and p, as described in Section 3.2, were calculated. Figure 6 shows the independent impact of these five separate simulation parameters on the three different dependant measures. Each plot in Figure 6 shows the difference in a performance metric between groups formed from the DWMV method and groups formed from the majority vote method as a function of one of the simulation parameters. Each plot shows the average and 95% confidence interval of all the simulation runs at the specified value of the independent measure. For example, the upper leftmost plot shows that the groups formed from the DWMV method were significantly more accurate than the groups formed from the majority vote method when the workers had low false positive rates. Then, as the maximum false positive rate of workers in the simulation grew, the improvement in accuracy fell but then increased.

The first positive takeaway from Figure 6 is that DWMV was almost always more accurate than majority vote, regardless of the simulation parameters. Only when the number of groups was high, or the maximum false negative rate of workers or links seen by workers was low did DWMV did not reliably out perform majority vote, but it did not significantly underperform either. At most, DWMV was slightly less accurate than majority vote when workers had very low false negative rates. Interestingly this increase in performance was not shared by both precision and recall. While recall followed the trend of accuracy and showed almost entirely positive improvements from using DWMV over majority vote, precision did not.

Another interesting finding is that while the trends in accuracy and precision tend to move in opposite directions, all three performance metrics increased as both the maximum false negative rate and fraction of links seen by workers increased. This implies that as workers answer more problems, and become worse at correctly identifying when items are similar, the benefit of using DWMV over majority vote increases.

Overall, t-tests [25] showed that using DWMV led to a statistically reliable ($p < 0.001$) 0.18% increase in accuracy, a statistically reliable 1.78% ($p < 0.001$) increase in recall, but no statistically reliable ($p = 0.28$) change in precision. Even though the plots in Figure 6 show a decrease in precision, this decrease was too insignificant to be statistically reliable. While small, these reliable improvements in accuracy and recall over the traditional majority vote method are an indication of the potential positive effects of transitioning to using DWMV instead of majority vote when aggregating crowdsourced opinion.

There were also some interesting differences in how different types of error affected the weights of workers as determined by the DWMV method. Figure 7 shows the average and 95% confidence interval of the DWMV weights of workers as a function of the workers' false positive and false negative rates. The false positive rate of the workers seems to decrease their weight in the final weighted majority vote of the DWMV method much more quickly than their false negative rate. A potential cause of this is that, in the simulated groups of similar items, there were far more pairs of items that were not similar to each other than there were pairs of items that were similar. For example, to have an equal number of items that are similar and not similar to each other, each item would have to be similar to half the items. The only way to facilitate that in the context of this simulation would be to have only two equally sized groups of items. In the simulation there were always at least five groups, and up to 25 groups of similar items, which caused most problems to not being similar to each other. Therefore, when a worker had a large false positive rate, there were more opportunities for them to make a mistake compared to a worker with a large false negative rate. Additionally, the large number of dissimilar problem pairs compared to the number of similar problem pairs caused workers with very low false positive rates to have higher weights than workers with equally low false negative rates, because workers with low false positive rates, regardless of their false negative rates, had much fewer opportunities to make a mistake. These findings suggest that the distribution of correct responses in crowdsourcing tasks affects which type of worker error has a larger impact on workers' weights in the DWMV method.

4.2 Empirical Study: Similar Problems

In total, six teachers and four students completed the crowdsourcing task of grouping 50 seventh grade mathematics problems. Using each worker's assertions of similarity, the DWMV method and traditional majority vote were used to aggregate the opinions of the workers into a final network of similarity, which was then used to create groups of similar problems using hierarchical clustering. This is the exact same process that was used to form groups in the simulation study. Figure 8 shows the progressive iterations of DWMV. Iteration 1 shows the unweighted average of each workers assertions. The DWMV method's process of iterating between calculating a weight for each worker and calculating the weighted majority vote shifted the weighted average of workers assertions toward the ground truth similarity of problems. This convergence was present in the simulated example in Section 3.1 as well. The benefit of the DWMV method over traditional majority vote lies in this ability to converge towards ground truth. Figure 9 shows the weight of each worker as a function of their error rate. The cohort of middle school mathematics teachers performed much better overall than the cohort of college students. The average accuracy of the teachers was about 97% while the average accuracy of the college students was only about 81%. Based on these weights, it is clear that the DWMV method valued the opinions of middle school mathematics teachers more than the opinions of college students, which is expected given the context and task. While, in this scenario, it might have been easy for a human in the loop to recognize that the teachers' opinions should be valued more, it will not always be the case that one group of workers is clearly more qualified than another group, and thus the DWMV method can help elucidate which workers are the most reliable.

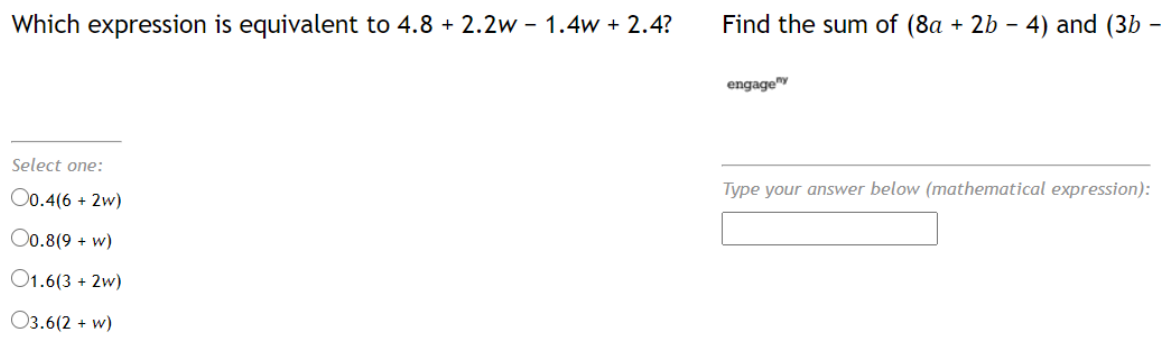


Figure 5: An example of two seventh grade mathematics questions.

Table 2 shows the difference in accuracy, precision, and recall between groups formed through hierarchical clustering from the assertions of similarity aggregated using DWMV and traditional majority vote. Similar to the simulation results, DWMV had the largest positive impact on recall, the second largest positive impact on accuracy, but no impact on precision. In this empirical study, both the traditional majority vote method and the DWMV method led to perfect precision, meaning all problems that were placed in groups together were similar to each other. However, traditional majority vote led to worse recall than DWMV. When tradition majority vote was used, three of the 50 problems were not placed in a group with any other problems, which is why the recall was so low. However, when DWMV was used, only one problem was not placed in a group of similar problems. This outlier problem, that neither traditional majority vote nor DWMV was able to correctly identify as similar to other problems in its group, had the following text:

> 22% of 65 is 14.3. What is 22.6% of 65? Round your answer to the nearest hundredths (second) decimal place.

Below are examples of problems in the same group as this problem, which were all correctly identified as similar to each other.

> Josiah and Tillery have new jobs at YumYum's Ice Cream Parlor. Josiah is Tillery's manager. In their first year, Josiah will be paid $14 per hour, and Tillery will be paid $7 per hour. They have been told that after every year with the company, they will each be given a raise of $2 per hour. Is the relationship between Josiah's pay and Tillery's pay rate proportional?

> To make a punch, Anna adds 8 ounces of apple juice for every 4 ounces of orange juice. If she uses 32 ounces of apple juice, which proportion can she use to find the number of ounces of orange juice x she should add to make the punch?

> A recent study claimed that in any given month, for every 5 text messages a boy sent or received, a girl sent or received 7 text messages. Is the relationship between the number of text messages sent or received by boys proportional to the number of text messages sent or received by girls?

Table 2: A comparison of majority vote to DWMV used to form groups of similar problems from crowdsourced assertions of similarity.

Metric	Majority Vote	DWMV	% Increase
Accuracy	0.987	0.997	1.054
Precision	1.000	1.000	0.000
Recall	0.903	0.977	8.228

Although all these problems are related to ratios and proportions, the other problems in the group with the outlier problem are longer word problems that do not explicitly use percentages. The teachers and students whose opinions were crowdsourced could have missed the connection due to the different wording in the problems, or they could believe that calculating percentages is a different skill than calculating proportions from word problems. Based on the differences between this single outlier problem and the other problems in its group, it is possible that the outlier problem was consciously excluded from its group and not simply an oversight.

The impact of using DWMV was larger in this empirical study than it was in the simulation. Looking at Figure 6, in the simulation there was a larger than average improvement in accuracy and recall when the workers had very low false positive rates. Given that in this empirical study both sets of groups of similar problems had perfect precision, it is likely that the workers in this study had very low false positive rates, which likely contributed to why the positive impact of using DWMV instead of majority vote was larger in this empirical study than in the simulation as a whole. The results of this empirical study suggest that not only can DWMV out-perform traditional majority vote in simulations, but can also improve the recall and accuracy of groups of similar problems formed from crowdsourced opinions on content similarity in real-life scenarios as well.

5. LIMITATIONS AND FUTURE WORK

Although the simulation study found that the DWMV method had a statistically reliable improvement in accuracy and re-

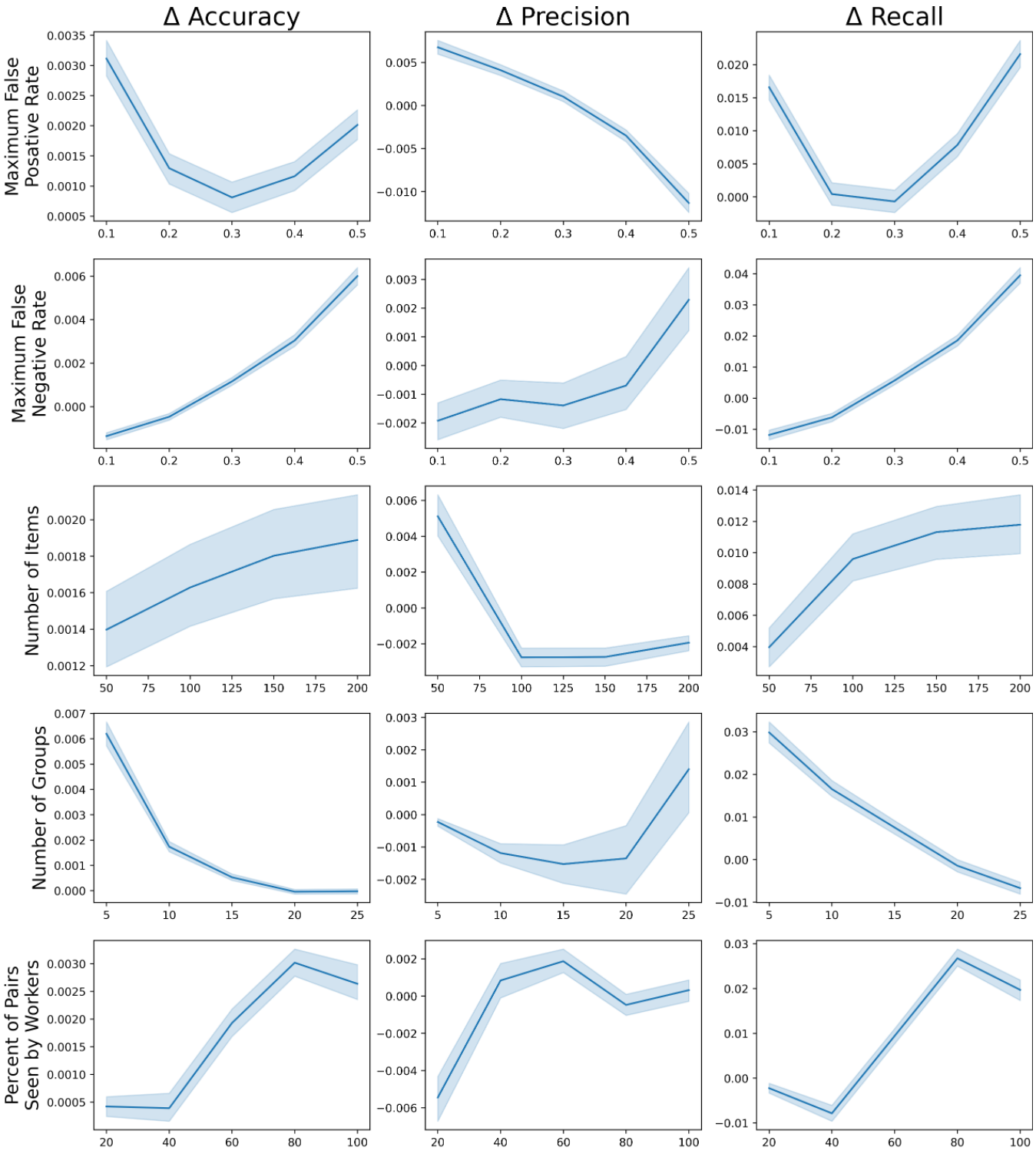


Figure 6: The difference in performance between DWMV and majority vote across five different simulation parameters and three different performance metrics.

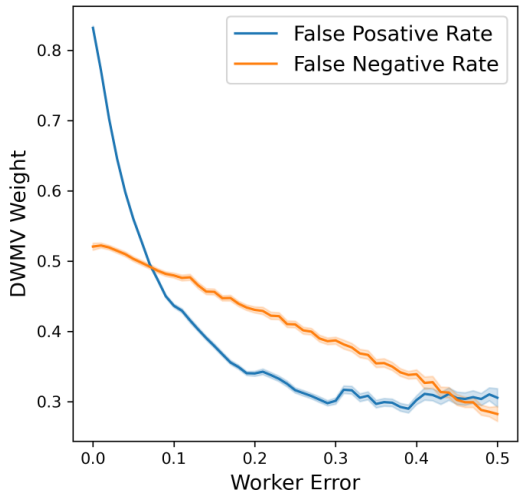


Figure 7: The average and 95% confidence interval of the DWMV weights of workers as a function of the workers' false positive and false negative rates.

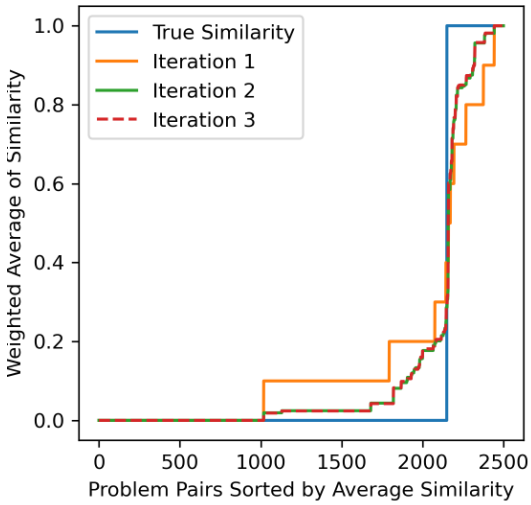


Figure 8: Progressive iterations of DWMV converging on empirical data.

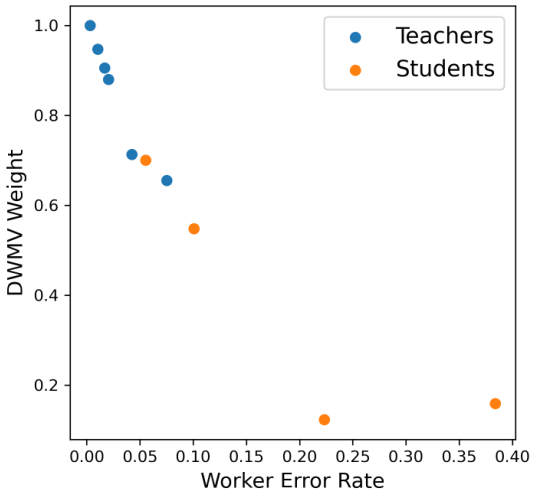


Figure 9: DWMV's confidence in each worker after the DWMV method converged.

call over the traditional majority vote method, the simulation poses some limitations. Primarily, a simulated worker is not the same as a real worker. Although steps were taken to mimic the behavior and reliability of workers, these steps were based on theory, not on data collected from past crowdsourcing endeavors. Future work could focus on creating a worker model based on patterns in actual workers' behavior. This model could take into account things like bias that comes from knowledge of previously asked questions and answers, for example, the uncertainty that comes from answering a multiple choice test and realizing one has chosen the same answer for the last ten questions. By modeling patterns in worker behavior and then simulating those patterns, there is potential for the simulation to better reflect how well DWMV performs compared to majority vote. Additionally, a larger simulation study could be performed with more conditions spanning a wider range. The number of workers could become a parameter of the simulation, and the number of groups and items could exceed 25 and 200 respectively. In the simulation study performed in this work, the chance of a worker mislabeling the similarity of two items never exceeded 50%, in other words, workers never performed worse than random chance. However, there could be a situation with antagonistic workers that either misunderstood their instructions or are intentionally providing incorrect responses to prompts. In the future, one could expand the simulation in this study to investigate how well the DWMV method compensates for antagonistic workers compared to the majority vote method.

The empirical study, while a good measurement of how well the DWMV method performed in a real-life scenario, also poses some limitations. It is important to consider the context of the empirical study. The workers who performed the best in the empirical study were middle school math teachers given the task of identifying similar seventh grade mathematics problems. They are experts in this subject, and even so, all of them were not 100% accurate. DWMV

works by finding workers that tend to agree with each other, and asserting that what they agree to be true is the truth. In this empirical study there were experts that could be used to determine this truth. However, if there are few or no experts in the group of workers whose opinions are being crowdsourced, it could be the case that common misconceptions among the workers could be mistaken for the truth, and lead to the few experts' assertions being weighted lower than the majority's misconceptions. Additionally, this empirical study was just one example of how DWMV performs, and it, like the simulation, only had ten workers. In the future, more empirical studies could be performed on more workers to evaluate how DWMV performs on a scale similar to Amazon Mechanical Turk[6], one of the largest crowdsourcing marketplaces. While the DWMV method has some clear improvements over the traditional majority vote method, it is not, by itself, a solution to all of the common crowdsourcing concerns. DWMV should be paired with other methods, such as training and testing of workers before allowing their opinions to be crowdsourced, in order to ensure as much accuracy as possible from crowdsourcing tasks.

6. CONCLUSION

Within online learning platforms and intelligent tutors, there is tremendous utility to knowing what content is similar to other content within the platform, but each application of similar content is likely to have different criteria for what is considered similar. Crowdsourcing opinions on the similarity of content is an accessible way for new applications to recognize similar content. However, crowdsourcing poses some difficulties, namely, how to identify reliable workers and properly aggregate opinions from multiple workers. This work has demonstrated the ability of the Dynamically Weighted Majority Vote method, a novel algorithm for aggregating crowdsourced opinion while rating workers, to accomplish those goals. DWMV has been shown, in both a simulation study and an empirical study, to lead to higher accuracy and recall that the traditional majority vote method on crowdsourcing tasks related to identifying similar content. In the simulation study, using DWMV before identifying groups of similar items through hierarchical clustering resulted in a statistically significant 0.18% increase in accuracy and a 1.78% increase in recall over using majority vote. The simulation study also revealed how the distribution of correct responses in the crowdsourcing tasks effects how the false positive and false negative rates of workers effects their weight in the DWMV method. In the empirical study, using DWMV before identifying groups of similar problems through hierarchical clustering resulted in about a 1% increase in accuracy and an 8% increase in recall over using majority vote, and provided perspective on the differences in accuracy between the expert middle school math teachers and the novice college students. Moving forward, when faced with the need to aggregate crowdsourced opinions, the learning science community can look to the DWMV method as an alternative to the traditional majority vote method. The DWMV method is a promising tool for increasing the reliability of crowdsourced opinion and, when paired with hierarchical clustering, identifying groups of similar content.

7. ACKNOWLEDGMENTS

Omitted for blinding purposes.

[6]https://www.mturk.com/

8. REFERENCES

[1] N. J. Cepeda, E. Vul, D. Rohrer, J. T. Wixted, and H. Pashler. Spacing effects in learning: A temporal ridgeline of optimal retention. *Psychological science*, 19(11):1095–1102, 2008.

[2] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[3] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys (CSUR)*, 51(1):1–40, 2018.

[4] F. N. Dempster. Spacing effects and their implications for theory and practice. *Educational Psychology Review*, 1(4):309–330, 1989.

[5] D. Faucher and S. Caves. Academic dishonesty: Innovative cheating techniques and the detection and prevention of them. *Teaching and Learning in Nursing*, 4(2):37–41, 2009.

[6] S. Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[7] J. Golden and M. Kohlbeck. Addressing cheating when using test bank questions in online classes. *Journal of Accounting Education*, 52:100671, 2020.

[8] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *International conference on intelligent tutoring systems*, pages 35–44. Springer, 2010.

[9] N. Gulbahce and S. Lehmann. The art of community detection. *BioEssays*, 30(10):934–938, 2008.

[10] C. Hanewicz, A. Platt, and A. Arendt. Creating a learner-centered teaching environment using student choice in assignments. *Distance Education*, 38(3):273–287, 2017.

[11] N. T. Heffernan and C. L. Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[12] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[13] S. H. Kang. Spaced repetition promotes efficient and effective learning: Policy implications for instruction. *Policy Insights from the Behavioral and Brain Sciences*, 3(1):12–19, 2016.

[14] J. D. Karpicke and A. Bauernschmidt. Spaced retrieval: absolute spacing enhances learning regardless of relative spacing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37(5):1250, 2011.

[15] A. Lancichinetti and S. Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

[16] P. C. Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936.

[17] F. Murtagh and P. Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[18] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.

[19] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*, 2009.

[20] C. Piech, J. Spencer, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. *arXiv preprint arXiv:1506.05908*, 2015.

[21] P. Ruvolo, J. Whitehill, and J. R. Movellan. Exploiting commonality and interaction effects in crowdsourcing tasks using latent factor models. In *Neural Information Processing Systems. Workshop on Crowdsourcing: Theory, Algorithms and Applications*. Citeseer, 2013.

[22] D. M. Stenhoff, B. J. Davey, B. Lignugaris, et al. The effects of choice on assignment completion and percent correct by a high school student with a learning disability. *Education and treatment of Children*, 31(2):203–211, 2008.

[23] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. Pearson Education India, 2016.

[24] H. A. Vlach and C. M. Sandhofer. Distributing learning over time: The spacing effect in children's acquisition and generalization of science concepts. *Child development*, 83(4):1137–1144, 2012.

[25] B. L. Welch. The generalization ofstudent's' problem when several different population variances are involved. *Biometrika*, 34(1/2):28–35, 1947.

[26] J. Whitehill, T.-f. Wu, J. Bergsma, J. Movellan, and P. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in neural information processing systems*, 22:2035–2043, 2009.

Chapter 2.3

Exploring Common Trends in Online Educational Experiments

Exploring Common Trends in Online Educational Experiments

Ethan Prihar
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
ebprihar@wpi.edu

Manaal Syed
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
msyed@wpi.edu

Korinn Ostrow
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
ksostrow@wpi.edu

Stacy Shaw
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
sshaw@wpi.edu

Adam Sales
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
asales@wpi.edu

Neil Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, Massachusetts
nth@wpi.edu

ABSTRACT

As online learning platforms become more ubiquitous throughout various curricula, there is a growing need to evaluate the effectiveness of these platforms and the different methods used to structure online education and tutoring. Towards this endeavor, some platforms have performed randomized controlled experiments to compare different user experiences, curriculum structures, and tutoring strategies in order to ensure the effectiveness of their platform and personalize the education of the students using it. These experiments are typically analyzed on an individual basis in order to reveal insights on a specific aspect of students' online educational experience. In this work, the data from 50,752 instances of 30,408 students participating in 50 different experiments conducted at scale within the online learning platform ASSISTments were aggregated and analyzed for consistent trends across experiments. By combining common experimental conditions and normalizing the dependent measures between experiments, this work has identified multiple statistically significant insights on the impact of various skill mastery requirements, strategies for personalization, and methods for tutoring in an online setting. This work can help direct further experimentation and inform the design and improvement of new and existing online learning platforms. The anonymized data compiled for this work are hosted by the Open Science Foundation and can be found at https://osf.io/59shv/.

Keywords

Randomized Controlled Experiments, Online Learning Platforms, Skill Mastery, Instructional Interventions, Online Tutoring

E. Prihar, M. Syed, K. Ostrow, S. Shaw, A. Sales, and N. Heffernan. Exploring common trends in online educational experiments. In A. Mitrovic and N. Bosch, editors, *Proceedings of the 15th International Conference on Educational Data Mining*, pages 27–38, Durham, United Kingdom, July 2022. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.6853041

1. INTRODUCTION

The use of online learning platforms has increased rapidly in the past decade [37]. As online learning platforms grow to become a permanent fixture of educational systems, they have the potential to democratize education by providing high quality free or low-cost resources to compliment traditional classroom practices [1]. While in some cases online tutoring has been shown to be at least as effective as traditional in-person educational practices [33, 11, 16], there is still a need to validate the effectiveness of the various methods by which educational content is delivered to students. Placing an emphasis on objectively measuring the effectiveness of these emerging methods through randomized controlled experimentation is essential for ensuring that the quality of educational resources continues to increase.

This study works towards that endeavor by aggregating the results from 50,752 instances of 30,408 students participating in 50 different randomized controlled experiments conducted by various groups of researchers since February, 2019 within the online learning platform ASSISTments. In these experiments, K-12 students were randomized between different conditions as they completed online mathematics assignments. These conditions changed factors such as students' assignment completion requirements, the format of the tutoring students' received when struggling with the assigned problems, and the types of interactions students could have within their assignment. While these types of studies have been conducted in ASSISTments before [34, 40], this work goes beyond reporting the results of each individual study, and instead aggregates the results of these studies together, ultimately investigating 19 different research questions across 50 randomized controlled experiments. To achieve this, the following steps were taken.

1. Identify the independent measures of every condition in each experiment.
2. Normalize the dependent measures of all the experiments so they can be compared to one another.
3. Combine the data from different experiments when the research questions of the experiments match.

4. Determine the effects of the various experimental instructional interventions using these combined datasets.

The results of this aggregate analysis revealed actionable trends that can contribute to a broader understanding of the effectiveness of different educational interventions, help direct further experimentation, and inform the design and improvement of new and existing online learning platforms.

2. BACKGROUND

2.1 Educational Experiments

Experiments revolving around educational practices have been conducted since the late 19th century [49]. These early experiments, conducted by William James, Edward Thorndike, and Alfred Binet along with others, focused on determining individual differences between students, why they occur, and what methods teachers can employ to improve educational outcomes for them [49]. By the early 20th century, with the increased accessibility of formal education, educational experiments were more focused on improving teaching methods [49] and connecting cognitive psychology to classroom practices [24]. These studies investigated the differences in learning between students of varying socioeconomic levels [7], the effect of increasing student autonomy in the classroom [29], and the value of assessment in learning [21].

In the years following these studies, theories on educational development, classroom practice and structure, and how to approach individual differences between students were developed. In particular, research around effective feedback has proven to increase performance [23], interest in learning [9], as well as increasing students' abilities to self-regulate their learning [35]. These studies varied in the types of feedback students' receive [9], level of specificity and frequency [48], level of praise present in the feedback [8], and what types of students benefit the most from certain types of feedback [14]. Data for these studies were collected from classroom observations of verbal feedback, collections of written feedback, and results on written assessments.

2.2 Experimentation within Online Learning Platforms

Computer-assisted instruction in education has been studied since the 1960s [47], results of these early studies show that providing specific, targeted feedback to student responses improves retention of information [19, 43]. In more recent years, educational data mining research has grown significantly, with large scale implementation of online A/B testing in web applications allowing thousands of users to be randomized into conditions simultaneously [3, 5]. With the rapid adoption of computers in the classroom in the past two decades, educational researchers now have access to an abundance of data on students. Online learning platforms track students' performance, demographics, interactions within the platform, statistics on content usage, feedback, and more [38]. Additionally, during the 2020-2021 school year many schools that had not previously used online learning platforms migrated to online learning platforms as a result of the COVID-19 pandemic [28]. This increase in the size and scope of available data has made it possible to gain insights into educational practices that were not previously possible with traditional methods.

Recent studies have focused on predicting student outcomes, improving domain specific content, examining the effects of different kinds of pedagogical support, and advancing knowledge about how people learn [5]. Similar to early studies on computer-assisted learning, learning analytics research aims to determine what types of feedback and presentations work well for what types of students, in other words, discovering the potential for personalization in online learning platforms [31, 5]. Prior studies on personalization show the benefit of explanatory feedback over corrective feedback for novice students [31], differences in effect of feedback between male and female students [32], and the effects of immediate and delayed feedback for students with different prior knowledge levels [45, 10]. Additionally, by taking advantage of recent advances in data collection, research has been able to focus on determining methods for personalizing based on students characteristics, such as district locale and student interaction data [2] and what types of crowdsourced content is effective for students [39]. This work provides another data-intensive analysis on the effectiveness of different aspects of online learning platforms, but unlike the aforementioned analyses, this work compiled data from dozens of studies performed within an online learning platform instead of focusing on a single study. This revealed trends across experiments that provided deeper insight into the effectiveness of various instructional interventions and online tutor designs.

2.3 ASSISTments and E-TRIALS

The data in this work comes from ASSISTments, an online learning platform that focuses on providing teachers with mathematics content and resources to effectively manage their students. Within ASSISTments, teachers have the option to assign problem sets and skill builders to their students. Problem sets are a series of mathematics problems that must all be completed, in order, to finish the assignment. These problem sets come from various open educational resources for mathematics such as Engage New York, Illustrative Mathematics, and The Utah Middle School Math Project. Skill builders are assignments that focus on a specific mathematics skill. When students complete skill builders they are given a series of problems on the same mathematics skill until they get a specific number of problems correct in a row. Usually students must answer three problems correct in a row to finish the assignment, but this number is configurable by the teacher.

Regardless of whether the student is assigned a problem set or skill builder, they complete their assignment in the ASSISTments tutor [20]. In the tutor, students receive immediate feedback when they submit a response to a problem, which informs them if they are correct [27]. In addition to this immediate feedback, students are able to request tutoring, which is available to them at any point during their completion of a problem regardless of whether or not they have already attempted the problem. Tutoring comes in the form of hints, which are a series of messages the student can request, one at a time, that explain how to solve parts of the problem; explanations, which are full worked solutions to the problem; examples, which are full worked solutions

of a similar problem; common wrong answer feedback messages, which explain how to correct a specific error made by the student; and scaffolding, which breaks the problem down into a series of simpler problems that guide the student through how to solve the original problem [27]. These different types of tutoring strategies can come in the form of videos, images, or text. An example of a student receiving a text-based explanation within the ASSISTments tutor is shown in Figure 1. Once students have finished their assignment, teachers are provided with reports that aggregate information such as how each student progressed through the assignment and what the class' most common mistakes were.

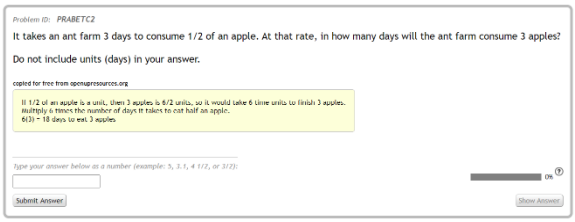


Figure 1: A student's view of the ASSISTments tutor after requesting tutoring and receiving a text-based explanation.

The variety of assignments and tutoring strategies that can be delivered to students through ASSISTments provides opportunities to explore various research questions in learning science, educational psychology, and human-computer interaction. A research test-bed, E-TRIALS (an EdTech Research Infrastructure to Advance Learning Science), was built to deploy randomized controlled experiments in classwork and homework settings at scale within ASSISTments [25]. Since 2005, researchers have been able to create and modify problem sets, skill builders, and tutoring strategies. The modified content contains the original content within it, but adds experimental conditions. For example, a researcher could modify a skill builder for calculating the area of a triangle to randomly provide students with text-based or video-based hints. Teachers assign the modified content as if it were the original, and when teachers assign these modified assignments, students will be randomized (on an individual basis, not at the class level) to one of multiple conditions. This allows researchers to evaluate the impact of different pedagogical decisions on students' learning [41]. The experiments run in ASSISTments cover a wide scope of research questions that range from whether offering students a choice in the difficulty of their instruction improves learning, to whether providing students with worked examples of similar problems is more effective than direct advice on the problem they are struggling with, to whether changing the number of problems students are required to complete affects their learning [44]. The following analysis of E-TRIALS experiments provides insight into the current state of experimentation within online learning platforms and can help inform the design of future experiments.

3. EXPERIMENT DATASET

The dataset used in this work comes from 50,752 instances of 30,408 unique students who participated in 50 E-TRIALS experiments since February, 2019. In addition to recording the purpose of the experiment, the experimental condition each student was placed in, and the resulting dependent measure, the dataset also includes information on students' performance within ASSISTments prior to participating in the experiment, the prior performance of the students' classes, the experience of their teachers, and an indicator of their socioeconomic status. Socioeconomic status is indicated by a student's school district's Opportunity Zone status, which is a particular tax classification in the United States of America that indicates whether a region has opportunities for economic growth. The regions in opportunity zones are typically low-income regions with fewer educational resources [15]. The Opportunity Zone status for each student was determined using the domain name of their teacher's school-provided email address. No demographic information was requested from students using ASSISTments to preserve their anonymity and prevent any bias associated with answering questions on how they identify themselves. The full set of features collected for each participant is shown in Table 1. In addition to containing features for each experiment participant, the dataset contains information on the independent and dependent measures used in the various experiments, which had to be aggregated in order to determine the common trends among the 50 experiments. The details of these independent and dependant measures and how they were aggregated are discussed in Sections 4.1 and 4.2.

4. METHODOLOGY

Due to the diversity in research questions, independent and dependent measures, and structure of the experiments, the first step to evaluate their overall trends was to identify similar conditions within multiple experiments. This process involved documenting each condition of each experiment and identifying when different experiments had an identical pair of conditions or the same research question. The second step was to normalize the various dependent measures such that they all represented similar metrics and used the same scale.

4.1 Pooling Experiment Data

To pool experimental data together, similar experiments had to be identified. To do this, every condition from every experiment was documented such that data from multiple experiments that each had an identical pair of conditions or research question could be aggregated. For example, if the following three experiments were run in ASSISTments, then Experiment 1 would have six documented conditions (one condition for each of the hint types for both choice and no choice), Experiment 2 would have two conditions (one for text-based hints and one for video-based hints), and Experiment 3 would have four conditions (one condition for each text color for both choice and no choice).

- **Experiment 1**: Randomize between A: giving students a choice of no hints, text-based hints, or video-based hints, or B: randomly selecting which type of hint to give them.
- **Experiment 2**: Randomize between A: text-based hints, or B: video-based hints.
- **Experiment 3**: Randomize between A: giving students a choice of black or red text color, or B: randomly selecting the text color.

Table 1: The Features Calculated for each Instance of a Student Participating in an Experiment

Feature Name	Description
Experiment Condition	An indication of which condition students are in.
Student Prior Started Skill Builder Count	Number of skill builders previously started by students.
Student Prior Skill Builder Percent Completed	Percent of skill builders completed by students.
Student Prior Started Problem Set Count	Number of problem sets previously started by students.
Student Prior Problem Set Percent Completed	Percent of problem sets completed by students.
Student Prior Completed Problem Count	Total number of problems completed by students.
Student Prior Median First Response Time	Students' median time to submit an answer to a problem.
Student Prior Median Time On Task	Students' median time to complete a problem.
Student Prior Average Attempt Count	Student's average attempts required to complete a problem.
Student Prior Average Correctness	The fraction of problems students answered correctly.
Class Age In Days	The number of days classes existed in ASSISTments.
Class Student Count	The number of students in the class.
Class Prior Started Skill Builder Count	Number of skill builders previously started by classes.
Class Prior Skill Builder Percent Completed	Percent of skill builders started by classes that were completed.
Class Prior Started Problem Set Count	Number of problem sets previously started by classes.
Class Prior Problem Set Percent Completed	Percent of problem sets started by classes that were completed.
Class Prior Completed Problem Count	Total number of problems completed by classes.
Class Prior Median First Response Time	Class' median time to to submit an answer to a problem.
Class Prior Median Time On Task	Class' median time to complete a problem.
Class Prior Average Attempt Count	Class' average attempts required to complete a problem.
Class Prior Average Correctness	The fraction of problems classes answered correctly.
Teacher Account Age In Days	The number of days teachers have had an ASSISTments account.
Experiment Id	The experiment students participated in.
Opportunity Zone	The school district's Opportunity Zone status [15].

In addition to documenting the conditions for the three experiments, Experiments 1 and 3 would be recorded as having the higher-level research question "Choice vs. No Choice" and Experiment 2 would be recorded as having no higher-level research question. To combine the results of these three experiments, students randomized to the text-based hint option of Condition B of Experiment 1 would be combined with students randomized to Condition A of Experiment 2 and students randomized to the video-based hint option of Condition B of Experiment 1 would be combined with students randomized to Condition B of Experiment 2. These groups would be used to evaluate the overall effect of giving video-based hints compared to text-based hints. Additionally, students randomized to Condition A of Experiments 1 and 3 would be grouped, and students randomized to Condition B of Experiments 1 and 3 would be grouped. These two groups would be used to evaluate the overall effect of offering students a choice.

When performing this aggregation on the real experiments, many experiments were too unique to have similar experimental conditions as other experiments. Additionally, some experiments were created incorrectly in ASSISTments or had broken links to videos, leading students to never be randomized to a condition. Even though 103 experiments have been deployed in ASSISTments since 2019, only 50 had at least one condition similar to a condition in another experiment and were complete enough to be included in the analyses. After parsing through the data and removing poorly structured and broken experiments, the most common research questions were selected for further analysis. Table 2 shows the selected research questions and statistics on the data aggregated to evaluate the research questions. Students were typically divided evenly between the different conditions, but for the research question "Emotion vs. No Emotion", there were six conditions that included positive emotional content and two conditions that did not include emotional content, which is why about three fourths of students are placed in the treatment condition.

The six research questions containing the phrase "Correct for Mastery" all investigated differences in the requirements to complete a skill builder assignment. In a skill builder, students must correctly answer a specific number of problems in a row to complete the assignment. The different values in these research questions represent the different number of problems students had to get correct in a row before finishing the assignment or completing a posttest. The six research questions that compare something to "Answer Only" investigated how six different tutoring strategies improved student learning compared to just giving struggling students the answer. Table 3 describes each tutoring strategy investigated by these research questions. The other seven research questions are not related to other research questions, but examined different aspects of the structure of assignments and tutoring in online learning platforms.

- **Video vs. Text** investigated the difference between providing two different types of tutoring which were almost identical, except in one condition the tutoring content was text-based, and in the other condition the same tutoring was provided in a video format.
- **Common Wrong Answer Feedback vs. No Feedback** investigated the effect of providing students with specific feedback messages when they submitted a common wrong answer to any of the the problems in their assignment.

Table 2: Research Questions Selected for Analysis

Research Question	Experiment #	Student #	% in Treatment
2 Correct for Mastery vs. 3 Correct for Mastery	4	1192	0.487
2 Correct for Mastery vs. 4 Correct for Mastery	4	1165	0.475
2 Correct for Mastery vs. 5 Correct for Mastery	3	846	0.483
3 Correct for Mastery vs. 4 Correct for Mastery	5	2030	0.492
3 Correct for Mastery vs. 5 Correct for Mastery	4	1683	0.494
4 Correct for Mastery vs. 5 Correct for Mastery	4	1681	0.495
Example vs. Answer Only	3	765	0.467
Explanation vs. Answer Only	1	85	0.471
Hint vs. Answer Only	5	1192	0.513
Scaffolding vs. Answer Only	7	2010	0.546
Video Example vs. Answer Only	1	366	0.484
Video Scaffolding vs. Answer Only	3	1033	0.509
Video vs. Text	5	2492	0.497
Common Wrong Answer Feedback vs. No Feedback	2	7046	0.497
Adaptive vs. Non-Adaptive	9	7754	0.498
Fill-In vs. Multiple Choice	2	4057	0.493
Choice vs. No Choice	9	12789	0.499
Emotion vs. No Emotion	2	1211	0.766
Motivational vs. Non-Motivational	14	12243	0.581

- **Adaptive vs. Non-Adaptive** investigated the impact of changing the difficulty of problems based on how well students performed at the beginning of their assignment. Students that got problems correct at the beginning were given more challenging problems than the students that got the beginning problems incorrect.
- **Fill-In vs. Multiple Choice** investigated the impact of requiring students to write the correct answer in themselves compared to selecting from multiple preset options when answering questions.
- **Choice vs. No Choice** investigated the impact of allowing students to choose which version of various configurations for their assignments they would complete.
- **Emotion vs. No Emotion** investigated the impact of including positive emotional phrases and images in the body of the problems in the assignment. For example, an emotional problem would say "Susan excitedly purchased three apples." instead of "Susan purchased three apples.".
- **Motivational vs. Non-Motivational** investigated the impact of interjecting motivational messages and videos into the assignment.

4.2 Normalizing Student Learning

In addition to identifying similar conditions and research questions, the different experiments dependent measures had to be normalized such that the results from one experiment could be compared to another experiment. Normally, it would be very difficult to combine dependent measures from different experiments, but conveniently, all of the experiment in ASSISTments are attempting to increase student learning, and therefore the various dependent measures are just different ways of measuring student learning and can thus be normalized and combined. In the various E-TRIALS experiments, there are five different dependent measures used, described in Table 4.

While all of these measures represent student learning, they do not all increase as student learning increases, nor do they all have the same range, nor do they all take into account when a student fails to complete the experimental assignment, which presumably means they learned the least. To rectify these concerns, Table 5 shows the function $f(x)$ applied to each of the dependent measures. After $f(x)$ is applied to the dependent measures, the values are z-scored within each experiment using the pooled standard deviation grouped by experimental condition. This ensured that all of the different measures of learning increased as student learning increased, had the same scale, and accounted for incomplete assignments. These transformations converted all the dependent measures into a measurement of how many standard deviations above or below average each student performed compared to other students that participated in the same experiment. $f(x)$ for problems to mastery is particularly complicated because unlike the other dependent measures, problems to mastery goes down the more a student learns, and problems to mastery is bounded in the range $[3, \infty)$. Therefore, to ensure that $f(x)$ for problems to mastery increases the more a student learns, problems to mastery was transformed by inverting it, then multiplying it by 3. However, this transformation alters problems to mastery non-linearly, so to correct some of the non-linearity, the square root is taken, which makes $f(x)$ appear linear in the range $[3, 10]$ where most of the results lie.

4.3 Evaluating Differences in Student Learning

To measure the effects of the various experimental treatment conditions, Cohen's d [12] was used to calculate the effect size between the control and treatment conditions for each research question. To test for a difference between treatment and control, we ran ordinary least squares models and examined the associated p-values and 95% confidence intervals of the mean differences between conditions, and used Cohen's d to capture the magnitude of any effect. This model was used to predict normalized student learning

Table 3: Descriptions of Different Tutoring Strategies

Tutoring Strategy	Description
Example	An explained solution to similar problem.
Explanation	An explained solution to the current problem.
Hint	Step-by-step advice on how to solve the current problem.
Scaffolding	A series of problems that break the current problem into smaller steps with explanations.
Video Example	An example recorded in a video instead of text.
Video Scaffolding	A scaffolding with explanations recorded in videos instead of text.

Table 4: Descriptions and Frequencies of the Dependent Measures used to Evaluate Student Learning

Dependent Measure	Frequency of Use	Description
Problems to Mastery	44%	# of problems the student completed to get n correct in a row.
Posttest Score	44%	% correct on posttest.
Learning Gains	7%	% correct on posttest - % correct on pretest.
Assignment Correctness	3%	# of problems correct / # of problems in condition.
Assignment Completion	1%	Binary indicator for if the student completed the assignment.

Table 5: Functions used to Scale the Dependent Measures Before z-Scoring

Dependent Measure	$f(x)$
Problems to Mastery	0 if incomplete else $\sqrt{\frac{3}{x}}$
Posttest Score	0 if incomplete else x
Learning Gains	0 if incomplete else $x+1$
Assignment Correctness	x
Assignment Completion	x

based on the experiment condition the student was placed in, the experiment the student participated in, and features of the student, their class, their teacher, and their school district. Including fixed effects for which experiment the student participated in allowed the model to associate differences in normalized student learning between experiments with those coefficients, and not the experiment condition coefficient, helping to reduce noise from covariates. The inputs related to students, classes, teachers, and school districts also helped to remove noise from the experiment condition coefficient. For example, students with high prior knowledge performed better on the experimental assignments than students with low knowledge, and by including students' prior knowledge in the model, the variability in students' success based on their prior knowledge will be associated with the prior knowledge coefficient, and have a lesser effect on the treatment coefficient. Table 1 contains a full list of the features used to model the effects of the various experimental conditions. The "Experiment Condition" feature was used to determine the 95% confidence interval and p-value of the impact of the various experimental instructional interventions. When some features were not available, such as when students that had not previously used ASSISTments participated in the experiments, the missing values were filled using the average value across the data used to fit the model. This limited the extent to which the missing values biased the model's coefficients.

5. RESULTS

5.1 Different Completion Requirements

Investigating the impact of different mastery requirements for skill builders found that requiring fewer problems led to higher student learning than requiring more problems, but that this effect is mostly due to students not completing the assignment when they were required to answer more problems correct in a row to proceed. Figure 2 shows the effect size and, in parentheses, the p-value of the effect of requiring students get different numbers of problems correct in a row. For example, the cell at row two, column three contains the effect size and p-value of requiring students get two problems correct in a row instead of three problems correct in a row. Figure 2 only shows significant positive effects when requiring students to complete two problems in a row correctly instead of three, four, or five.

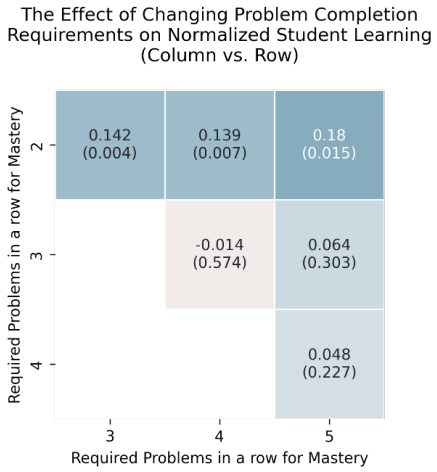


Figure 2: The effect of changing problem completion requirements on normalized student learning. Each cell contains the effect size, determined using Cohen's d, and in parentheses, the p-value.

To investigate further, the effect of changing problem completion requirements on assignment completion and the effect of changing problem completion requirements on student learning for only students that completed the assignment were calculated. Figure 3 shows the results of these

analyses. Based on these results, there is no statistically significant effects on student learning for students that completed their assignment, regardless of how many problems they had to complete correctly in a row before finishing the assignment. The vast majority of the effects seen in Figure 2 come from more students failing to complete their assignment when having to complete more problems correct in a row. Essentially, when students have to complete more problems they are less likely to complete their assignment, but if students complete their assignment their learning will be unaffected by how many problems they had to complete.

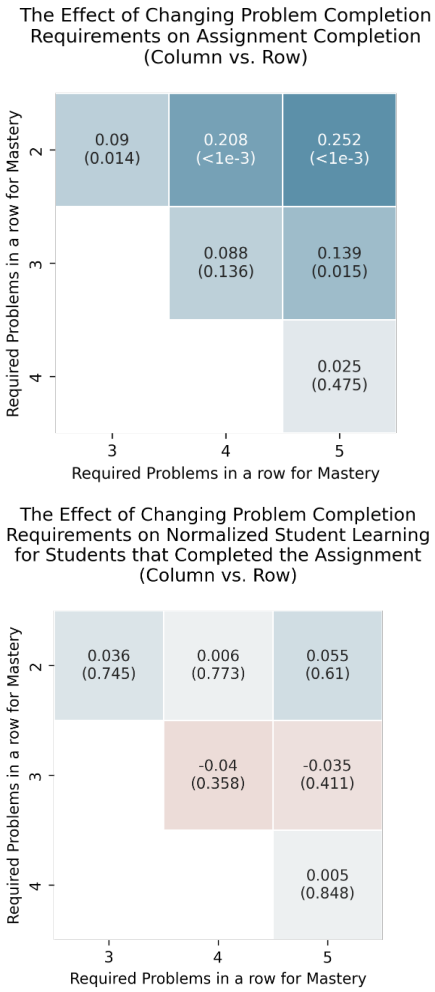


Figure 3: The effect of changing problem completion requirements on assignment completion (top) and normalized student learning for only students that completed the assignment (bottom). Each cell contains the effect size, determined using Cohen's d, and in parentheses, the p-value.

One would expect that if any of these mastery requirements were a meaningful metric for determining if students had learned the material, then there would be a statistically significant difference in students' learning between students that had to complete above or below a certain number of problems correct in a row. However, this was not the case. These results imply that a more sophisticated method could be necessary to evaluate whether students have mastered the mathematics concepts present in their assignments. It may therefore be advisable to integrate Knowledge Tracing [13] or Performance Factors Analysis [36], which are both effective methods for evaluating students' mastery of individual skills, into ASSISTments and other online learning platforms.

5.2 Different Tutoring Strategies

Investigating the effects of different types of tutoring on student learning found that most tutoring is effective, and that giving students tutoring instead of showing them the answer is more effective for low knowledge students than high knowledge students. Figure 4 shows the confidence interval, effect size, number of students, and p-value for the effect of giving students each type of tutoring instead of just providing the answer. The only tutoring strategy that had no significant impact on student learning was explanations, which had a wide confidence interval and relatively few participants.

Prior studies done in ASSISTments reported that lower knowledge students benefited more from scaffolding while higher knowledge students benefited more from short explanations [42]. Therefore, in addition to evaluating the effect of each of the above tutoring strategies on all students that participated in the experiments, the data from the experiments were divided into below and above average prior knowledge groups based on whether students' prior average correctness was above or below the average of all students' prior average correctness. Figure 5 shows the difference in the effectiveness of four of the six tutoring strategies for each of these groups of students. Only four of the six tutoring strategies from Figure 4 are included in these plots because the other two tutoring strategies were used in experiments that did not have any participants that had used ASSISTments previously, and therefore no prior average correctness was available for those students. The below average prior knowledge students consistently had statistically significant positive effects from being provided with tutoring and greater effect sizes for three out of the four tutoring strategies. These results agree with previous studies on the effectiveness of different tutoring strategies on different groups of students [42]. Additionally, Figure 5 shows that examples had the largest difference in their effectiveness between below and above average prior knowledge students and were the only tutoring strategy that had a statistically significant positive effect for below average prior knowledge students, but not for above average prior knowledge students.

Disparities in education, particularly in math, are often due to unequal access to opportunities to learn from highly qualified educators, otherwise known as the "opportunity gap" [17]. Although online learning platforms cannot replace a highly qualified educator, these results indicate that some online tutoring strategies can support in closing this opportunity gap for the most vulnerable students instead of just helping the more knowledgeable students succeed.

5.3 Other Instructional Interventions

Sections 5.1 and 5.2 covered two groups of related research questions, but there were many other research questions that did not fall into a group. Figure 6 shows the confidence intervals, number of participants, p-values, and effect sizes of these research questions. Of the various experiments, the ef-

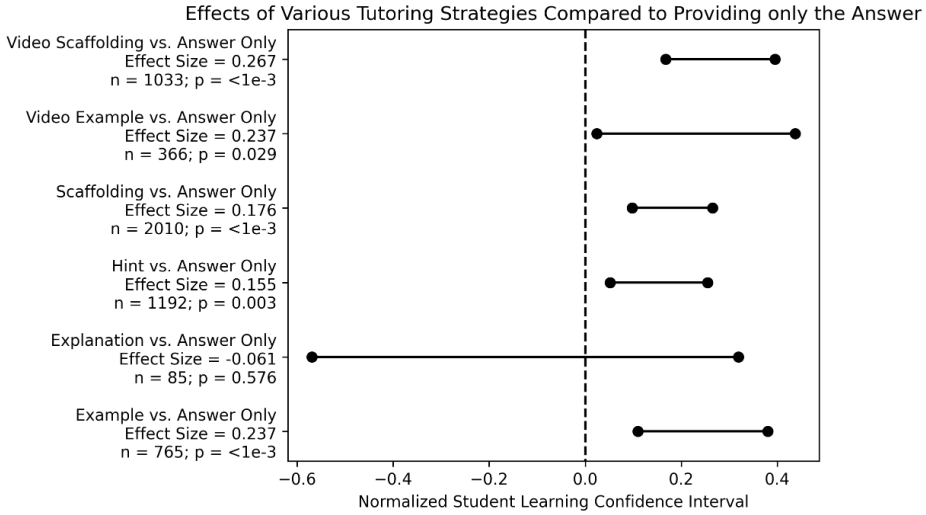


Figure 4: The effects of various tutoring strategies compared to providing only the answer. Effect size was determined using Cohen's d, the confidence interval and p-value come from the experiment condition model coefficient.

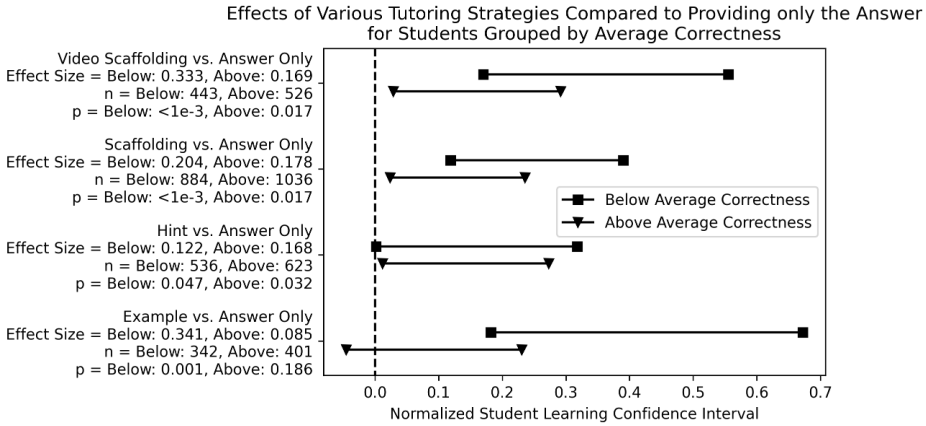


Figure 5: The effects of various tutoring strategies for below average and above average students. Effect size was determined using Cohen's d, the confidence interval and p-value come from the experiment condition model coefficient.

fect of giving video-based tutoring compared to text-based tutoring had the largest effect size, with students learning more from video-based tutoring than text-based tutoring. This could have been due to videos being more engaging and not requiring students to also be proficient readers. Additionally, giving students mathematics problems with open responses, where they are not given optional answers to choose from, resulted in more learning than when they were given problems with multiple choices. This could have been due to the added difficulty of attempting to answer a problem without knowing what the potential solutions are. Another significant finding was that adapting students' assignments based on their prior knowledge by altering the material given to them had a statistically significant positive effect, lending support to the idea that learning platforms should personalize students' learning based on their prior knowledge, which has been found to be true in various studies and meta-analyses [42, 26]. Lastly, it was found that motivational messages have a negative impact on students' learning. This could be a result of students finding the messages distracting. However, students' perceptions of the messages were not recorded as part of these experiments, and follow-up experiments should be performed to investigate further.

5.3.1 Video vs. Text

Although providing students with video-based tutoring instead of text-based tutoring resulted in an overall positive effect for all types of tutoring, it is possible this was due to a particularly large impact of receiving video instead of text for one type of tutoring strategy. Figure 7 shows the effect of providing video-based tutoring instead of text-based tutoring for the three types of tutoring strategies that were used in experiments where a video-based and text-based version of the same content was provided to students. Video-based scaffolding had the only significant positive effect on learning compared to a text-based control. Hints and examples had no statistically significant difference in their effective-

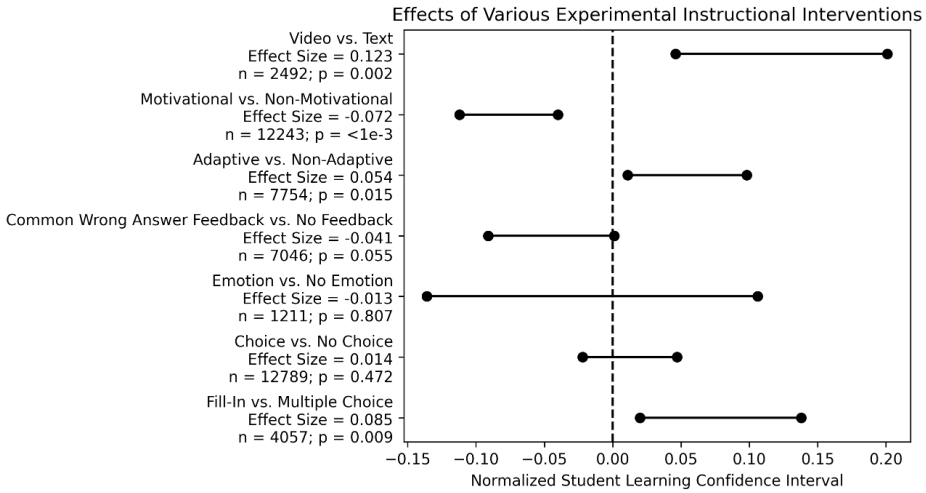


Figure 6: The effects of various experimental instructional interventions. Effect size was determined using Cohen's d, the confidence interval and p-value come from the experiment condition model coefficient.

ness when video-based or text-based. From these results, one can infer that students benefit differently from different types of tutoring being video-based. Scaffolding offers a series of simpler problems to help students understand the problem they are struggling with. It could be that students are more likely to engage with videos that give them necessary context. The scaffolding videos ask students questions that they must solve to move on, without watching the videos, they cannot know what the question is. Students may not be as willing to watch videos that provide relevant, but not entirely necessary information on a problem they must solve.

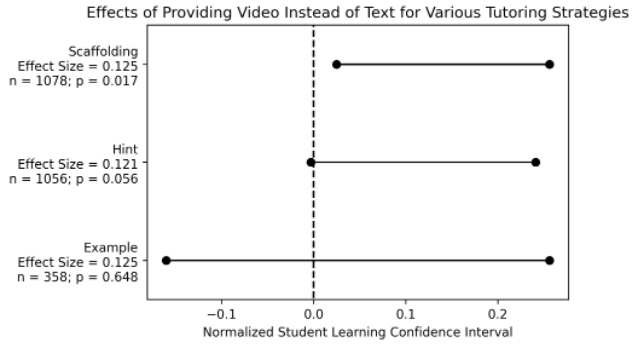


Figure 7: The effects of providing video instead of text for various tutoring strategies. Effect size was determined using Cohen's d, the confidence interval and p-value come from the experiment condition model coefficient.

6. LIMITATIONS

The results in this work help to reinforce a foundation of knowledge on educational experimentation and can be used to influence the next generation of experimentation, but there are two notable limitations to the extent to which these results can be applied in the future. Firstly, the scope of the experiments analyzed in this work is limited to experiments conducted within ASSISTments. It could be that the user interface of ASSISTments effects how beneficial certain instructional interventions are. For example, the way ASSISTments takes away partial credit for some tutoring strategies but full credit for others could impact the generalizability of these findings into a context where there is no scoring of student responses. All of the experiments also take place within skill builder assignments, in which students are given a series of similar problems on the same mathematics topic. The instructional interventions in the experiments analyzed in this work could have different effects on students completing assignments on topics outside of mathematics, or even a variety of mathematics topics within the same assignment. There could also be an issue generalizing these findings to contexts outside of online learning platforms. The differences between different tutoring strategies could be inconsequential if there is a teacher in the room to answer questions, and while the results of these experiments implied that motivational messages had a negative impact on learning, this was likely due to the distracting and impersonal nature of the motivational messages. Previous studies have shown the need for trust between teachers and their students and how this can lead to more motivated and academically successful students [4], but the trusting relationship needed for that impact is unlikely to exist between a student and a website.

Secondly, this work investigates many different research questions using data from a combination of experiments with similar, but not identical designs, which has increased the potential of discovering false positives in the analysis. This should influence the confidence that one has in the results of this work. While the results with effect sizes greater than 0.1 and p-values in range of of 10^{-3} can likely be trusted, there are many weaker findings that some might consider significant while others may be more critical. By providing the sample sizes, effect sizes, confidence intervals, and p-values for every comparison carried out, for all the research questions investigated in this work, others can make an informed decision on the extent to which they should believe each of these findings, and which findings merit follow-up investigations and repeat experiments.

7. CONCLUSION

In this work, data from 50,752 instances of one of 30,408 students participating in one of 50 different experiments on a variety of instructional interventions conducted within ASSISTments were combined to investigate their impact on learning. Using this data, 19 different research questions regarding the effectiveness of these various instructional interventions were investigated, and this investigation revealed multiple actionable findings that can be used to design more effective online learning experiences.

The first insight discovered was that changing the number of problems students must get correct in a row to be considered as having mastered a skill had no impact on the learning gains of the students that were able to complete the assignment, but the more problems required, the more likely students were to stop doing the assignment before mastering the material, which overall decreased their learning gains. Based on this result, when creating mastery-based content, it might be better to use something like Knowledge Tracing [13] to evaluate mastery instead of forcing students to complete a fixed number of problems.

It was also discovered that across multiple experiments, the tutoring provided to students by ASSISTments had almost entirely a positive effect on students' learning compared to just giving students the answer when they were struggling. This falls in line with the larger findings from cognitive psychology that show students learn more when they productively struggle with solving problems, rather than being provided solutions [6]. Additionally, below average prior knowledge students benefited more from this tutoring overall than above average prior knowledge students, which can help to close opportunity gaps, and for all students, when scaffolding problems were video-based, they had a larger positive impact than when they were text-based. These results could help inform developing platforms on how to allocate limited resources when creating tutoring. For example, creating new tutoring could be prioritized for remedial courses, and the extra effort of making video-based tutoring could be saved for scaffolding.

Another insight from these analyses was that students showed greater learning patterns when they completed open-response questions rather than multiple choice questions. This corroborates some research that finds that memory and learning benefit most from free recall of information (e.g. answering an open-ended question) compared to cued-recall (e.g. multiple-choice items) during learning [22, 30]. Based on this, online learning platforms could move away from multiple choice questions when possible.

This study also found that adjusting students' assignments based on their prior knowledge level had a positive effect on their learning. This supports the idea that personalized learning can help students. Within ASSISTments, a previous study found that high-knowledge students learned more from explanations, while low-knowledge students learned more from scaffolding [42]. This is one example of how personalization based on prior knowledge within online learning platforms has been found to be effective in the past. Additionally, a meta-analysis of studies that measured the learning gains of students after grouping them by ability level found that the instructional material was more than twice as effective when it was tailored to the students' ability levels than when it was held constant for all students [26]. The results of this study agree with prior work, and imply that personalizing students' education based on their prior knowledge increases their learning.

Another interesting result from these experiments was that motivational messages had a negative impact on learning. Past research has found positive effects of motivational interventions for some students, so why might these studies show a negative effect? One speculation is that the motivational videos may have unintentionally produced an effect similar to what is referred to as "seductive details" or highly engaging but unrelated information that is unnecessary for learning [46]. Including seductive details can lead to worse performance both in the classroom and in online learning environments [18], and is theorized to disrupt learning by redirecting attention away from the material and toward superfluous information, stopping students from appropriately allocating cognitive resources to the educational material. Providing motivational videos in the middle of the learning period may have produced a negative effect on learning because it disrupted cognitive processes necessary for learning, but more research is needed to fully investigate this and other possible mechanisms at play.

In addition to these results' capacity for improving online learning platforms, these results can help inform the next round of experimentation within online learning platforms. Future experiments could continue to investigate the inconclusive findings in this analysis, and expand upon the conclusive findings. For example, more types of problems besides multiple choice and open response problems could be compared to each other, and the effectiveness of different tutoring strategies could be investigated for differences based on subject matter or grade level. Through these analyses, learning platforms can continue to improve their design and increase their positive impact for all students that use them.

8. ACKNOWLEDGMENTS

We would like to thank NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, & R305A120125), GAANN (e.g., P200A180088 & P200A150306), EIR (U411B190024 & S411B210024), ONR (N00014-18-1-2768) and Schmidt Futures. None of the opinions expressed here are that of the funders.

9. REFERENCES

[1] D. Acemoglu, D. Laibson, and J. A. List. Equalizing superstars: The internet and the democratization of education. *American Economic Review*, 104(5):523–27, 2014.

[2] S. Adjei, K. Ostrow, E. Erickson, and N. Heffernan. Clustering students in assistments: exploring system-and school-level traits to advance personalization. In *The 10th International Conference on Educational Data Mining*, pages 340–341. ERIC, 2017.

[3] R. S. Baker, K. Yacef, et al. The state of educational data mining in 2009: A review and future visions.

Journal of educational data mining, 1(1):3–17, 2009.

[4] A. Bennett, B. L. Bridglall, A. M. Cauce, H. T. Everson, E. W. Gordon, C. D. Lee, R. Mendoza-Denton, J. S. Renzulli, and J. K. Stewart. All students reaching the top: Strategies for closing academic achievement gaps. a report of the national study group for the affirmative development of academic ability. *North Central Regional Educational Laboratory*, 2004.

[5] M. Bienkowski, M. Feng, and B. Means. Enhancing teaching and learning through educational data mining and learning analytics: An issue brief. *Office of Educational Technology, US Department of Education*, 2012.

[6] E. L. Bjork, R. A. Bjork, et al. Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning. *Psychology and the real world: Essays illustrating fundamental contributions to society*, 2(59-68), 2011.

[7] J. S. Bruner and C. C. Goodman. Value and need as organizing factors in perception. *The journal of abnormal and social psychology*, 42(1):33, 1947.

[8] P. C. Burnett and V. Mandel. Praise and feedback in the primary classroom: Teachers' and students' perspectives. *Australian Journal of Educational & Developmental Psychology*, 10:145–154, 2010.

[9] R. Butler. Enhancing and undermining intrinsic motivation: The effects of task-involving and ego-involving evaluation on interest and performance. *British journal of educational psychology*, 58(1):1–14, 1988.

[10] C. Candel, E. Vidal-Abarca, R. Cerdán, M. Lippmann, and S. Narciss. Effects of timing of formative feedback in computer-assisted learning environments. *Journal of Computer Assisted Learning*, 36(5):718–728, 2020.

[11] A. K. Clark and P. Whetstone. The impact of an online tutoring program on mathematics achievement. *The Journal of Educational Research*, 107(6):462–466, 2014.

[12] J. Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013.

[13] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[14] K. P. Cross. Feedback in the classroom: Making assessment matter. ERIC, 1988.

[15] S. Eastman and N. Kaeding. Opportunity zones: What we know and what we don't. *Tax Foundation Fiscal Fact*, 630, 2019.

[16] M. Feng, J. Roschelle, N. Heffernan, J. Fairman, and R. Murphy. Implementation of an intelligent tutoring system for online homework support in an efficacy trial. In *International Conference on Intelligent Tutoring Systems*, pages 561–566. Springer, 2014.

[17] A. Flores. Examining disparities in mathematics education: Achievement gap or opportunity gap? *The High School Journal*, 91(1):29–42, 2007.

[18] L. Fries, M. S. DeCaro, and G. Ramirez. The lure of seductive details during lecture learning. *Journal of Educational Psychology*, 111(4):736, 2019.

[19] D. A. Gilman. Comparison of several feedback methods for correcting errors by computer-assisted instruction. *Journal of Educational Psychology*, 60(6p1):503, 1969.

[20] N. T. Heffernan and C. L. Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[21] A. T. Jersild. Examination as an aid to learning. *Journal of Educational Psychology*, 20(8):602, 1929.

[22] J. D. Karpicke. Retrieval-based learning: A decade of progress. *Grantee Submission*, 2017.

[23] A. N. Kluger and A. DeNisi. The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological bulletin*, 119(2):254, 1996.

[24] D. R. Krathwohl. A revision of bloom's taxonomy: An overview. *Theory into practice*, 41(4):212–218, 2002.

[25] N. J. Krichevsky and K. P. Spinelli. E-trials: Developing a web application for educational research. 2020.

[26] C.-L. C. Kulik and J. A. Kulik. Effects of ability grouping on secondary school students: A meta-analysis of evaluation findings. *American educational research journal*, 19(3):415–428, 1982.

[27] P. McGuire, S. Tu, M. E. Logue, C. A. Mason, and K. Ostrow. Counterintuitive effects of online feedback in middle school math: results from a randomized controlled trial in assistments. *Educational Media International*, 54(3):231–244, 2017.

[28] K. V. Middleton. The longer-term impact of covid-19 on k–12 student learning and assessment. *Educational Measurement: Issues and Practice*, 39(3):41–44, 2020.

[29] M. Montessori. *The advanced Montessori method*, volume 1. Frederick A. Stokes Company, 1917.

[30] B. F. T. Moreira, T. S. S. Pinto, D. S. V. Starling, and A. Jaeger. Retrieval practice in classroom settings: a review of applied research. In *Frontiers in Education*, page 5. Frontiers, 2019.

[31] R. Moreno. Decreasing cognitive load for novice students: Effects of explanatory versus corrective feedback in discovery-based multimedia. *Instructional science*, 32(1):99–113, 2004.

[32] S. Narciss, S. Sosnovsky, L. Schnaubert, E. Andrès, A. Eichelmann, G. Goguadze, and E. Melis. Exploring feedback and student characteristics relevant for personalizing feedback strategies. *Computers & Education*, 71:56–76, 2014.

[33] T. Nguyen. The effectiveness of online learning: Beyond no significant difference and future horizons. *MERLOT Journal of Online Learning and Teaching*, 11(2):309–319, 2015.

[34] K. Ostrow and N. Heffernan. Testing the multimedia principle in the real world: a comparison of video vs. text feedback in authentic middle school math assignments. In *Educational Data Mining 2014*, 2014.

[35] J. M. Parr and L. Limbrick. Contextualising practice: Hallmarks of effective teachers of writing. *Teaching and Teacher Education*, 26(3):583–590, 2010.

[36] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*, 2009.

[37] A. G. Picciano, J. Seaman, P. Shea, and K. Swan. Examining the extent and nature of online learning in american k-12 education: The research initiatives of the alfred p. sloan foundation. *The internet and higher education*, 15(2):127–135, 2012.

[38] E. Prihar, A. Botelho, M. Corace, A. Shanaj, Z. Dia, and N. T. Heffernan. Student engagement during remote learning. In *Companion Proceedings 11th International Conference on Learning Analytics Knowledge*, pages 49–51, 2021.

[39] E. Prihar, T. Patikorn, A. Botelho, A. Sales, and N. Heffernan. Toward personalizing students' education with crowdsourced tutoring. In *Proceedings of the Eighth ACM Conference on Learning@ Scale*, pages 37–45, 2021.

[40] L. Razzaq and N. T. Heffernan. Scaffolding vs. hints in the assistment system. In *International Conference on Intelligent Tutoring Systems*, pages 635–644. Springer, 2006.

[41] L. M. Razzaq, M. Feng, G. Nuzzo-Jones, N. T. Heffernan, K. R. Koedinger, B. Junker, S. Ritter, A. Knight, E. Mercado, T. E. Turner, et al. Blending assessment and instructional assisting. In *AIED*, pages 555–562, 2005.

[42] L. M. Razzaq and N. T. Heffernan. To tutor or not to tutor: That is the question. In *AIED*, pages 457–464, 2009.

[43] W. Roper. Feedback in computer assisted instruction. *Programmed learning and educational technology*, 14(1):43–49, 1977.

[44] D. Selent, T. Patikorn, and N. Heffernan. Assistments dataset from multiple randomized controlled experiments. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 181–184, 2016.

[45] M. H. Smits, J. Boon, D. M. Sluijsmans, and T. Van Gog. Content and timing of feedback in a web-based learning environment: Effects on learning as a function of prior knowledge. *Interactive Learning Environments*, 16(2):183–193, 2008.

[46] N. Sundararajan and O. Adesope. Keep it coherent: A meta-analysis of the seductive details effect. *Educational Psychology Review*, 32(3):707–734, 2020.

[47] P. Suppes et al. Computer-assisted instruction: Stanford's 1965-66 arithmetic program. 1968.

[48] L. Voerman, P. C. Meijer, F. A. Korthagen, and R. J. Simons. Types and frequencies of feedback interventions in classroom interaction in secondary education. *Teaching and teacher education*, 28(8):1107–1115, 2012.

[49] B. J. Zimmerman and D. H. Schunk. *Educational psychology: A century of contributions: A Project of Division 15 (educational Psychology) of the American Psychological Society*. Routledge, 2014.

Chapter 2.4

Comparing Different Approaches to Generating Mathematics Explanations Using Large Language Models

Comparing Different Approaches to Generating Mathematics Explanations Using Large Language Models *

Ethan Prihar[1 [0000−0002−5216−9815]], Morgan Lee[1 [0000−0002−9839−9608]], Mia Hopman[1 [0000−0001−8267−5455]], Sofia Vempala[1 [0000−0002−5855−4503]], Allison Wang[1 [0000−0002−4646−7621]], Gabriel Wickline[1], Adam Tauman Kalai[2 [0000−0002−4559−8574]], and Neil Heffernan[1 [0000−0002−3280−288X]]

[1] Worcester Polytechnic Institute, Worcester, MA 01609, USA
{ebprihar,mplee,mahopman,svempala,awang9,gwickline,nth}@wpi.edu
[2] Microsoft Research, 1 Memorial Dr, Cambridge, MA 02142, USA
adam.kalai@microsoft.com

Abstract. Large language models have recently been able to perform well in a wide variety of circumstances. In this work, we explore the possibility of large language models, specifically GPT-3, to write explanations for middle-school mathematics problems, with the goal of eventually using this process to rapidly generate explanations for the mathematics problems of new curricula as they emerge, shortening the time to integrate new curricula into online learning platforms. To generate explanations, two approaches were taken. The first approach attempted to summarize the salient advice in tutoring chat logs between students and live tutors. The second approach attempted to generate explanations using few-shot learning from explanations written by teachers for similar mathematics problems. After explanations were generated, a survey was used to compare their quality to that of explanations written by teachers. We test our methodology using the GPT-3 language model. Ultimately, the synthetic explanations were unable to outperform teacher written explanations. In the future more powerful large language models may be employed, and GPT-3 may still be effective as a tool to augment teachers' process for writing explanations, rather than as a tool to replace them. The explanations, survey results, analysis code, and a dataset of tutoring chat logs are all available at https://osf.io/wh5n9/.

Keywords: Large Language Models · GPT-3 · Online Learning · Tutoring

* We would like to thank NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, & R305A120125), GAANN (e.g., P200A180088 & P200A150306), EIR (U411B190024 & S411B210024), ONR (N00014-18-1-2768), NHI (R44GM146483), and Schmidt Futures. None of the opinions expressed here are that of the funders.

1 Introduction

Online learning platforms offer students tutoring in a variety of forms, such as one-on-one messaging with real human tutors [1] or providing expert-written messages for each question that students are required to answer [8]. These methods, while effective, can be costly and time consuming to scale. However, recent advances in Language Models (LMs) may provide an opportunity to offset the cost of providing effective tutoring to students.

In this work, we explore the effectiveness of using LMs to create explanations of mathematics problems for students within the ASSISTments online learning platform [8]. Recent transformer-based LMs have exhibited breakthrough performance on a number of domains [3,4]. In this work, we perform experiments using one of the most powerful currently available LMs, GPT-3 [3], accessed through OpenAI's API.

Two different approaches to generate this content were explored. The first approach used few-shot learning [3] to generate new explanations from a handful of similar mathematics problems with answers and explanations, and the second approach attempted to generate new explanations by using the LM to summarize message logs between students and real human tutors. After each method was used to generate new explanations, these explanations were compared to existing explanations in the ASSISTments online learning platform through surveys given to mathematics teachers. Comparing teachers' evaluations of the quality of the various explanations enabled an empirical evaluation of each LM-based approach, as well as an evaluation of their applicability in a real-world setting.

To summarize, in this work we evaluate the following research questions:

1. What is the most effective way to use an LM to create explanations from existing mathematics problems and their answers and explanations?
2. What is the most effective way to use an LM to create explanations from chat logs between students and tutors?
3. How effective are these methods compared to expert-written explanations?

While the explanations we generated using GPT-3 were not rated as highly as those generated by humans, our approach and evaluation methodology is general and could easily be applied to future LMs.

2 Background

2.1 Language models

Recent neural LMs are a type of deep learning model trained to generate human-like text. They are trained on a massive dataset of millions of web pages, books, and other written documents, and are capable of generating text that is often indistinguishable from human-written text [3,4]. In this work, we focus on GPT-3 since it is a powerful LM that is publicly accessible through a paid API. When using GPT-3, one can specify parameters for the text generation such as

Frequency Penalty, which penalizes GPT-3 for repeating phrases in its response, Temperature, which increases the frequency of picking a less-that-most-likely word to include in the response, and Max Tokens, which specifies the maximum length of the response [3].

Prior work has demonstrated the ability of LMs to summarize a wide variety of information. Most relevant is work specifically on summarizing chat logs between students [15]. In that work, chat logs between students as they discussed strategies during a collaborative game were summarized by an LM for teachers to review. The LM was able to summarize the strategies and opinions of students, as well as students' affect. LMs have also been able to summarize teachers responses in classroom simulations using few-shot learning [11]. Given that LMs have been successful in a variety of fields and specifically with summarizing dialogue between students and teacher responses, it is likely that LMs would be able to summarize chat logs between a tutor and a student.

LMs have also been used to solve mathematics problems at a middle-school [20] and college [9,6] level. Beyond solving problems, LMs have been used to generate explanations for problems, though most commonly computer science problems and code explanations [18,12]. Due to LMs' wide applicability in these domains, it seems likely that LMs would be capable of writing explanations for existing mathematics problems when provided with the text of the problem, the correct answer, and examples of explanations written for similar mathematics problems.

2.2 ASSISTments

The data used to generate explanations using few-shot learning came from the ASSISTments online learning platform [8]. Within ASSISTments, middle-school mathematics students complete mathematics problem sets assigned to them by their teacher. If students are struggling with their assignment, ASSISTments will provide them with an explanation upon their request. When a student requests an explanation, a message that explains how to solve the mathematics problem they are currently struggling with and the solution to the problem is provided to them. Explanations in ASSISTments have been crowdsourced from graduate students, teachers, and researchers [13,16], and therefore take on a wide variety of structures and formats.

2.3 Live Online Tutoring

The data used to generate explanations from summaries of tutoring chat logs comes from a provider of online tutoring in partnership with ASSISTments. Organizations like Yup[3], Tutor.com[4], and UPchieve[5] offer online tutoring to students. Typically, students use these services by logging into the platform and

[3] https://yup.com/
[4] https://www.tutor.com/
[5] https://upchieve.org/

requesting a session with a tutor, at which point they are connected to a volunteer or payed tutor and placed into a tutoring session. Within these tutoring sessions, students can chat with the tutor via text, or communicate via a virtual white board.

Recently, ASSISTments partnered with a live tutoring provider and, for some students, replaced the ability to request an explanation with the ability to chat with a live tutor. When a live tutor was requested, a tutoring session was opened via the live tutoring provider. This new feature provided the opportunity to compare explanations generated by an LM using both a few-shot learning approach with existing explanations and a summarization approach using the tutoring chat logs for the same mathematics problems.

3 Data Processing

The style and format of the text generated by LMs is highly dependent on subtle changes in the prompt used to generate it. There have been many studies of how to properly engineer a prompt for LMs [17,5,19]. In order to examine the effects of changes to the prompts on the generate explanations in a way that would not bias the results of the analysis, all of the available data for generating explanations was split in half. Half of the data was used for prompt engineering (development set). This data was used iteratively to examine how small variations in the prompt effected the resulting explanations. Once the generated explanations reached a satisfactory level, the most effective prompts were used on the second half of the data (evaluation set). The analysis of the validity and quality of explanations discussed in the results was performed only on this second half of the data, eliminating any bias from the prompt engineering process.

3.1 Tutoring Chat Logs

During the live tutoring partnership period, there were 244 tutoring sessions across 93 students and 110 problems covering various middle-school mathematics skills. Of these tutoring sessions, 2 were excluded because they contained no interaction between student and tutor (the student never responded to a tutor's opening question) and 2 were excluded because they were longer than GPT-3's 4,000 token limit. The remaining 240 logs were randomly split into development and evaluation sets based on student IDs. While student IDs were used to split the data, we wanted to ensure the datasets would be similar, and verified that both sets contained problems of the same skills in similar proportions. Information on problems' skills was provided by ASSISTments, using the Common Core State Standards for Mathematics [2]. The split we generated was mostly balanced, except for examples where a particular student provided the only example of a particular skill. These instances were placed in the evaluation set. Overall, there were 121 chat logs from 45 students answering 53 problems in the development set, and 119 chat logs from 48 students answering 61 problems in the evaluation set.

3.2 Problem-Level Explanations

To prepare ASSISTments data for few-shot learning, only problems and explanations from the Engage New York[6] and Illustrative Mathematics[7] curricula were considered because within ASSISTments, those two curricula are the most popular and have the most explanations. Only non-open response problems and explanations that were fully text-based could be used for few-shot learning because few-shot learning required the problem, answer, and explanation to be in the prompt. Additionally, some ASSISTments problems were excluded because they were follow-up problems to previous problems and did not provide all the context necessary to solve the problem. Of the 40,523 problems and 22,944 explanations available, only 9,200 problems and 11,345 explanations remained after removing problems that could not be used in the few-shot learning prompt. These problems and their explanations were evenly partitioned into a development and evaluation set as well, stratified by problem skills.

In order to compare few-shot learning based explanations to summarization based explanations, the few-shot learning approach was used only to generate explanations for the problems that were discussed within the tutoring chat logs. Problems with skills different from the problems in the tutoring chat logs were removed, leaving 315 development problems and 599 evaluation problems for the few-shot learning approach.

4 Methodology

4.1 Tutoring Chat Log Summarization

Development data was used to engineer a four step process for generating explanations from tutoring chat logs. The prompts are shown below, with the GPT-3 parameters shown in parentheses as (Frequency Penalty, Temperature, Max Tokens). The text-davinci-003 model was used for all prompts.

1. Does the the tutor successfully help the student in the following chain of messages? [The tutoring chat log.] (0, 0.7, 128)
2. Explain the mathematical concepts the tutor used to help the student, including explanations the tutor gave of these concepts, and ignoring any names. [The tutoring chat log.] (0.25, 0.9, 750)
3. Reword the following explanation to not include references to a tutor or student, and to be in the present tense: [The previously generated explanation.] (0.25, 0.9, 750)
4. Summarize the following explanations, making sure to include the most generalizable math advice. [The previously generated explanations.] (0.25, 0.9, 500)

[6] http://www.nysed.gov/curriculum-instruction/engageny
[7] https://illustrativemathematics.org/

Step 1 asked GPT-3 to evaluate the initial tutoring chat log to determine if the tutor provided help to the student. This step was initially broken into two steps: one which asked if the tutor provided mathematical help to the student and one to ask if the mathematical help was useful to the student. This two step evaluation of helpfulness proved to be too restrictive, as during the initial prompt engineering phase, about 60% of the original chat logs were excluded, even though some contained valid mathematics advice. The prompt was then changed to evaluate only if the tutor helped the student. This only filtered out message chains where little information was transferred between tutor and student, resulting in about 20% of chat logs being deemed unhelpful and discarded. Step 2 asked GPT-3 to summarize the mathematical help given by the tutor to the student. The outputs generated by GPT-3 at this stage contained mathematical content, but were worded as a past-tense summary of the interaction between tutor and student. To refine these summaries, Step 3 was added, which asked GPT-3 to reword the output of Step 2 into a present tense explanation by removing references to the interactions between tutor and student. Finally, some problems had multiple tutoring chat logs discussing them. In order to generate a single explanation per problem, a final step was added to summarize all the previously generated explanations for a problem when more than one generated explanation existed.

4.2 Problem-Level Explanation Few-Shot Learning

Before generating explanations for the 53 problems in the summarization development set, problems that were open response or not text-based had to be removed. After this, 40 problems remained. This was deemed to be an acceptable level of loss, and no steps were taken to try an include problems with images in the few-shot learning approach. For each of the 40 remaining problems a prompt was constructed by randomly sampling problems of the same skill from the development set, and appending the phrase below, replacing the content in brackets with the problem content.

Problem: [The text of the problem.]
Answer: [The answer to the problem.]
Explanation: [The explanation for the problem.]

A problem was considered to be of the same skill as another if the grade level and subject were the same. This was decided because if the entire Common Core Skill Code had to be identical, there would not have been enough problems for prompt generation. At the end of the prompt, the phrase above was used for the problem for which an explanation was being generated, but nothing was added for the explanation, allowing for GPT-3 to fill in the explanation. Due to the 4,000 token prompt limit, if including all the related problems in the prompt made the prompt over 11,523 characters long, related problems were randomly removed from the prompt until the prompt was less than 11,523 characters long, which was determined, using the development set, to approximate the 4,000

token limit. For these prompts, the Frequency Penalty was 0, the Temperature was 0.73, the Max Tokens was 256, and the code-davinci-003 model was used.

4.3 Empirical Analysis of Generated Explanations

After the summarization and few-shot learning processes were completed for the evaluation data using the processes developed with the development data. The explanations from both processes were manually evaluated by subject-matter experts for both structural and mathematical correctness. Structural correctness required that the explanation generated by GPT-3 be in the format of a mathematics explanation. For example, if GPT-3 generated the explanation "Go take a walk, then come back and try again.", that would be structurally incorrect. Mathematical correctness refers to whether or not the explanation given by GPT-3 is mathematically correct. For example, if GPT-3 generated the explanation "To solve for x in the equation $x + 3 = 5$, subtract 3 from both sides of the equation, which gives you $x = 3$.", that would be structurally correct because it is in the format of a mathematics explanation, but mathematically incorrect, because $x = 2$, not 3.

After structurally or mathematically incorrect explanations were removed by experts, the remaining explanations were mixed with any existing explanations already in ASSISTments written by teachers for the same problems. The source of the explanations from summarization, few-shot learning, and teachers was blinded, and mathematics teachers were given a picture of each mathematics problem and the text of the explanation and told to rate, on a scale from 1-5[10], how likely it is that the explanation would help a student. Mathematics teachers have proven in the past to be effective creators of explanations for ASSISTments [13], therefore, they are likely reliable evaluators of the quality of this content. After collecting all of the teachers' survey results, the correlations between the teachers ratings were calculated to examine the inter-rater reliability of the survey results. A Pearson correlation [14] matrix was used to examine the similarity of different teachers' results. A correlation matrix was used because it allowed for the explanation ratings to be treated as continuous variables. By treating the ratings as continuous, as opposed to a categorical variable with five categories, teachers that were more or less strict with what they considered to be an excellent explanation would still have positive correlations as long as they generally agreed on how good the explanations were relative to other explanations. Additionally, only a small number of teachers were expected to participate in the survey, making the correlation matrix easily interpretable. Additionally, the correlation matrix had the potential to reveal different modes of thought among teachers, i.e., there could be clusters of teachers with similar opinions that differ from other clusters of teachers.

Once the survey results were deemed consistent, a multi-level model [7] was used to predict the rating of each explanation given random effects for the rater and the mathematics problem, and fixed effects for the source of the explanation. Random effects were used to compensate for low sample sizes while still taking into account the differences between raters, problems, and the effects that those

differences had on the ratings of explanations. The effects for the sources of explanations were used to determine if there were any statistically significant differences between the sources.

5 Results

5.1 Summarization

Performing the summarization process on the evaluation data resulted in 26 acceptable explanations. Initially, there were 119 chat logs across 61 problems. Step 1 of the summarization process removed 14 tutoring logs, resulting in 105 explanations across 57 problems. The complete process generated 57 explanations. Expert review of the final explanations found 14 invalid explanations due to bad structure and 17 due to incorrect mathematics, which is only about a 43% success rate. The 26 valid summarization based explanations were included in the explanation quality survey.

5.2 Few-Shot Learning

Performing the few-shot learning process on the evaluation data resulted in only 6 acceptable explanations. 28 of the initial 61 evaluation problems were removed because they were not solely text-based. Of the 33 remaining problems, 1 was invalid due to bad structure, and 26 due to incorrect mathematics, which is only about a 10% generation success rate. The 6 valid few-shot learning based explanations were included in the explanation quality survey.

5.3 Empirical Analysis of Generated Explanations

After both procedures for generating explanations using GPT-3 were complete, and the structurally or mathematically incorrect explanations were removed, any explanations for 61 problems in the evaluation set that were already written by teachers for ASSISTments were combined with the remaining GPT-3 generated explanations. In total, 26 summarization, 6 few-shot learning, and 10 ASSISTments explanations were included for a total of 42 survey questions. Five current or former middle-school or high-school mathematics teachers completed the survey. The correlation between all the teacher's ratings is shown in Figure 1, where each teacher is anonymized as a letter of the alphabet, and the value in the cell shows the correlation between the row and column teachers' ratings. Although some teachers had a low correlation between their ratings, no teachers had a negative correlation between their ratings. Teachers A, C, and E have the highest correlation with each other, while Teachers B and D were less correlated with other teachers. Although some teachers were more or less strict than others. which lowered their correlations, in general, teachers agreed on what makes an explanation good or bad.

Once the inter-rater reliability was deemed sufficient, a multi-level model [7] was fit with random effects for the rater and the mathematics problem, and fixed

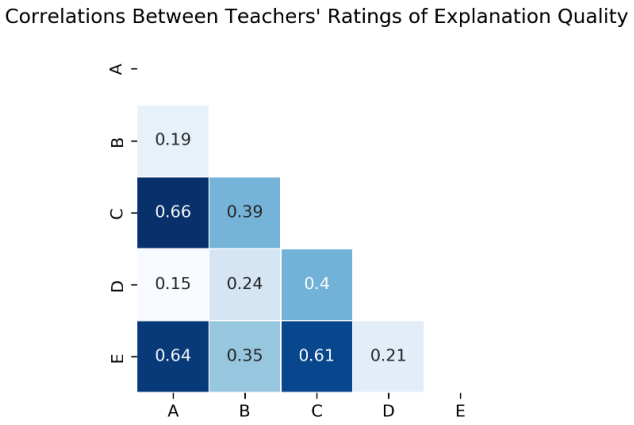


Fig. 1. The correlations between all teachers ratings of explanation quality, determined using the survey results.

effects for the source of the explanation. Two different models were fit, one that only included teachers ratings of valid explanations, and one that included all the generated explanations, with a rating of 1 for explanations that were invalid. The effects and 95% confidence intervals of the different sources of explanations are shown in Figure 2. ASSISTments explanations are rated the highest, with an average rating of about 4.2. It is unsurprising that the explanations written by teachers were the most highly rated. Summarization based explanations were statistically significantly worse than ASSISTments explanations, with an average rating of about 2.6 for the valid explanations and 1.7 for all explanations. Qualitatively, teachers reported that the summarization based explanations used terms that the students did not necessarily know, and tended to give advice that was too general. Few-shot learning based explanations received an average rating of about 3.6 for valid explanations, which was not statistically significantly worse than ASSISTments explanations, but only 6 of the few-shot learning based explanations were valid. If the invalid explanations are included in the analysis, then few-shot learning based explanations received an average rating of about 1.6, which is statistically significantly worse than ASSISTments explanations.

6 Limitations and Future Work

While this study makes it apparent that GPT-3's explanations are worse than teacher-written explanations, it is limited to just middle school mathematics problems. A difficult part of generating explanations for simple mathematics problems is that often GPT-3 writes explanations with the assumption that fundamental mathematics concepts are already known. Based on the success that other studies have had using GPT-3 to interpret college level mathematics [6],

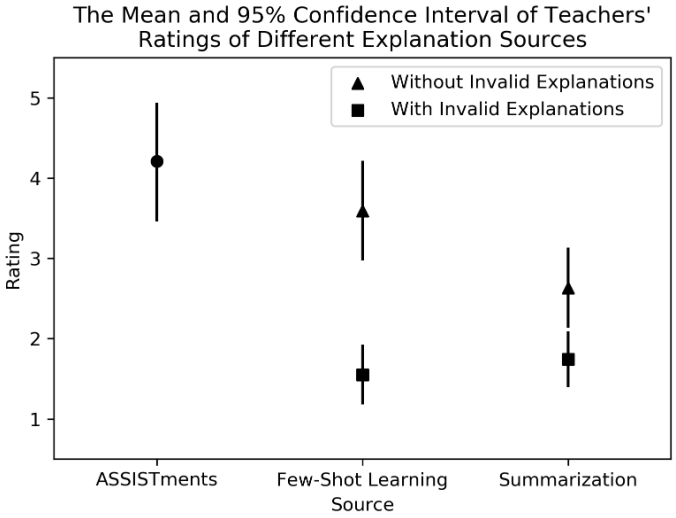


Fig. 2. The mean and 95% confidence interval of teachers' ratings of explanation quality by source, determined using the survey results. Invalid explanations, when included in the model, are assumed to have the lowest rating for quality.

it may be, for example, easier for an LM to understand integrals than scientific notation because there is far more language used in the descriptions and use cases of integrals than there is in the description of scientific notation, which is just a simple mathematics operation.

Additionally, there is no closed-form solution for prompt engineering. To avoid bias, a development set was used to create prompts via trial and error, but there is no guarantee that this work constructed the best prompts for generating explanations from tutoring chat-logs or from similar problems and their answers and explanations. Even with a four-stage process for summarizing tutoring chat logs, a better, more concise prompt might be achievable when approaching the generation process differently. More work to explore and refine the prompts used to generate explanations could be done to better understand how to get the best content from an LM.

Even if one assumes that there exists a prompt that would generate effective explanations of mathematics problems, the entire process is still limited to purely text-based content. Many mathematics problems use diagrams or equations to represent information. Without the ability to interpret this information, the capacity to use an LM to create explanations will be limited to a small subset of mathematics curricula. In the short term, efforts should be made to algorithmically generate text alternatives to mathematics diagrams and equations. Then, these text alternatives could be substituted for the diagrams and equations they represent. In the long term, a large LM capable of interpreting

mathematics, or even logic, could have a tremendous impact on the quality of the content generated from the model.

7 Conclusion

Overall it seems that GPT-3 based explanations do not compare in quality to those created by teachers. Fundamentally, GPT-3 was trained to understand language, but not mathematics, and while the structure of what GPT-3 generated made proper use of the English language, it often generated incorrect mathematical content, or simply failed to generate content in the proper format. When summarizing tutors' advice to students, four different prompts had to be used before the content began to resemble an explanation suitable for integration into an online learning platform, and even after removing invalid explanations, the explanations that were both structurally and mathematically valid were statistically significantly worse in quality than teacher-written explanations. The valid explanations generated through few-shot learning were not statistically significantly worse than teacher-written explanations, but only 10% of the generated explanations were valid. Almost all the other explanations were mathematically invalid.

Ultimately, the latest version of GPT-3 does not seem to have the grasp of mathematics necessary to generate high-quality explanations, but it has other strengths that should be taken advantage of. There are likely much more effective use cases where GPT-3 can improve online learning. Interpreting student affect or identifying the sentiment, emotional, or motivational content in tutoring chat logs all seem like tasks that GPT-3 is more applicable to than explanation generation, and all of these tasks could be used to study and improve students' experiences within online learning platforms. Additionally, Larger language models which perform better on tasks, including mathematical tasks, are being released with increasing frequency. For instance, the PaLM LM, with 540B parameters, is reported to outperform GPT-3 with 175B parameters on a number of tasks [4]. While we do not have access to this LM, our methodology can be applied to future more powerful LMs.

References

1. Upchieve's mission, https://upchieve.org/mission
2. Akkus, M.: The common core state standards for mathematics. International Journal of Research in Education and Science **2**(1), 49–54 (2016)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
4. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J.,

Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A.M., Pillai, T.S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., Fiedel, N.: Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022), https://arxiv.org/abs/2204.02311

5. Denny, P., Kumar, V., Giacaman, N.: Conversing with copilot: Exploring prompt engineering for solving cs1 problems using natural language. arXiv preprint arXiv:2210.15157 (2022)
6. Drori, I., Zhang, S., Shuttleworth, R., Tang, L., Lu, A., Ke, E., Liu, K., Chen, L., Tran, S., Cheng, N., et al.: A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. Proceedings of the National Academy of Sciences **119**(32), e2123433119 (2022)
7. Gelman, A., Hill, J.: Data analysis using regression and multilevel/hierarchical models. Cambridge university press (2006)
8. Heffernan, N.T., Heffernan, C.L.: The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. International Journal of Artificial Intelligence in Education **24**(4), 470–497 (2014)
9. Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., Misra, V.: Solving quantitative reasoning problems with language models (2022). https://doi.org/10.48550/ARXIV.2206.14858, https://arxiv.org/abs/2206.14858
10. Likert, R.: A technique for the measurement of attitudes. Archives of psychology (1932)
11. Littenberg-Tobias, J., Marvez, G., Hillaire, G., Reich, J.: Comparing few-shot learning with gpt-3 to traditional machine learning approaches for classifying teacher simulation responses. In: International Conference on Artificial Intelligence in Education. pp. 471–474. Springer (2022)
12. MacNeil, S., Tran, A., Mogil, D., Bernstein, S., Ross, E., Huang, Z.: Generating diverse code explanations using the gpt-3 large language model. In: Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 2. pp. 37–39 (2022)
13. Patikorn, T., Heffernan, N.T.: Effectiveness of crowd-sourcing on-demand assistance from teachers in online learning platforms. In: Proceedings of the Seventh ACM Conference on Learning@ Scale. pp. 115–124 (2020)
14. Pearson, K.: Vii. note on regression and inheritance in the case of two parents. proceedings of the royal society of London **58**(347-352), 240–242 (1895)
15. Phillips, T., Saleh, A., Glazewski, K.D., Hmelo-Silver, C.E., Mott, B., Lester, J.C.: Exploring the use of gpt-3 as a tool for evaluating text-based collaborative discourse. Companion Proceedings of the 12th International Conference on Learning Analytics Knowledge (LAK22) p. 54 (2022)
16. Prihar, E., Syed, M., Ostrow, K., Shaw, S., Sales, A., Heffernan, N.: Exploring common trends in online educational experiments. In: Proceedings of the 15th International Conference on Educational Data Mining. p. 27 (2022)
17. Reynolds, L., McDonell, K.: Prompt programming for large language models: Beyond the few-shot paradigm. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. pp. 1–7 (2021)

18. Sarsa, S., Denny, P., Hellas, A., Leinonen, J.: Automatic generation of programming exercises and code explanations using large language models. In: Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1. pp. 27–43 (2022)
19. Zhou, Y., Muresanu, A.I., Han, Z., Paster, K., Pitis, S., Chan, H., Ba, J.: Large language models are human-level prompt engineers. arXiv preprint arXiv:2211.01910 (2022)
20. Zong, M., Krishnamachari, B.: Solving math word problems concerning systems of equations with gpt-3. In: Proceedings of the Thirteenth AAAI Symposium on Educational Advances in Artificial Intelligence (2022)

Chapter 3
Multi-Armed Bandit Integration and Design

The following research papers all revolve around integrating multi-armed bandit-based recommendation algorithms into ASSISTments. These papers investigate the impact that reinforcement learning has at scale within online learning platforms, and attempts to personalize students' learning, or at least gain insight into ways that students' learning could be personalized in the future.

The first paper, "**Automatic Interpretable Personalized Learning**", covers the design of a service for ASSISTments which receives requests for support, and uses a multi-armed bandit algorithm to select the support most likely to help the student, and returns the selected tutoring to ASSISTments. I also present a novel algorithm that combines Thompson sampling with decision trees, which can make contextual recommendations in an interpretable way. This full paper was published at L@S 2022.

The second paper, "**Investigating the Impact of Skill-Related Videos on Online Learning**", uses the service developed in the first paper to recommend skill-related videos to students. These videos came from YouTube and were aggregated via algorithmic searches and validated through crowdsourcing of teacher opinions. This paper provided more insight into the effectiveness of multi-armed bandits when used to recommend other types of content. Ultimately, students did not like the skill-related videos, preferring more specific instruction. This gives valuable insight into where personalization is most effective. This full paper has been submitted to L@S 2023.

The third paper, "**A Bandit you can Trust**", developed a novel contextual multi-armed bandit algorithm that was used at scale to recommend skill-related videos to students. The goal of this work was to develop an algorithm that could both personalize students' learning, and provide unbiased statistical insight into the relationships between features of students and features of the content available to them. Ultimately the algorithm developed in this work succeeded at effectively personalizing content for students while allowing for unbiased statistical analysis. This work can be used to refine other algorithms in similar ways to continue the trend of creating interpretable, unbiased, and effective methods of personalizing students' learning. This full paper has been submitted to UMAP 2023.

Chapter 3.1

Automatic Interpretable Personalized Learning

Automatic Interpretable Personalized Learning

Ethan Prihar
ebprihar@wpi.edu
Worcester Polytechnic Institute
Worcester, MA, USA

Aaron Haim
ahaim@wpi.edu
Worcester Polytechnic Institute
Worcester, MA, USA

Adam Sales
asales@wpi.edu
Worcester Polytechnic Institute
Worcester, MA, USA

Neil Heffernan
nth@wpi.edu
Worcester Polytechnic Institute
Worcester, MA, USA

ABSTRACT

Personalized learning stems from the idea that students benefit from instructional material tailored to their needs. Many online learning platforms purport to implement some form of personalized learning, often through on-demand tutoring or self-paced instruction, but to our knowledge none have a way to automatically explore for specific opportunities to personalize students' education nor a transparent way to identify the effects of personalization on specific groups of students. In this work we present the Automatic Personalized Learning Service (APLS). The APLS uses multi-armed bandit algorithms to recommend the most effective support to each student that requests assistance when completing their online work, and is currently used by ASSISTments, an online learning platform. The first empirical study of the APLS found that Beta-Bernoulli Thompson Sampling, a popular and effective multi-armed bandit algorithm, was only slightly more capable of selecting helpful support than randomly selecting from the relevant support options. Therefore, we also present Decision Tree Thompson Sampling (DTTS), a novel contextual multi-armed bandit algorithm that integrates the transparency and interpretability of decision trees into Thomson sampling. In simulation, DTTS overcame the challenges of recommending support within an online learning platform and was able to increase students' learning by as much as 10% more than the current algorithm used by the APLS. We demonstrate that DTTS is able to identify qualitative interactions that not only help determine the most effective support for students, but that also generalize well to new students, problems, and support content. The APLS using DTTS is now being deployed at scale within ASSISTments and is a promising tool for all educational learning platforms.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; • **Theory of computation** → **Online learning algorithms**.

L@S '22, June 1–3, 2022, New York City, NY, USA.

ACM ISBN 978-1-4503-9158-0/22/06.
https://doi.org/10.1145/3491140.3528267

KEYWORDS

Contextual Bandits, Multi-Armed Bandits, Personalized Learning, Intelligent Tutoring Systems

ACM Reference Format:
Ethan Prihar, Aaron Haim, Adam Sales, and Neil Heffernan. 2022. Automatic Interpretable Personalized Learning. In *Proceedings of the Ninth ACM Conference on Learning @ Scale (L@S '22), June 1–3, 2022, New York City, NY, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3491140.3528267

1 INTRODUCTION

Personalized learning revolves around providing each student with the instruction that best suits them. At the core of personalization is the idea that there exists two groups of students, Group A and Group B, and that Group A benefits more from one teaching method, Method X, than another method, Method Y, and Group B benefits more from Method Y than Method X. This relationship, referred to as a qualitative interaction, can involve different instructional mechanisms, changing the pace of instruction, and using different evaluation methods, all with the goal of helping each student achieve their full potential. This is already difficult to achieve in small settings where teachers can directly interact with students. Effectively implementing personalized learning throughout an entire online learning platform is even more difficult.

While some platforms have created infrastructure to evaluate individual research questions regarding personalized learning [13, 18], if one instructional method is better than all other methods for all groups of students, then at least half of the students in the study received sub-optimal support, which could negatively impact the students as well as the long-term adoption of the platform. Multi-armed bandit algorithms can be used to adjust how often students receive each support option by estimating each option's effectiveness and intentionally giving more students the most effective option. Simulations using real student data found that across 22 educational experiments, using multi-armed bandit algorithms to assign students to conditions statistically significantly increased students' assignment completion rates and proficiency [16].

To automate the discovery of opportunities to personalize learning and increase students' performance within online learning platforms, this work presents the **Automatic Personalized Learning Service (APLS)**. A fully deployed service in ASSISTments, an online learning platform, that provides personalized support to struggling students upon their request. The APLS works by linking support content to aspects of students' learning environment, such

as their current assignment, problem, or prior mistakes. After compiling the relevant support content, the APLS uses a multi-armed bandit algorithm to determine which content is likely to be most helpful to the student, and then provides the selected support to the student. Two examples of student supports that can be shown to a student are shown in Figure 1.

The APLS provides student support content throughout ASSISTments whenever multiple instances of support are available, enabling multi-armed bandit algorithms to personalize students' experience at scale. However, the results of the initial APLS study using Beta-Bernoulli Thompson Sampling[21] as the recommendation algorithm were not as positive as expected. Therefore, in addition to the APLS, this work presents **Decision Tree Thompson Sampling (DTTS)**, a novel contextual multi-armed bandit algorithm designed to address the specific limitations of attempting to personalize learning within online learning platforms in an interpretable way. DTTS integrates the transparency and interpretability of decision trees into the multi-armed bandit framework and was shown in simulation to be more effective than existing multi-armed bandit algorithms when used to personalize students' support within an online learning environment.

This paper seeks to accomplish the following objectives and in doing so, provide a method for researchers to integrate personalized learning into other platforms while informing educators of the insight gained through this method:

(1) Provide a description of the APLS.
(2) Report on the results of the initial APLS study.
(3) Provide a novel algorithm (DTTS) to address the difficulties encountered when attempting to personalize students' support.
(4) Simulate the effectiveness of DTTS in realistic scenarios.

2 PRIOR WORK

There is an abundance of online platforms that attempt to personalize learning either by adjusting students' lessons based on their skill level or adapting to students' needs. McGraw-Hill Thrive, Lexia, PracTutor, HMH FUSE, Carnegie Learning's Cognitive Tutor, AutoTutor, and ASSISTments all claim to implement one or both of these methods of personalized learning [5]. However, the effectiveness of most of these platforms' personalized learning methodology has not been empirically evaluated in a classroom setting, and when evaluated, either does not show consistent positive results, as was the case with HMH FUSE [5], or achieves positive results through the use of the platform as a whole, without directly evaluating the benefit of personalizing students' experience using the platform, as was the case with ASSISTments [20] and Carnegie Learning's Cognitive Tutor [14].

Although most online learning platforms have shown little evidence for the effectiveness of personalizing students' education, there are some encouraging results as well. For example, a randomized controlled experiment done in ASSISTments found that high-knowledge students learned more from being given an entire explanation of their mistakes, while low-knowledge students learned more from being given smaller instructional segments [17]. Additionally, a meta-analysis of studies measuring the learning gains of students when grouping them by ability level found that the average effect size in 21 studies in which the students were given different instructional material was more than twice the average effect size of 30 studies in which the instructional material remained the same for each group of students [10]. The ASSISTments study and the meta-analysis both support the idea that personalizing educational content based on students knowledge level increases students' learning, but do not evaluate the effect of this personalization at scale.

While most platforms have not evaluated the overall benefits of personalization, some have allowed for testing individual hypotheses regarding personalization. The MOOClet Framework and ASSISTments have already taken steps to allow for researchers to create and deploy studies of different tutoring methods within their platforms. The MOOClet Framework provides educators with the ability to create multiple sets of material for their courses and evaluates the most effective content through randomization [18]. The ASSISTments E-TRIALS TestBed allows researchers to create modified versions of problem sets. These modified problem sets include internal random assignment of supportive content, enabling randomized controlled experimentation across all the students in ASSISTments that are assigned these problem sets [13]. In addition to allowing researchers to create randomized controlled experiments, ASSISTments also crowdsources student supports from teachers that use the platform and distributes these supports through a program called TeacherASSIST. Relevant student supports are randomly selected and provided to students upon their request, effectively creating a randomized controlled experiment in situations where more than one teacher has created support for the same problem [15]. While these platforms offer ways to gather data and test individual hypotheses, they do not automatically evaluate and deploy candidate methods for personalized learning to students throughout the platform.

3 APLS ARCHITECTURE

The APLS is designed to facilitate personalized learning in a modular way, such that regardless of the available context or student support options, the system can make an intelligent and informed decision as to which support is likely to be most beneficial to students. The APLS has two components, an online and an offline component. The online component is responsible for receiving and responding to support requests. The APLS uses the content of support requests to retrieve context it has stored on the students' learning environment, identify the potential student supports it can return, and select a multi-armed bandit model to determine which support is likely to be most effective. After a model has been chosen, potential student supports have been identified, and context on the student supports and students' environment has been gathered, the APLS uses this information to predict which student support is likely to have the most positive effect on learning. After a student support is selected, the APLS sends its prediction to the ASSISTments Tutor, which displays the support to the student.

The offline component facilitates updating the multi-armed bandit models and the context used by the models during low-load periods, e.g., at night when students that use ASSISTments are asleep. The offline component first determines the effectiveness of

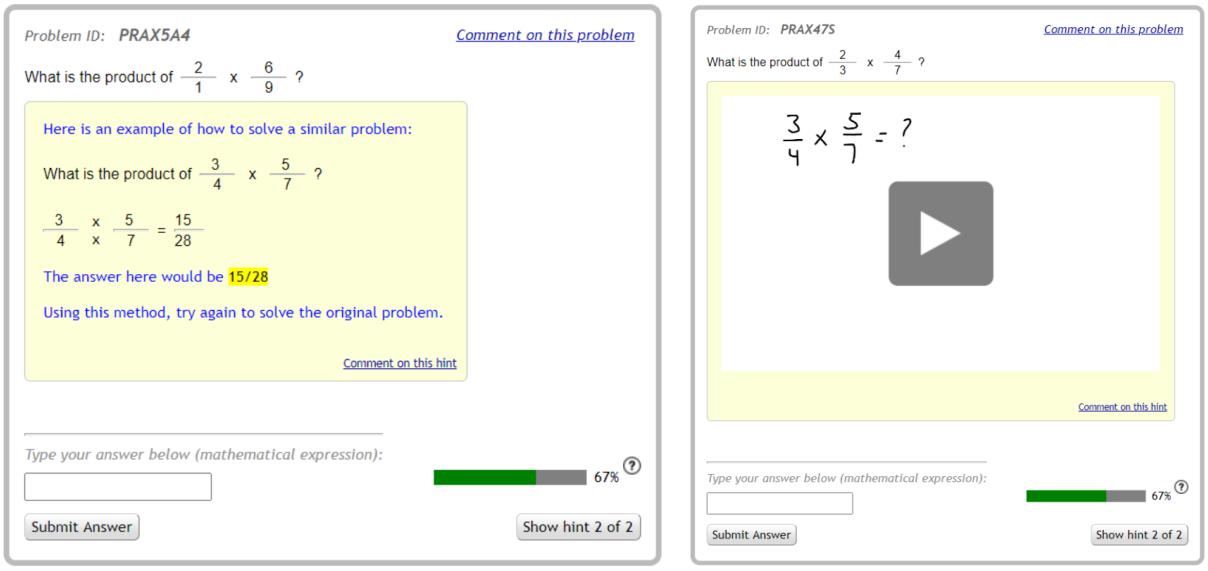


Figure 1: Two views of the ASSISTments tutor in which a student has requested support and received one of two available student supports. The student support on the left is a written explanation, and the student support on the right is a video explanation.

each recommendation made by the APLS during the day by reviewing logs of students' actions, then it uses this information to update the multi-armed bandit models. Lastly, it updates any context based on the same logs of student's actions. These offline updates allow the APLS to learn over time how to most effectively personalize students' learning. A flowchart of the online and offline tasks of the APLS is shown in figure 2.

3.1 Data Collection

In ASSISTments, as students complete their assigned problem sets, each action the student takes is recorded. This includes their use of student supports, correct and incorrect responses to problems, and duration of their engagement with various aspects of their assignments. Offline, these action logs are used to aggregate statistics on students and problems within ASSISTments, these statistics, referred to as the learning environment context, are used by the multi-armed bandit models in the APLS. The action logs are also used to determine the reward for each time a model made a recommendation, which the model uses to adjust how it makes recommendations in order to maximize reward.

Every night, the APLS collects data from the ASSISTments action logs and creates context for the recommendation models to use the following day. This context contains statistics on students' prior performance, statistics on the prior performance of all students across each problem, qualities of the problem such as text length, the skills required to solve it, its structure, e.g. multiple choice or short answer, whether it uses a diagram, and more. In addition to collecting statistics on various aspects of students' learning environments, the APLS collects information on each student support available to students. When new student supports are created, their format and HTML bodies are used to extract context such as the length of the text in the student support, whether the student support uses videos or images, whether the student support contains a question, and more. After each nightly update, the context of each student's learning environment and all student supports is updated, this ensures that the following day the models in the APLS can use the context to make recommendations. The full context of all learning environments and student supports collected by the APLS during its initial trial is hosted by the Open Science Foundation and can be found at https://osf.io/9pgv5/, and a description of all the context features used by the APLS can be found in Appendix A.

In addition to updating the learning environment and student support context, every night the reward is calculated for each recommendation made by the APLS using Algorithm 1. Algorithm 1 returns 1 when the next graded problem that the student had the opportunity to complete was completed correctly on the student's first attempt. When a student completes the next graded problem incorrectly, or when they fail to complete the next graded problem, Algorithm 1 returns 0. In cases when the student did not have an opportunity to complete a graded problem within the same assignment after requesting support, e.g., if the student requested support on the last problem in an assignment, or if all the following problems were ungraded open response problems, Algorithm 1 returns *null*. When the reward is *null*, nothing was learned about the quality of the student support, and therefore the multi-armed bandit model is not updated based on that recommendation.

3.2 Content Recommendation

In real-time, as students use the ASSISTments Tutor, they can request support. Whenever a student starts a problem, the APLS is sent a support request and uses a multi-armed bandit algorithm to return the support it predicts will be most likely to result in a high

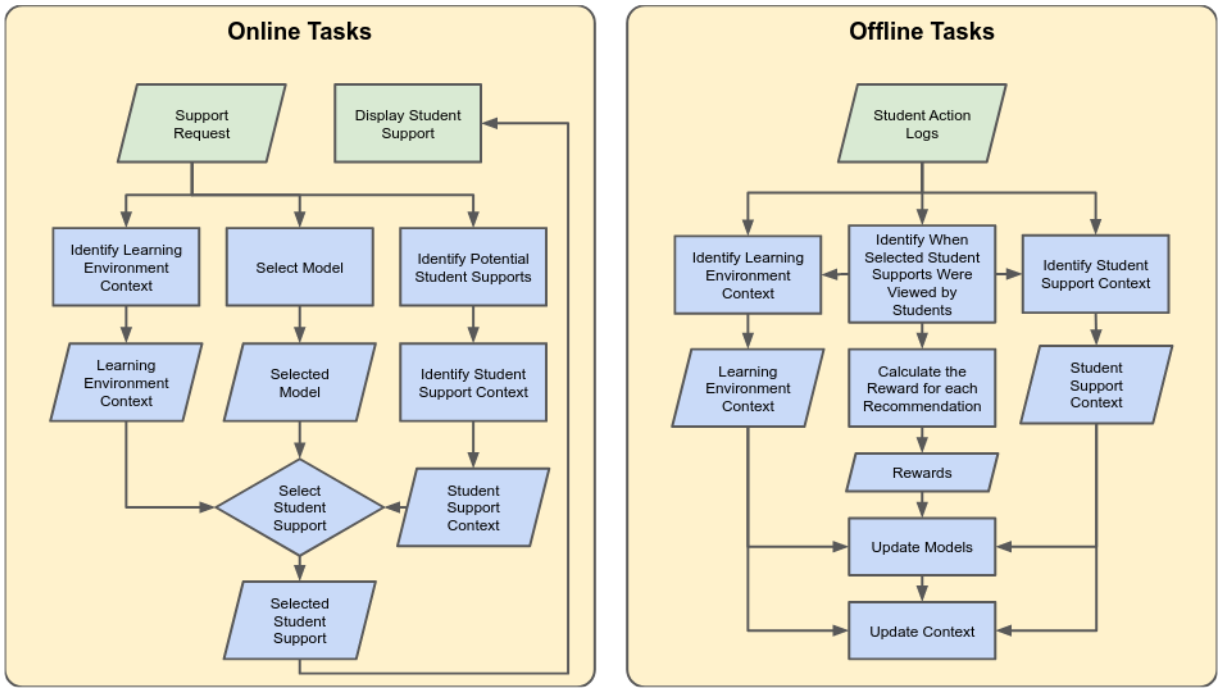


Figure 2: The online (real-time) and offline (nightly) tasks performed by the APLS. Tasks and data from the ASSISTments Tutor are in green, tasks and data from the APLS are in blue.

Algorithm 1 APLS Reward Calculation

```
a is an assignment with n problems
p_i is the ith problem in a
if student s requested tutoring for p_i then
  for m = i + 1, ..., n do
    if p_m is a graded problem then
      if s completed p_m correctly then
        return 1
      else
        return 0
      end if
    end if
  end for
else
  return null
end if
```

reward, which in this case indicates that student was able to get the next graded problem correct, and therefore likely learned from the support. If the student requests support while completing the problem, they are shown the support recommended by the APLS. The first step that the APLS performs when recommending content is to identify all of the supports that are relevant to the request. Each support request contains different IDs, e.g., a user ID, problem ID, assignment ID, and skill ID. These IDs are linked to student supports and determine which student supports are available for recommendation. For example, the most common link is between a student support and a problem ID. These links imply that the student support was written specifically for the problem. A less common link is between a student support and a skill ID, which implies that the student support was written for any problem relevant to a particular skill. This modular linking system allows for educators and researchers to create content related to any aspect of students' learning environment by simply linking the student support to the relevant ID, enabling these student supports to be assessed for their effectiveness and used to further personalize students' learning.

After the potential student supports have been identified, context for the multi-armed bandit algorithm is gathered from the IDs of the request and the potential student supports. Just as each ID in the request can link to a student support, the IDs of the request and the potential student supports can link to an array of features. For example, each problem ID links to an array of performance statistics and HTML-based attributes of that particular problem, and each student support ID links to an array of features on the structure and content of the support. All of the relevant feature arrays are concatenated and become the context used by the multi-armed bandit algorithm. It is useful to note that the context of each potential student support will only differ in the values related to the specific student support, the context related to any ID of the support request will be constant across all potential student supports, as the IDs in the request are not changing between the potential student supports of a single support request.

Once the context of each potential student support is gathered, The APLS selects a multi-armed bandit model using a similar process to the one used to identify potential student supports. First, potential models are selected by identifying models linked to IDs

in the support request, then, one of the potential models is selected at random. Randomly selecting a model facilitates randomized controlled experiments that compare the effectiveness of different models. The first of these experiments is reported on in section 4. Once a model is selected, the context derived from the support request and potential student supports is provided to the model and the model selects the student support it predicts will lead to the highest reward and therefore be most likely to help the student learn. The selected student report is returned to the ASSISTments Tutor and displayed if requested by the student.

3.3 Model Updates

The traditional way to update a multi-armed bandit model is to provide the model with a reward after each recommendation it makes. The model then uses this reward to adjust its internal logic such that it will be more likely to receive higher rewards after future recommendations. However, the models in the APLS can only observe a reward once a student has finished the problem they requested support on, and the order that students complete problems is not necessarily the order that recommendations were made in. Therefore, instead of attempting to update the models in real-time, the models are updated in batches during lulls in user activity.

The APLS tracks which model made each recommendation and the context used to make the recommendation. Each night, after identifying the reward for each recommendation using Algorithm 1, the APLS provides each context-reward pair to the model used to make the recommendation in the order that the recommendations were made. During this process any student support requests sent to the APLS are processed using the models in their states prior to the current update to ensure that there is no downtime or ambiguity in the APLS recommendations. After every model has been updated, the APLS begins to use the updated models to make recommendations.

4 APLS EXPERIMENT

4.1 Experiment Design

To determine the effectiveness of using multi-armed bandit algorithms at scale, a randomized controlled experiment was performed within ASSISTments between November 4th, 2021 and January 2nd, 2022. In this experiment, the APLS randomly selected which of two models it would use to recommend content to students. Students were randomized between the two models each time they started a problem. Therefore, the same student could have received content from both models, but would be unable to determine which model was recommending them content because both models were selecting from the same potential student supports.

The first model used random selection (RS) to select which student support to recommend, and the second model used Beta-Bernoulli Thompson Sampling (BBTS) to select which student support to recommend. BBTS is a simple contextual bandit algorithm for environments with binary rewards. It models the potential reward of each student support as a beta distribution $Beta(\alpha, \beta)$ where α is the number of times the student support was recommended and the model received a reward of 1, and β is the number of times the student support was recommended and the model received a reward of 0. When determining which student support to recommend, a random value is drawn from the beta distribution of each possible student support. The student support corresponding to the highest random value is recommended to the student. BBTS is not aware of the context in which it made recommendations, and learns simply from the rewards that it received in the past [21]. Although BBTS is simple compared to a contextual bandit algorithm, it is a strong baseline from which insight can be gained and advances can be made [6].

Table 1: Results of Thompson Sampling vs. Random Selection

	RS	BBTS
Total Requests	49,740	50,379
Requests with Rewards	40,878 (82.2%)	41,306 (82.0%)
Total Reward	13,529 (33.1%)	13,805 (33.4%)

4.2 Experiment Results

Over the two months that the experiment ran, support was requested by students on about 16.5% of problems. 49,740 support requests were responded to by the APLS using RS, and 50,379 requests were responded to using BBTS. Of all these requests, 82.2% of the RS requests and 82.0% of the BBTS requests did not have a reward due to students receiving support prior to ungraded open ended response questions or on the last problem in their assignment. The average reward received by each model was 0.331 and 0.334 respectively. Table 1 summarizes these findings.

Although BBTS out performed RS, its impact was less than expected based on previous simulations [16]. To understand why BBTS was less effective than expected, it helps to examine the challenges associated with making recommendations within an online learning platform. Firstly, consider the breadth of questions for which supports are being recommended. The BBTS model recommended tutoring for 2,923 different problems, and only recommended each student support an average of 6.5 times. This is very little information to learn from, and thus limited the model's ability to significantly out perform RS. Secondly, consider the way in which students interact with online learning platforms. It is common for a teacher to assign a particular problem set to be completed within the day or week. If an entire class attempts this problem set, and only 16.5% of students request support, half of those supports are going to the RS model, and only four out of five of the support requests can be learned from, than only around 6.5% of students will affect the BBTS model. Furthermore, these students are not coming in a steady stream, but rather in batches of classes. The student supports that work best for one class are not necessary going to be the best for every class. Teachers use different methods of explaining content and the supports that align with the teachers' instruction are more likely to be effective. Therefore, without context of the students' learning environment, the BBTS model could learn relationships from one class that are detrimental to the following class. While the online learning environment made it difficult for BBTS to perform well, a contextual bandit algorithm that can utilize context of the learning environment and student supports to

share insight between student supports can theoretically overcome the shortcomings of BBTS.

5 MOVING FORWARD: DTTS

To overcome the challenges of making recommendations within an online learning platform and to be able to identify opportunities for personalization that can be applied outside of the contextual bandit framework, a recommendation algorithm must be able to accomplish the following:

(1) Apply insight gained from recommending a student support to recommendations of other supports.
(2) Identify generally applicable qualitative interactions between students' learning environment and the effectiveness of different supports.
(3) Learn interpretable insights that can be easily extracted from the model and understood by educators and researchers.

To be able to apply insight gained from recommending a student support to recommendations of other supports, the model must take into account the context of each potential student support when selecting which support to recommend. Models like LinUCB, which use a separate regression for each potential student support would not be able to inform educators of the common trends between student supports without doing additional analysis to combine each regressions' findings [11].

To be able to identify generally applicable qualitative interactions between students' learning environment and the effectiveness of different supports, not only must the model combine context of the learning environment and supports, but it must do so in a non-linear way. Models like Hybrid-LinUCB [11] and Linear Thompson Sampling [1], both popular models that combine context of the learning environment and supports, only form linear relationships between these parameters. A qualitative interaction is a non-linear relationship. To make these models capable of identifying qualitative interactions, one must manually add all the potential interaction terms, and doing so would create thousands of additional variables, making these models slow and prone to over-fitting.

To be able to learn interpretable insights that can be easily extracted from the model and understood by educators and researchers, the model must not abstract the context it is given too severely. There are many neural network based contextual bandit algorithms [19], but the insight gained by these algorithms is represented by weights in neural networks, which are notoriously difficult to interpret as the size of the network increases. Furthermore, difficult to interpret models can lead to skepticism and lack of adoption due to the rising increase in demand for interpretable AI, therefore educators are more likely to accept recommendations from a model that can explain its reasoning [12].

These factors point towards using decision trees to find opportunities for personalization because decision trees are capable of combining the context of students' learning environments and the potential student supports in a non-linear and easily interpretable way. While decision trees are a strong contender for delivering interpretable personalization to students, adapting them to the contextual bandit framework, where they must balance exploration and exploitation to learn how to determine the optimal support for each student, is non-trivial. In the past, researchers attempted to combine decision trees and reinforcement learning by fitting a decision tree each time a recommendation was made on a random subset of the data, which simulated the effect of sampling from a prior distribution [8]. This method has the limitation of needing to retrain the tree to ensure exploration, which is very time-consuming. Another method that attempted to integrate decision trees into the multi-armed bandit framework first explored all possible branches the tree could form, and then selected the branch that led to the highest reward [9]. This method has the limitation of being unable to adapt over time to changing interactions within the context, and requires the exploration of all possible branches, which becomes overwhelmingly time-consuming when there are many possible branches.

Decision Tree Thompson Sampling (DTTS) is a novel decision tree based multi-armed bandit algorithm designed to interpretably recommend support to students within online learning platforms. DTTS integrates decision trees into the multi-armed bandit framework by using the decision tree not as as explicit predictor of the expected reward, but as a model of the prior reward distribution, which can be sampled from to make recommendations via Thompson sampling. As shown in Algorithm 2, given an uninitialized decision tree DT, an empty set of all context observations X, and an empty set of all reward observations R, every n observations, DT is trained on up to m past observations to predict the expected reward given the context. At each time-step t, for all time-steps in T, the leaf node of DT, $l_{x_{t,a}}$, that corresponds to the observed context $x_{t,a}$ is identified for each available action a. In this case, each action is a potential student support. Then, prior distributions are calculated from $R_{l_{x_{t,a}}}$, the subset of R used to create the tree that reached the same leaf node l for each action a out of all K available actions. The prior distribution calculated from $R_{l_{x_{t,a}}}$ is a beta distribution $Beta(\alpha, \beta)$ where $alpha$ is the total number of times the reward was 1 in $R_{l_{x_{t,a}}}$, and β is the number of times the reward was 0 in $R_{l_{x_{t,a}}}$. If the decision tree has not been created yet, then the prior distribution is assumed to be a uniform distribution in the range $[0, 1)$. After determining the prior distributions for each potential action, like in Thompson sampling, each prior distribution is randomly sampled from, and the action corresponding to the random sample $\tilde{\theta}_{t,a}$ with the highest value is chosen. After the action is taken and a reward is observed, the context $x_{t,a}$ and reward r_t are added to X and R respectively. If the length of X and R is greater than m, the oldest observation is removed from X and R. Only storing the most recent m observations helps to keep DT up to date with the current trends in the observations.

6 DTTS SIMULATION

6.1 Simulation Data

The data used to evaluate DTTS came from the ASSISTments TeacherASSIST program. Within ASSISTments, a crowdsourcing effort called TeacherASSIST allowed teachers to create their own student supports for mathematics problems, which were then distributed to all students using ASSISTments. In 2019, a randomized controlled experiment was carried out within ASSISTments in which students were randomized between receiving different

Algorithm 2 Decision Tree Thompson Sampling

DT is an uninitialized CART decision tree
n is an integer. *DT* is trained every *n* observations
X is a set of length *m* that will hold observed contexts
R is a set of length *m* that will hold observed rewards
for t = 1, ..., T **do**
  for a = 1, ..., K **do**
    observe context $x_{t,a}$
    if *DT* is uninitialized **then**
      sample $\tilde{\theta}_{t,a}$ from $[0, 1)$
    else
      sample $\tilde{\theta}_{t,a}$ from $P(R_{l_{x_{t,a}}})$
    end if
  end for
  choose action $a_t = \underset{a}{argmax}\ \tilde{\theta}_{t,a}$
  observe reward r_t
  add $x_{t,a}$ and r_t to *X* and *R* respectively
  if Length of $X > m$ **then**
    remove the oldest observation from both *X* and *R*
  end if
  if $t \geq n$ and $t \pmod n = 0$ **then**
    train *DT* to predict *R* using *X*
  end if
end for

crowdsourced supports. This study found that providing students with crowdsourced supports led to greater learning gains than when students were only given the answer after requesting support [15]. In 2020, the 2019 study was repeated and the same results were found. The data collected from the TeacherASSIST program contains 399,869 instances of a struggling student requesting support, being given one of multiple possible crowdsourced student supports, and then having the opportunity to answer a graded problem within the same assignment. 1,946 teachers, 5,635 classes, 27,712 assignments, 62,056 students, 5,470 mathematics problems, and 13,394 different student supports are contained within this data. For each teacher, class, assignment, user, problem, and next problem, prior performance statistics are available for every support request. Various features of the problem sets, problems, and student supports are also available. Additionally, in order to make the insight gained from these features more interpretable, for this simulation, each continuous feature was converted into a binary indicator of whether the value was above average or not. The full dataset and a description of all of its contents is hosted by the Open Science Foundation at https://osf.io/9pgv5/. In total, 98 of the available features were used in this analysis, specifically the features focused on prior statistics and aspects of the students, problems, and student supports. This data is ideal for determining the effectiveness of different recommendation algorithms within online learning platforms because it contains real information on the learning gains of thousands of students after receiving one of thousands of supports, and the features are nearly identical to the context used by the APLS described in Appendix A.

6.2 Simulation Design

The data from TeacherASSIST can be sampled from to simulate a mutli-armed bandit algorithm operating within an online learning platform. This can be achieved using the following simulation strategy:

(1) Initialize a multi-armed bandit algorithm that may take features of the student, problem, and potential student supports as context and uses the reward metric described by Algorithm 1.
(2) Randomly sample with replacement a single instance of a student receiving support from all the TeacherASSIST data.
(3) Use the bandit algorithm to recommend which tutoring, from all the possible student supports, the student in the random sample should receive.
(4) If the recommended support was the support that the student actually received, then update the bandit algorithm using the calculated reward, otherwise ignore this sample and go back to step 2.
(5) Repeat steps 2-4 as many times as desired to simulate the contextual bandit algorithm making multiple sequential recommendations.

6.3 Learning Impact

The process described in Section 6.2 can be used to answer the question "How would students' learning gains have been different if a multi-armed bandit algorithm besides BBTS was used to recommend student supports?". To evaluate the effectiveness of DTTS, three different simulation studies were run in which DTTS was compared to RS, BBTS [21], and Linear Thompson Sampling (LTS) [1]. In all three simulations, DTTS used a CART decision tree [4] that used Gini Impurity as its split criteria, and the ϵ and δ parameters of LTS were set using Theorem 1 and Remark 2 of [1] to obtain a 95% chance of having an $\tilde{O}(d^2\sqrt{T})$ regret bound. The first simulation used the process described in Section 6.2 to sample from all the TeacherASSIST data 376,674 times, which is how many recommendations with reward were collected by TeacherASSIST in the year 2020. This gives insight into how DTTS would have performed compared to random selection and popular multi-armed bandit algorithms over the course of a full year.

The second and third simulations investigated how capable DTTS is of generalizing its insight to new content. The second simulation is similar to the first, but it divided the data into groups, where each group contained unique students. Each group was sampled from approximately 1,721 times, which is the average number of recommendations with reward made in a single day, before moving on to the next group. This simulation helped evaluate DTTS's ability to learn contextual insight beneficial to new students because each simulated day, the students were unique. Therefore DTTS would have to gain generalizable insight from students' context to be able to make helpful recommendations. The third simulation was similar to the second except it divided the data into groups of unique problems and student supports. This simulation helped evaluate DTTS's ability to learn generalizable insight applicable to new problems and supports. The cumulative reward over 376,674 observations and the percent of recommendations with a reward of 1 for every model in each simulation is shown in Table 2.

Table 2: Cumulative Rewards and % Maximum Reward for all Simulations

Simulation Description	RS	BBTS	LTS	DTTS
All Data	127,302 (33.8%)	133,601 (35.5%)	126,449 (33.6%)	**146,983 (39.0%)**
Student Groups	127,024 (33.7%)	**134,615 (35.7%)**	127,080 (33.7%)	134,073 (35.6%)
Problem Groups	127,000 (33.7%)	134,028 (35.6%)	127,274 (33.8%)	**134,714 (35.8%)**

Table 2 shows that, overall, DTTS not only outperformed BBTS, which is currently used in the APLS, but also outperformed LTS, a well established contextual bandit model. In the student groups simulation, BBTS sightly out performed DTTS. This implies that while DTTS is capable of generalizing findings between students, these findings do not provide more insight than modeling the reward of each support independently. In the problem groups simulation, DTTS was again able to out-perform all other models, which implies that DTTS can learn relationships that generalize across problems and tutoring. Across all three simulations, DTTS demonstrated that the contextual insight it learned was applicable both to new students, and new problems and student supports. Overall, it seems that interactions between features of the problems and student supports are more generalizable than interactions with features of the students.

The difficulty of using multi-armed bandit algorithms in an online learning environment is made clear by LTS's inability to outperform BBTS. Although this simulation was focused on evaluating the effectiveness of DTTS, this is a valuable finding because it implies that the quality of the student support is dependent on the learning environment. LTS can only model how much each feature of student supports influences the likelihood of the student getting the next graded problem correct. Without interactions between context of the student supports and context of the learning environment, it struggles to gain insight. For example, LTS can only learn that longer student supports have an overall positive or overall negative effect on students' learning, but BBTS can learn that for one particular problem, a longer support is better, and for a different problem, a shorter support is better, even though it does not know the length of the student supports. Without being able to generalize this finding across multiple supports, BBTS still out-performs LTS. This finding implies that DTTS outperformed BBTS and LTS because it could both utilize non-linear relationships, and generalize these relationships to previously unseen students, problems, and student supports.

6.4 Personalization Insight

Although we have shown that DTTS is more capable of increasing students' learning gains than other multi-armed bandit algorithms, there is no guarantee that it does so by identifying and taking advantage of qualitative interactions between students' learning environment and the potential student supports. To investigate if this is the case, the simulation strategy described in Section 6.2 was again used to simulate a year of DTTS based recommendations, but the simulation only used a random half of the TeacherASSIST data. After the simulation finished, the total amount that each feature of the context contributed to the reduction of Gini Impurity in the final decision tree formed by DTTS was compared to the strength of the qualitative interactions that existed in the other half of the TeacherASSIST data not used in the simulation. To make this comparison, the first step was to use Equation 1 to fit a model for each of the 2,001 possible qualitative interactions between the student supports and the learning environment, where x_1 is a feature of the student supports, x_2 is feature of the learning environment, and y is the reward calculated using Algorithm 1.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 (x_1 \oplus x_2) \tag{1}$$

For there to be a qualitative interaction, the effect of x_1 when $x_2 = 0$, $\text{eff}_{x_1|x_2=0}$, must have the opposite sign as the effect of x_1 when $x_2 = 1$, $\text{eff}_{x_1|x_2=1}$, or in other words, the product of the two effects must be less than zero. This can be determined by whether or not β_3^2 is greater than β_1^2, using the following logic.

$$E[y|x_1 = 0, x_2 = 0] = \beta_0$$
$$E[y|x_1 = 1, x_2 = 0] = \beta_0 + \beta_1 + \beta_3$$
$$E[y|x_1 = 0, x_2 = 1] = \beta_0 + \beta_2 + \beta_3$$
$$E[y|x_1 = 1, x_2 = 1] = \beta_0 + \beta_1 + \beta_2$$

$$\text{eff}_{x_1|x_2=0} = E[y|x_1 = 1, x_2 = 0] - E[y|x_1 = 0, x_2 = 0]$$
$$\text{eff}_{x_1|x_2=0} = (\beta_0 + \beta_1 + \beta_3) - (\beta_0) = \beta_1 + \beta_3$$
$$\text{eff}_{x_1|x_2=1} = E[y|x_1 = 1, x_2 = 1] - E[y|x_1 = 0, x_2 = 1]$$
$$\text{eff}_{x_1|x_2=1} = (\beta_0 + \beta_1 + \beta_2) - (\beta_0 + \beta_2 + \beta_3) = \beta_1 - \beta_3$$

$$(\beta_1 + \beta_3)(\beta_1 - \beta_3) < 0$$
$$\beta_1^2 - \beta_3^2 < 0$$
$$\beta_1^2 < \beta_3^2$$

Once it has been determined whether or not there is a qualitative interaction between two features, the absolute value of the t-value of β_3 can be used to measure the strength of the qualitative interaction in a way that takes into account the magnitude and the standard error of the β_3 coefficient.

After the presence and strength of each potential qualitative interaction was determined, the strength was compared to the product of the total reduction of Gini Impurity that each of the features in the interaction is responsible for in the decision tree formed by DTTS. If the magnitude of the t-value is correlated with the product of the total reductions in Gini Impurity, this implies that stronger qualitative interactions are used more by DTTS. This helped determine whether the insight learned by the decision tree involved personalization, and if that insight was applicable to new

Table 3: Correlation Between Qualitative and Non-Qualitative Interactions and the Product of the Total Reduction in Gini Impurity of Each Feature in the Interaction Determined Using the Decision Tree Formed by DTTS

	Total #	*rho*	*p*-value
Qualitative Interactions	823	0.11	0.002
Non-Qualitative Interactions	1178	0.06	0.052

data or just over-fit assumptions based on the data used to create the decision tree.

Using the above methodology, the correlation between the magnitude of each potential qualitative interaction's t-value and the product of the total reductions in Gini Impurity of each feature of the interaction in the DTTS decision tree was determined using Spearman's rank correlation coefficient [22]. The results of this, shown in Table 3 demonstrate that when there is a qualitative interaction between two features, the decision tree utilizes the features more as the strength of the qualitative interaction increases, and when there is no qualitative interaction, there is no correlation with how valuable the features are to the decision tree. This implies that the decision tree formed by DTTS is taking advantage of qualitative interactions to personalize students' learning. Due to the interpretability of decision trees, it is possible to search the tree, identify these interactions, and use them to inform the design of curricula with the intent of personalizing students' education.

7 LIMITATIONS AND FUTURE WORK

The APLS, while a promising tool for personalizing students' online education, has yet to reach its full potential. While the process of recommending content is modular, it relies on having many optional supports available. Personalized learning relies on having enough content such that each student can be delivered support suitable for their needs. In the future, collecting more student support messages, either through crowdsourcing or algorithmic generation, will be key in ensuring that any online learning platform can provide quality personalization. Additionally, the ability to collect relevant context on the student supports and learning environment is imperative to the success of personalized learning at scale. The APLS currently collects context on the prior statistics, format, and HTML-based attributes of the students, problems, and student supports. This can be expanded, not only to include more features from within the online learning platform, such as teachers' behavioral patterns, but also to include features such as the sentiment or tone of the student supports or the emotional state of the students. These features, which can be collected with a variety of different algorithms, e.g. [2, 3], would help the APLS make more informed and insightful recommendations.

The APLS has only been tested using BBTS, a simple multi-armed bandit algorithm to recommend support to students. However, DTTS was able to overcome the difficulties of recommending content within an online learning platform and outperformed both BBTS and LTS in simulation. While the future of DTTS is bright, more work needs to be done to confirm the findings of this simulation. The next step is to integrate DTTS into the APLS and empirically measure its impact on learning at scale. Additionally, while DTTS was the most effective algorithm, it was also the only algorithm that needed to record previous observations. Both BBTS and LTS need to only store statistics on the previous observations, making them much more memory efficient. Moving forward, experimenting with using Hoeffding Trees instead of CART Decision Trees could reduce the memory cost. Hoeffding Trees are designed to use each observation as input only once and store statistics on each observation instead of a history of observations [7]. Beyond exploring improvements to DTTS, the purpose of using decision trees was to ensure interpretability. Therefore, when DTTS is integrated into online learning platforms, work should be done to make the recommendations made by DTTS as transparent as possible. Creating a user interface to help educators and researchers understand what qualitative interactions DTTS is taking advantage of when recommending content would both facilitate adoption of DTTS and inform personalized learning pedagogy.

8 CONCLUSION

This work presented the Automatic Personalized Learning Service (APLS), a novel infrastructure for personalized learning within ASSISTments, an online learning platform with around 300,000 active users. An empirical study was performed and it was demonstrated that using the APLS to recommend support for struggling students with Beta-Bernoulli Thompson Sampling (BBTS), a common and effective multi-armed bandit algorithm, was only slightly better that selecting from the optional relevant student supports at random. Further investigation revealed that the lack of significant improvement was due to the breadth of problems for which support needed to be recommended and the sparsity in opportunity to recommend the same content multiple times. These shortcomings prompted the creation of Decision Tree Thompson Sampling (DTTS), a novel multi-armed bandit algorithm for recommending content that combines the interpretability and non-linearity of decision trees with Thompson sampling's proven approach to the exploitation-exploitation trade-off. DTTS was shown in simulation to outperform both BBTS and Linear Thompson Sampling (LTS), demonstrating its ability to learn generalizable insights into how to effectively personalize learning for students, problems, and student supports that it had no prior exposure to. Additionally, using DTTS to simulate recommending student supports, then correlating the importance of each feature, as determined by the decision tree made by DTTS, to the magnitude of every potential qualitative interaction in a separate dataset found that the importance of the features correlated with the strength of their qualitative interaction. Implying that the insight gained by DTTS is applicable to new content and relies on the identification of qualitative interactions, which are essential for personalization. Moving forward, DTTS will be integrated into the APLS in ASSISTments where it can begin to personalize learning for thousands of students across the world and other platforms can begin to integrate the APLS framework using DTTS.

ACKNOWLEDGMENTS

We would like to thank NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889,

1636782, 1535428), IES (e.g., R305N210049, R305D210031, R305A170-137, R305A170243, R305A180401, R305A120125), GAANN (e.g., P200A180088 P200A150306), EIR (U411B190024), ONR (N00014-18-1-2768) and Schmidt Futures.

REFERENCES

[1] Shipra Agrawal and Navin Goyal. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*. PMLR, 127–135.
[2] María Lucía Barrón-Estrada, Ramon Zatarain-Cabada, Raúl Oramas-Bustillos, and Francisco González-Hernández. 2017. Sentiment analysis in an affective intelligent tutoring system. In *2017 IEEE 17th international conference on advanced learning technologies (ICALT)*. IEEE, 394–397.
[3] Anthony F Botelho, Ryan S Baker, and Neil T Heffernan. 2017. Improving sensor-free affect detection using deep learning. In *International conference on artificial intelligence in education*. Springer, 40–51.
[4] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. 2017. *Classification and regression trees*. Routledge.
[5] Monica Bulger. 2016. Personalized learning: The conversations we're not having. *Data and Society* 22, 1 (2016), 1–29.
[6] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. *Advances in neural information processing systems* 24 (2011), 2249–2257.
[7] Pedro Domingos and Geoff Hulten. 2000. Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. 71–80.
[8] Adam N Elmachtoub, Ryan McNellis, Sechan Oh, and Marek Petrik. 2017. A practical method for solving contextual bandit problems using decision trees. *arXiv preprint arXiv:1706.04687* (2017).
[9] Raphaël Féraud, Robin Allesiardo, Tanguy Urvoy, and Fabrice Clérot. 2016. Random forest for the contextual bandit problem. In *Artificial intelligence and statistics*. PMLR, 93–101.
[10] Chen-Lin C Kulik and James A Kulik. 1982. Effects of ability grouping on secondary school students: A meta-analysis of evaluation findings. *American educational research journal* 19, 3 (1982), 415–428.
[11] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.
[12] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2021. Explainable ai: A review of machine learning interpretability methods. *Entropy* 23, 1 (2021), 18.
[13] Korinn Ostrow, Neil Heffernan, and Joseph Williams. 2017. Tomorrow's EdTech today: Establishing a learning platform as a collaborative research tool for sound science. *Teachers College Record* 119, 3 (2017), 1–36.
[14] John F Pane, Beth Ann Griffin, Daniel F McCaffrey, and Rita Karam. 2014. Effectiveness of cognitive tutor algebra I at scale. *Educational Evaluation and Policy Analysis* 36, 2 (2014), 127–144.
[15] Thanaporn Patikorn and Neil T Heffernan. 2020. Effectiveness of crowd-sourcing on-demand assistance from teachers in online learning platforms. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*. 115–124.
[16] Anna Rafferty, Huiji Ying, Joseph Williams, et al. 2019. Statistical consequences of using multi-armed bandits to conduct adaptive educational experiments. *Journal of Educational Data Mining* 11, 1 (2019), 47–79.
[17] Leena M Razzaq and Neil T Heffernan. 2009. To Tutor or Not to Tutor: That is the Question.. In *AIED*. 457–464.
[18] Mohi Reza, Juho Kim, Ananya Bhattacharjee, Anna N Rafferty, and Joseph Jay Williams. 2021. The MOOClet Framework: Unifying Experimentation, Dynamic Improvement, and Personalization in Online Courses. In *Proceedings of the Eighth ACM Conference on Learning@ Scale*. 15–26.
[19] Carlos Riquelme, George Tucker, and Jasper Snoek. 2018. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127* (2018).
[20] Jeremy Roschelle, Mingyu Feng, Robert F Murphy, and Craig A Mason. 2016. Online mathematics homework increases student achievement. *AERA open* 2, 4 (2016), 2332858416673968.
[21] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. 2017. A tutorial on thompson sampling. *arXiv preprint arXiv:1707.02038* (2017).
[22] Charles Spearman. 1961. The proof and measurement of association between two things. (1961).

A APLS FEATURE DESCRIPTIONS

For users, summary statistics based on all prior problems completed by the user are calculated. These same summary statistics are calculated for problems based on all prior instances of a student attempting the problem. These summary statistics, listed below, make up 20 of the context features.

- **total_assignments_completed**: For users, the total number of assignments completed previously. For problems, The total number of assignments with the problem in it completed previously.
- **total_problems_completed**: For users, the total number of problems completed previously. For problems, The total number of times the problem was completed previously.
- **assignment_completion_percentage**: For users, the percent of previously started assignments that were completed. For problems, the percent of previously started assignments with the problem in it that were completed.
- **problem_completion_percentage**: For users, the percent of previously started problems that were completed. For problems, the percent of previous times the problem was started that is was also completed.
- **median_time_on_task**: For users, the median time on task spent on problems. For problems, the median time on task spent by students on the problem.
- **median_first_response_time**: For users, the median time spent before submitting an answer or requesting tutoring when completing a problem. For problems, the median time spent before submitting an answer or requesting tutoring for students completing the problem.
- **average_correctness**: For users, the fraction of times they got previously completed problems correct on their first attempt. For problems, the fraction of times that students attempting the problem got it correct on their first attempt.
- **average_attempt_count**: For users, the average number of attempts per problem on previously completed problems. For problems, the average number of attempts for students that completed the problem.
- **average_hint_count**: For users, the average number of hints used per previously attempted problem. For problems, the average number of hints used by students that previously completed the problem.
- **average_first_action_answer**: For users, the fraction of times that the student attempted to answer the problem before requesting tutoring. For problems, the fraction of times that students completing the problem attempted to answer the problem before requesting tutoring.

One-hot encoded categorical variables for problem answer type, grade level, and subject are also included in the context of the APLS. Descriptions of the 10 answer types are listed below.

- **problem_type_1**: Multiple Choice
- **problem_type_2**: Check All That Apply
- **problem_type_3**: Place Items In Order
- **problem_type_4**: Exact Match (case sensitive)
- **problem_type_5**: Legacy Algebraic Expression, e.g., z = 2y
- **problem_type_11**: Exact Match (ignore case)
- **problem_type_13**: Number, e.g., 93
- **problem_type_14**: Numeric Expression, e.g., 3 + 2 * 4
- **problem_type_15**: Exact Fraction, e.g., 3/2
- **problem_type_17**: Algebraic Expression, e.g., z = 3x + 2y

The 15 grade level features and 32 subject features are described by the Common Core State Standards for Mathematics, which can be found at http://www.corestandards.org/Math/. The first and second section of the Common Core Skill Code correspond to the grade level and subject respectively. For example, for the skill code **7.RP.A.2.d**, the grade level is **7** and the subject is **RP**.

Lastly, problems and student supports have context used by the APLS corresponding to their HTML structure. Student supports have more structural features than problems. Therefore, in the following list, every feature exists for student supports and starred features also exist for problems.

- **student_support_content_creator_id**: The ID of the creator of the student support.
- **student_support_hint**: A binary indicator of whether or not the student support is a hint.
- **student_support_explanation**: A binary indicator of whether or not the student support is an explanation.
- **student_support_message_count**: The number of messages contained within the student support.
- ***_text_length**: The character count of all the text in the problem or student support.
- ***_contains_video**: A binary indicator of whether or not the problem or student support contains a video.
- ***_contains_image**: A binary indicator of whether or not the problem or student support contains an image.
- ***_contains_link**: A binary indicator of whether or not the problem or student support contains a link.
- ***_color_use**: A binary indicator of whether or not the problem or student support uses different text colors.
- ***_font_use**: A binary indicator of whether or not the problem or student support uses different text fonts.
- ***_text_size_use**: A binary indicator of whether or not the problem or student support uses different text sizes.

Chapter 3.2

Investigating the Impact of Skill-Related Videos on Online Learning

Investigating the Impact of Skill-Related Videos on Online Learning

Ethan Prihar
Worcester Polytechnic Institute
Worcester, Massachusetts, USA
ebprihar@wpi.edu

Aaron Haim
Worcester Polytechnic Institute
Worcester, Massachusetts, USA
ahaim@wpi.edu

Tracy Shen
The Pennsylvania State University
University Park, Pennsylvania, USA
jqs5443@psu.edu

Adam Sales
Worcester Polytechnic Institute
Worcester, Massachusetts, USA
asales@wpi.edu

Dongwon Lee
The Pennsylvania State University
University Park, Pennsylvania, USA
dongwon@psu.edu

Xintao Wu
University of Arkansas
Fayetteville, Arkansas, USA
xintaowu@uark.edu

Neil Heffernan
Worcester Polytechnic Institute
Worcester, Massachusetts, USA
nth@wpi.edu

ABSTRACT

Many online learning platforms and MOOCs incorporate some amount of video-based content into their platform, but there are few randomized controlled experiments that evaluate the effectiveness of the different methods of video integration. Given the large amount of publicly available educational videos, an investigation into this content's impact on students could help lead to more effective and accessible video integration within learning platforms. In this work, a new feature was added into an existing online learning platform that allowed students to request skill-related videos while completing their online middle-school mathematics assignments. A total of 18,535 students participated in two large-scale randomized controlled experiments related to providing students with publicly available educational videos. The first experiment investigated the effect of providing students with the opportunity to request these videos, and the second experiment investigated the effect of using a multi-armed bandit algorithm to recommend relevant videos. Additionally, this work investigated which features of the videos were significantly predictive of students' performance and which features could be used to personalize students' learning. Ultimately, students were mostly disinterested in the skill-related videos, preferring instead to use the platforms existing problem-specific support, and there was no statistically significant findings in either experiment. Additionally, while no video features were significantly predictive of students' performance, two video features had significant qualitative interactions with students' prior knowledge, which showed that different content creators were more effective for different groups of students. These findings can be used to inform the design of future video-based features within online learning platforms and the creation of different educational videos specifically targeting higher or lower knowledge students. The data and code used in this work is hosted by the Open Science Foundation and can be found at https://osf.io/cxkzf/.

L@S '23, July 20–22, 2023, Copenhagen, Denmark.

ACM ISBN -.
https://doi.org/-

CCS CONCEPTS

• **Applied computing** → **Education**; **Distance learning**; **Computer-assisted instruction**.

KEYWORDS

Video Tutoring, Randomized Controlled Experiments, Multi-Armed Bandit Algorithms, Personalized Learning

ACM Reference Format:
Ethan Prihar, Aaron Haim, Tracy Shen, Adam Sales, Dongwon Lee, Xintao Wu, and Neil Heffernan. 2023. Investigating the Impact of Skill-Related Videos on Online Learning. In *Proceedings of the Tenth ACM Conference on Learning @ Scale (L@S '23), July 20–22, 2023, Copenhagen, Denmark.* ACM, New York, NY, USA, 10 pages. https://doi.org/-

1 INTRODUCTION

There is currently a plethora of educational content available for free online. While this can empower students savvy enough to navigate to relevant content on their own, searching for relevant content can frustrate less experienced students, increasing their cognitive load and making it more difficult for them to obtain the same benefits [8]. Often, learning platforms will develop their own instructional content by working with, or crowdsourcing from experts, e.g., [3, 13], but this can be time consuming and expensive. In many cases, teachers will search for hours to find relevant instructional content to distribute to their students [11]. We are interested in reducing the cost for learning platforms to provide relevant instructional content to students and taking the burden of identifying and distributing relevant instructional content off teachers.

Prior research has shown that distributing educational videos to students has a positive impact on their learning [12, 17]. In these studies, the problem-specific videos were created by the researchers

and were designed to explain how to solve the specific mathematics problems for which they were provided. Building off this prior research, this work investigated if free and publicly available skill-related videos have a similar positive effect on students' learning. Videos aggregated from YouTube via automated searches were incorporated into the ASSISTments online learning platform [7] and provided to students upon their request. In addition to a randomized experiment investigating the effectiveness of these videos, multi-armed bandit algorithms (MABs) were used to identify which videos were most effective for each mathematics skill using the ASSISTments Automatic Personalized Learning Service (APLS) [15]. The effectiveness of the videos recommended via MAB were compared to randomly recommended videos to investigate the impact that MABs could have on the incorporation of these videos into online learning platforms.

Additionally, features of these videos, extracted using various machine learning APIs, were evaluated for their correlation with students' performance and for their ability to personalize students learning based on students' prior knowledge. For a feature to be capable of personalizing students' learning, there must be a qualitative interaction between the feature and prior knowledge. A qualitative interaction indicates that one group of students benefits more from one value of the feature while another group benefits more from a different value of the feature. For example, if high-knowledge students benefited more from long videos and low-knowledge students benefited more from short videos, then the video length feature could be used to personalize students' learning.

To summarize, this work answers the following research questions:

(1) What is the effect of incorporating publicly available skill-related videos into an online learning platform on students' performance?
(2) What is the effect of using multi-armed bandit algorithms to recommend videos on students' performance?
(3) What features of these videos are most predictive of students' performance?
(4) What qualitative interactions between video features and students' prior knowledge are most predictive of students' performance?

2 BACKGROUND

2.1 Instructional Videos

Instructional videos have been used successfully in the context of online learning many times. In a randomized controlled study in which the same problem-specific tutoring was provided to students in video or text format, it was shown that videos led to higher student performance than text [12]. Additionally, a combined analysis of five different randomized controlled experiments that compared video feedback to text feedback within the ASSISTments online learning platform found that videos were more effective than text across a variety of measures such as mastery speed and posttest score [17]. While these studies demonstrate the effectiveness of videos for problem-specific support, in this work we propose using videos to give more general, skill-related instruction.

Massive open online courses (MOOCs) are a good example of using videos not to provide specific feedback for individual problems, but to convey information on various topics in general. Many MOOCs feature videos in a wide variety of formats [19], from recordings of classroom lectures, to completely virtual presentations, to hybrid approaches, as well as various levels of integration with online assessments to enable students to practice as they learn. Not only do videos come in a variety of formats, but students use videos differently, and prefer videos formatted in a variety of ways. For example, a study of MITx MOOCs found that there was a distinct bimodal distribution in students' video usage across different courses, demonstrating differences in preference of how to use the MOOC videos [23]. Additionally, prior work has found that some students prefer classroom lecture recordings while others prefer fully digital presentations, and that these preferences are statistically significantly correlated with their motivation for enrolling in the MOOC [24]. While the study in this work is not done within a MOOC, these MOOC studies show the variety of formats and preferences for video-based content. The skill-related videos provided to students in this study may follow usage trends similar to the videos in MOOCs.

2.2 Multi-Armed Bandit Algorithms

Multi-Armed bandit algorithms (MABs) are a simple type of reinforcement learning where the algorithm takes one of multiple possible actions, is given a numeric reward based on criteria defined by the researcher, and models the relationship between each action and the expected reward. Over time, a MAB uses its model to try and maximize the reward it receives by taking actions with the highest expected reward [20]. MABs assume that the reward received for an action is independent of the sequence of actions taken, unlike more complicated reinforcement learning algorithms.

Research has shown in simulation that MABs would be able to increase students' learning during randomized experiments performed within online learning platforms, but would also increase the false-discovery rate of significant experiment results [18]. Although there are methods to adjust how a MAB operates to correct for some of the increase in false-discovery rate [25, 26], to avoid any bias, this work includes a randomized controlled experiment to investigate the effectiveness of providing students with skill-related videos. However, MABs have been shown via a large-scale randomized experiment to sightly improve students' performance by learning the most effective problem-specific support messages for middle-school mathematics problems [15]. Compared to randomly receiving one of multiple relevant problem-specific supports, students that received the support recommended by the MAB got the next problem in their assignment correct more often [15]. Therefore, to both maximize the benefit of skill-related videos and to study the effects of MABs on student performance in a different but similar context to the previous study, this work also studies the effect of using a MAB to recommend skill-related videos to students.

2.2.1 Thompson Sampling. The MAB used in this work is Thompson sampling. Thompson sampling was used in previous studies comparing MABs to random selection [15, 18] and has outperformed other algorithms when recommending content to students [15, 18]. Thompson sampling models the expected reward

of each action it can take as a distribution of the rewards it has received for that action before. Each time Thompson sampling receives a reward for taking an action, that reward is used to update the action's prior distribution. Thompson sampling selects which action to take by randomly sampling from each action's prior reward distribution, and then takes the action corresponding to the highest random sample [22]. By randomly sampling from the prior distributions, Thompson sampling balances learning more about actions that have not been taken frequently with taking actions that lead to the highest reward on average. At the beginning of Thompson sampling's use it will know very little about each action, and thus each prior distribution will have a high variance. The high variance will lead to random samples far from the mean reward of each action, which will make Thompson sampling's choice of action very similar to random selection. However, once each action has been taken many times, the variance of the prior reward distributions tends to decrease, and Thompson sampling will begin to take the action with the highest expected reward more frequently. The Thompson sampling algorithm used in this work is Beta-Bernoulli Thompson Sampling (BBTS), which models the prior distribution of a binary reward as a Beta distribution, and has been proven to be asymptotically optimal in [9]. BBTS has been used successfully in the past to recommend problem-specific support to students [15].

2.3 The ASSISTments APLS

The experiments in this work were performed within ASSISTments, an online learning platform that focuses on middle-school mathematics. Since 2021, ASSISTments has been able to use MABs to personalize the content provided to students through the Automatic Personalized Learning Service (APLS) [15]. The APLS allows for algorithms to make content recommendations for students in real-time. The APLS has the capacity to incorporate features of students, problems, and the content itself to its decision of what content to provide to a student. When multiple recommendation algorithms are available in the APLS, one is selected randomly, which enables randomized experiments between algorithms [15]. In this work, a random selection model and a BBTS model were added to the APLS for recommending videos. This way, the APLS administers the experiment comparing MABs to random selection, and the random selection model administers the experiments comparing videos.

Each night, the APLS calculates a reward for each recommendation made in the past 24-hours and updates each recommendation algorithm using these rewards. If a student was able to complete the next problem on their first try without any additional tutoring, the algorithm receives a reward of 1 for its video recommendation. Otherwise, the algorithm receives a reward of 0. In the studies in this work, the algorithms received rewards regardless of whether or not the student observed the skill-related video because both the random selection model and the BBTS model had the option to recommend no video. If a reward was only given when students viewed the videos, a reward could never be calculated for recommending no video. The downside of this is that the population of students that never observe the skill-related videos, while not biasing the prior reward distributions, add noise, making it more difficult to learn the differences in effectiveness between videos.

3 SKILL-RELATED VIDEOS

3.1 The Show Video Button

Prior to this work, ASSISTments only had the capacity to offer students problem-specific support. Given that it has been shown multiple times that the problem-specific support in ASSISTments benefits students [13, 17, 21], it would have been potentially detrimental to replace this problem-specific support with skill-related videos. Instead of replacing this tutoring, a new button was added to the ASSISTments Tutor. The ASSISTments Tutor is shown in Figure 1. Figure 2 shows the explanation, in yellow, that appears when a student clicks the Show Explanation button, which is the pre-existing button used to request problem-specific support. This tutoring only explains how to solve the specific problem on screen.

The new Show Video button is to the left of the Show Explanation button. When a student clicks on the Show Video button, a new tab containing a skill-related video opens in the student's web browser. Viewing a skill-related video does not directly explain how to solve the specific problem in the Tutor, and therefore, there is no penalty for requesting a skill-related video, unlike the problem-specific support, which removes a fraction of a student's score when requested. To familiarize students with the new Show Video button, an information icon, shown in Figure 1 directly to the left of the Show Video button, was provided. When students hover over the information icon, the message "Clicking this button does not reduce your score. It shows a video to help you solve the problem" is displayed. Figure 2 shows an example of a video[1] opened in a new tab when a student clicks the Show Video button.

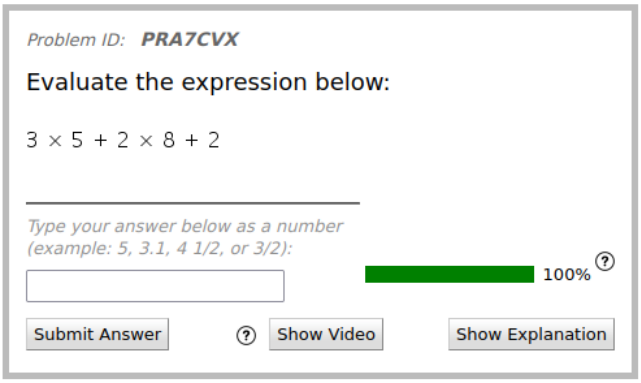


Figure 1: A mathematics problem in the ASSISTments Tutor. The new Show Video button appears to the left of the pre-existing Show Explanation button.

3.2 Video Incorporation

To incorporate skill-related videos into the ASSISTments APLS, the following steps had to be taken.

(1) Skill Labeling: Tag every problem in ASSISTments with the most relevant Common Core Skill Code [1].
(2) Video Filtering: Identify publicly available YouTube videos relevant to each skill.

[1] https://www.youtube.com/embed/xLcgug8iEYY

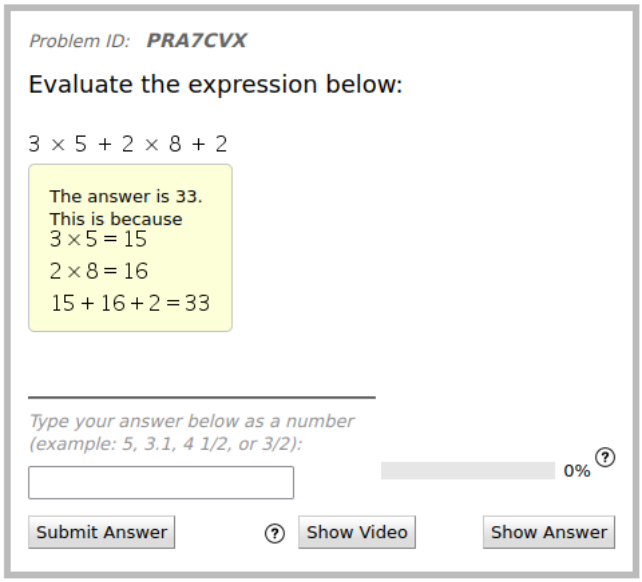


Figure 2: A mathematics problem in the ASSISTments Tutor with an explanation highlighted in yellow.

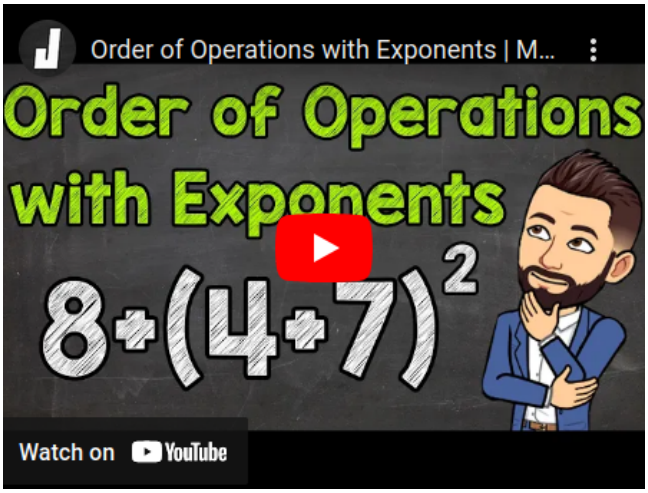


Figure 3: An example of a skill-related video.

(3) Feature Extraction: Create features of the videos and incorporate them into the APLS in order to investigate their impact on student performance.

3.2.1 Skill Labeling. The Common Core State Standards for Mathematics [1] discretize the United States mathematics curriculum into a tree of branching codes, where each leaf refers to a specific concept that students must learn during their mathematics education. For example, the Skill Code 7.G.A.1 refers to a 7th grade geometry problem (7.G). The letter A refers to a section of the 7th grade geometry curricula, specifically the section described as "Draw, construct, and describe geometrical figures and describe the relationships between them" The number 1 is the final part of the skill code which refers to the skill in section A described as "Solve problems involving scale drawings of geometric figures, including computing actual lengths and areas from a scale drawing and reproducing a scale drawing at a different scale"

For each 6th grade through 8th grade mathematics problem in the Engage New York[2], Illustrative Mathematics[3], and Utah Middle School Math Project[4] curricula, two teachers labeled each mathematics problem with the Common Core Skill Code most relevant to solving the problem. If the two teachers agreed, then that was the final skill code incorporated into ASSISTments. If the teachers disagreed, a third teacher was used to decide which of the two skill codes was correct. Essentially, two out of three teachers had to agree on the skill code for each mathematics problem before it was labeled. In total, 16,167 mathematics problems were tagged with their most relevant skill.

3.2.2 Video Filtering. After all the mathematics problems were tagged with their most relevant skill code, the skill code descriptions were used as the search term in YouTube in order to find relevant videos for each skill. The first ten results of each search were collected and shown to middle-school math teachers. The teachers were instructed to select the first five relevant videos for each skill. If less than five videos were relevant, then the teachers were instructed to go to YouTube and find the remaining videos themselves. Even though part of this work was to investigate how well BBTS would be able differentiate between more and less effective videos, the videos were still evaluated by teachers because at no point in this work would it have been acceptable for students to have been shown noneducational content. This process was used to find five relevant videos for each skill. The number five was chosen somewhat arbitrarily, with the goal of having enough videos for there to be variations between them, but few enough videos that BBTS would have time to learn the effectiveness of each video. In total, 1,315 videos were collected for 263 skills.

3.2.3 Feature Extraction. Once five videos for each skill were collected, a variety of machine learning APIs and YouTube metadata was used to create features for each video. Two APIs, Speechace[5] and DeepAffects[6], were used to extract features related to the voice of the speaker in the video if there was one. The Azure Face API[7] was used to examine qualities of the face in the video if the speaker included their face. Lastly, YouTube metadata from the video pages[8] was used to extract features related to the length and appeal of the videos. The number of dislikes for a video was made private by YouTube on November 10th, 2021[9], but these features were extracted prior to that change. Of the dozens of features available through these sources, 12 were included as features in the APLS and used for further analysis of the experimental results. If all the features had been included, the false discovery rate of features that significantly impact student performance would have been much higher. The following 12 features were chosen because of

[2] http://www.nysed.gov/curriculum-instruction/engageny
[3] https://illustrativemathematics.org/
[4] http://utahmiddleschoolmath.org/
[5] https://docs.speechace.com/
[6] https://docs.deepaffects.com/docs/introduction.html
[7] https://azure.microsoft.com/en-us/products/cognitive-services/face/
[8] https://www.youtube.com/
[9] https://blog.youtube/news-and-events/update-to-youtube/

their relevance to the educational quality of the videos, as determined qualitatively by a combination of middle-school mathematics teachers and researchers.

- **Length**: The length, in seconds, of the video, determined using YouTube metadata.
- **View Count**: The number of views of the video, determined using YouTube metadata.
- **Percent Likes**: The ratio of likes to views, determined using YouTube metadata.
- **Percent Dislikes**: The ratio of dislikes to views, determined using YouTube metadata.
- **Percent Comments**: The ratio of comments to views, determined using YouTube metadata.
- **Pronunciation Score**: A score from 0-100 that assesses how well the words in the video are pronounced, determined using Speechace API.
- **Unknown Pronunciation Score**: A binary indicator for whether or not Speechace was unable to calculate a pronunciation score.
- **Male Tone**: A binary indicator for whether or not the tone of the speaker sounded as though they were male, determined using the DeepAffects API.
- **Reading Tone**: A binary indicator for whether or not the tone of the speaker sounded as though they were reading, determined using the DeepAffects API.
- **Passionate Tone**: A binary indicator for whether or not the tone of the speaker sounded passionate, determined using the DeepAffects API.
- **Unknown Tone**: A binary indicator for whether or not DeepAffects was unable to analyse part of the tone.
- **Face Included**: A binary indicator of whether or not there was a face included in the video, determined using Azure Face API.

4 METHODOLOGY

4.1 Empirical Studies

Two randomized controlled experiments were performed using the ASSISTments APLS between March 3rd, 2022 and July 18th, 2022. The first experiment investigated the impact of skill-related videos on student performance, and the second experiment investigated the impact of using a MAB, specifically BBTS, to recommend skill-related videos compared to randomly recommending skill-related videos. Both studies were run simultaneously at the problem level, on different subsets of the student population. When a student started a problem, they were first randomized with equal probability between receiving a randomly recommended video or a BBTS recommended video. Students randomized to a BBTS recommended video were the treatment population for the second experiment, and BBTS was used to recommend one of the five relevant videos for the skill the problem was tagged with or no video (six options per recommendation). Students randomized to a randomly recommended video were the control population for the second experiment, and were randomly given one of the five relevant videos for the skill the problem was tagged with or no video with equal probability (1/6 chance of receiving each video, 1/6 chance of receiving no video). Students in the control population of the second experiment that were randomized to no video were considered the control population for the first experiment, and students randomized to any of the five videos were considered the treatment population.

Essentially, all students participated in the second experiment, and the half of students that were given randomly recommended videos participated in the first experiment as well. Both experiments were intent-to-treat analyses because the Show Video button was visible or not based on which condition a student was in. Because the presence of the button could have an effect on students' behavior, a student was included in the analysis if they were randomized into a condition, regardless of whether or not they viewed the skill-related video. Both experiments used next-problem correctness as the dependent measure. Correctness is a binary indication of whether the student got the problem correct on their first try without any additional support (1) or not (0). Next-problem correctness was chosen because it is an immediate measure that has been shown in prior work to be an effective surrogate for learning, and it correlates with other measures of learning such as posttest score and mastery speed[13, 15–17]. Additionally, while one could use students' engagement with the videos as a dependent measure, e.g., number of videos requested or time spent watching videos, students' preferences do not always correlate with their learning [6, 17]. Therefore, next-problem correctness was chosen, as it provides an immediate and effective measure of learning.

4.1.1 Video Vs. No Video Analysis. To analyse the results of the first experiment, a mixed-effects logistic regression model [4] was fit to predict students next-problem correctness given the following inputs.

(1) A constant.
(2) The average correctness of the student across the prior weeks problems.
(3) The average correctness of the problem a skill-level video was provided (or not provided) for across the prior weeks instances of students completing the problem.
(4) The average correctness of the next problem used to calculate the dependent measure across the prior weeks instances of students completing the problem.
(5) A binary indication of whether or not the student was in the treatment (1) or control (0) condition.
(6) A random effect for each skill's impact on the treatment effect.

Inputs 2, 3, and 4 are all covariates meant to remove variations in the results from students with different prior knowledge and problems of different difficulty. Input 5 measures the average effect of offering students the opportunity to request a skill-related video, and each of the skill-level random effects in Input 6 measures the effect of offering students the opportunity to request a skill-related video for each skill separately. The random effects were included because each skill has a different set of five videos available for it, and it could be that some skills had very helpful videos while other skills did not, which would not be captured by Input 5.

The coefficient and statistical significance of Input 5 can be used to measure the impact of providing students with the opportunity to request skill-related videos on their performance, and the coefficients and statistical significance for the random effects can be used to determine the skill-level impact of this new feature.

4.1.2 BBTS Vs. Random Selection Analysis. To analyse the results of the second experiment, a mixed-effects logistic regression model [4] was fit to predict students next-problem correctness given the same inputs as the mixed-effects model for the first experiment but with the treatment variable now being whether or not BBTS (1) or random selection (0) was used to determine which video was made available to the student, and the following additional inputs.

(1) The number of recommendations made so far by the selected model for the given skill.
(2) The interaction between Input 1 and whether or not the student was in the treatment (1) or control (0) condition.
(3) A random effect for each skill's impact on Input 1.
(4) A random effect for each skill's impact on Input 2.

Unlike the first experiment, where we do not expect the effect of having a video available to change over time, we do expect the effect of the videos provided through BBTS to change over time compared to randomly selected videos because at the beginning of BBTS's use, it makes basically random recommendations, but over time, BBTS learns which videos are most effective and offers them to students more often.

The coefficient and statistical significance of Input 2 captures this change over time and measures the impact of using BBTS to select videos compared to randomly selecting videos. The mixed effects in Input 4 capture how the impact of using BBTS to select videos changes for each skill.

4.2 Video Feature Analysis

In addition to measuring the impact that videos and the methods used to select them have on student performance, this work used the data from the first experiment to investigate what features of videos made them more or less effective for the students that requested them. A logistic regression model [10] was fit using only the data from samples where students viewed the randomly recommended videos to predict students' next problem correctness given the following inputs.

(1) A constant.
(2) Random effects for the average effectiveness of videos for each skill.
(3) The average correctness of the student across the prior weeks problems.
(4) The average correctness of the problem a skill-level video was provided (or not provided) for across the prior weeks instances of students completing the problem.
(5) The average correctness of the next problem used to calculate the dependent measure across the prior weeks instances of students completing the problem.
(6) All of the video features except for Unknown Pronunciation Score and Unknown Tone.

In the regression, Inputs 1 and 2 allow for the average likelihood of getting the next problem correct after viewing a video to vary based on skill. This is important because different skills could be easier or harder to explain via video, and the model should be able to take this into account. Inputs 3, 4, and 5 are covariates to account for the variance in students' propensity to get the next problem correct. The video features "Unknown Pronunciation Score" and "Unknown Tone" were excluded from the logistic regression because many factors could have influenced either of the video feature APIs abilities to extract features, and these features being significant would not be an interpretable finding. Considering that every feature investigated for its impact on student learning increases the severity of the hypothesis correction used in this analysis, these two features were intentionally left out.

The coefficients and confidence intervals of the video features were used to determine if they had an impact on student performance. The Benjamini-Hochberg procedure [2] was used to correct the false discovery rate of significant features.

4.3 Opportunities for Personalization

In addition to exploring the impact that different video features had on students' performance, this work used the data from students that requested randomly selected videos to look for qualitative interactions between features of the videos and the students' prior knowledge. A qualitative interaction exists if one group of students benefits more from one type of content, while another group of students benefits more from a different type of content. For example, a qualitative interaction between students' prior knowledge and video length would exist if high-knowledge students got the next problem correct more often after viewing long videos and low-knowledge students got the next problem correct more often after viewing short videos. These qualitative interactions are each an opportunity to personalize students' learning. To identify any qualitative interactions in the data, the same method used in [15] to identify statistically significant qualitative interactions between students and the content available to them was used. Using this method, the regression $y = \beta_0+\beta_1 x_1+\beta_2 x_2+\beta_3(x_1 \oplus x_2)$ is fit, where x_1 is a video feature, converted to a binary indicator of whether or not the value is above or below average for that feature, x_2 is a binary indicator of whether or not the student's prior correctness is above or below average, and y is the student's next problem correctness. Using this model, a qualitative interaction exists if β_3^2 is greater than β_1^2, which is derived with more detail in [15]. p-values for the statistical significance of these qualitative interactions were calculated using a bootstrapping approach in which the regression above was fit 10,000 times on different samples of equal size to the original data sampled with replacement from the original data. The distribution of $\beta_3^2 - \beta_1^2$ was used to perform a one-sample t-test to determine the p-value of the null hypothesis: $\beta_3^2 - \beta_1^2 \leq 0$. The p-values for the significance of different video features' qualitative interactions were corrected for multiple hypothesis testing using the Benjamini-Hochberg procedure [2].

5 RESULTS

From March 3rd, 2022 to July 18th, 2022, 479,032 video recommendations were made to 18,267 students as they completed one of 27,589 problems. More problems were included in the experiments than were tagged for this work because some problems in ASSISTments were already tagged with their most relevant skill. On average, about 1,835 recommendations were made per skill, and each video was recommended an average of about 369 times. Unfortunately, out of all these recommendations, only 3,196 videos were actually requested by students. The vast majority of the time, students did not request videos. Compared to the about 15% of the time that

students request problem-specific support, students only requested skill-related videos about 0.7% of the time.

Of the 2,383 students that requested at least one video, only 22% percent of those students requested a second, and less than 1% of those students requested at least 5 videos. Figure 4 shows this trend in skill-related video requests compared to problem-specific support requests. Students were not only less interested in skill-related videos from the start, but after requesting one video, students were much less likely to request another compared to the trend for problem-specific supports. Additionally, about 51% percent of the time that a video was requested, the problem-level support for the same problem was requested afterwards. Due to the intent-to-treat design of the randomized experiments, students' lack of interest in videos added a tremendous amount of noise to the results.

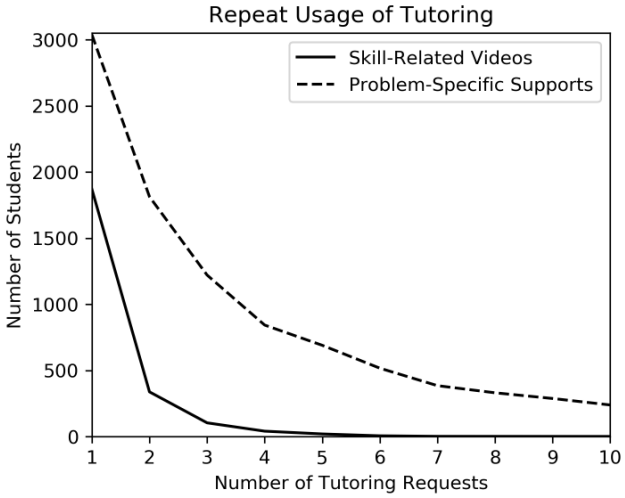


Figure 4: The number of students that requested from one to ten instances of tutoring for both skill-related videos and problem-specific support.

5.1 Video Vs. No Video

In the first experiment, 280,646 samples of a student being randomized when they started a problem between having the option to request a skill-related video or not were collected. In the control condition, there were 46,707 instances of one of 11,840 students completing one of 13,491 problems without the option to request a video. In the treatment condition, there were 233,939 instances of one of 16,974 students completing one of 23,119 problems with the option to request a video. There are more samples in the treatment than the control because students were randomized with equal probability to each of the five relevant videos or no video. Therefore, there are about five times more samples in the treatment condition than the control.

Using the model described in Section 4.1.1, the coefficient and 95% confidence interval for the average treatment of being shown a video was about 0.0002 ± 0.0250, which is far from being statistically significant. Figure 5 shows the coefficients and confidence intervals for the random effects of being offered a skill-related video for each skill separately, sorted from lowest to highest coefficient. Even when examining the effect of offering students skill-related videos on a per-skill basis, there were no significant effects. The model fit to determine these coefficients was a logistic regression, so the coefficients in Figure 5 should not be interpreted as effect sizes, they should solely be interpreted as indications that there were no statistically significant effects, which makes determining effect size moot.

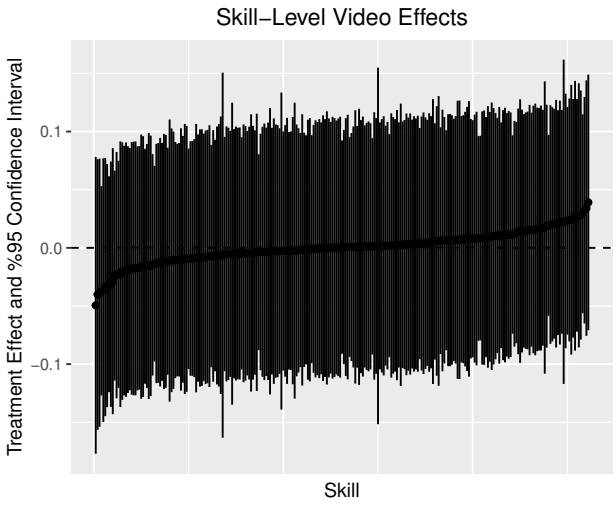


Figure 5: The coefficients and 95% confidence intervals for the random effects of offering students skill-related videos compared to not offering videos, sorted from lowest to highest coefficient.

5.2 BBTS Vs. Random Selection

In the second experiment, 559,917 samples of a student being randomized when they started a problem were collected. Students were randomized between BBTS or random selection determining which video (or lack thereof) they could request. In the control condition, there were 280,646 instances of one of 17,377 students completing one of 24,276 problems with the option to request a randomly recommended video. In the treatment condition, there were 279,271 instances of one of 17,309 students completing one of 24,315 problems with the option to request a BBTS recommended video. There are about an equal number of samples in each condition because students were randomized with equal probability to receive BBTS recommendations or random recommendations.

Using the model described in Section 4.1.2, the coefficient and 95% confidence interval for the average impact over time of using BBTS to recommend videos was about -0.10 ± 0.14, which is again, far from being statistically significant. Figure 6 shows the coefficients and confidence intervals for the random effects of the impact over time of using BBTS to recommend videos for each skill separately, sorted from lowest to highest coefficient. Even when examining the effect of using BBTS to recommend videos on a per-skill basis, there were no significant effects. The model fit to determine these

coefficients was a logistic regression, so the coefficients in Figure 6 should not be interpreted as effect sizes, they should solely be interpreted as indications that there were no statistically significant effects, which makes determining effect size moot.

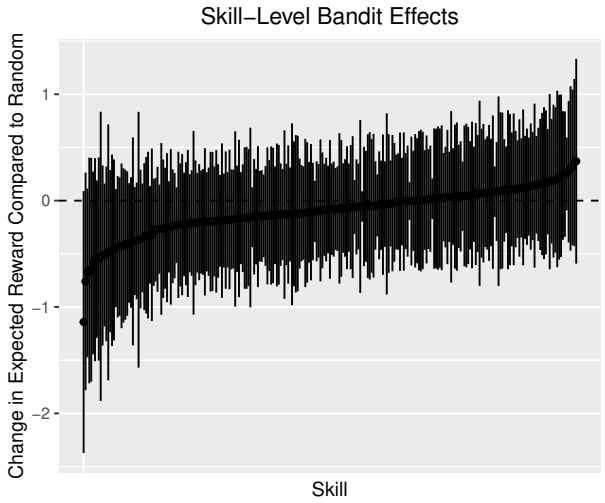


Figure 6: The coefficients and 95% confidence intervals for the random effects of the impact over time of using BBTS to recommend videos compared to randomly recommending videos, sorted from lowest to highest coefficient.

5.3 Video Features

In total, 1,677 randomly recommended videos were requested by 1,372 different users across 1,303 problems. Using the model described in Section 4.2, Figure 7 shows the coefficients and 95% confidence intervals for the video features. The confidence intervals in Figure 7 are calculated prior to any hypothesis correction. After hypothesis correction using the Benjamini-Hochberg procedure [2], none of the video features were significant predictors of students' next-problem correctness. The model fit to determine these coefficients was a logistic regression, so the coefficients in Figure 7 should not be interpreted as effect sizes, they should solely be interpreted as indications of which features were significant prior to correcting for multiple hypotheses.

5.4 Opportunities for Personalization

Using the methodology described in Section 4.3, of the ten potential qualitative interactions between students' prior knowledge and video features, two qualitative interactions were present and statistically significant. Both qualitative interactions are shown in Figure 8. In both plots, students with above-average prior correctness outperform students with below-average prior correctness on average, regardless of video features. However, students with below-average prior correctness benefited more from videos with above-average pronunciation scores and male toned speakers while students with above-average prior correctness benefited more from videos with below-average pronunciation scores and non-male toned speakers.

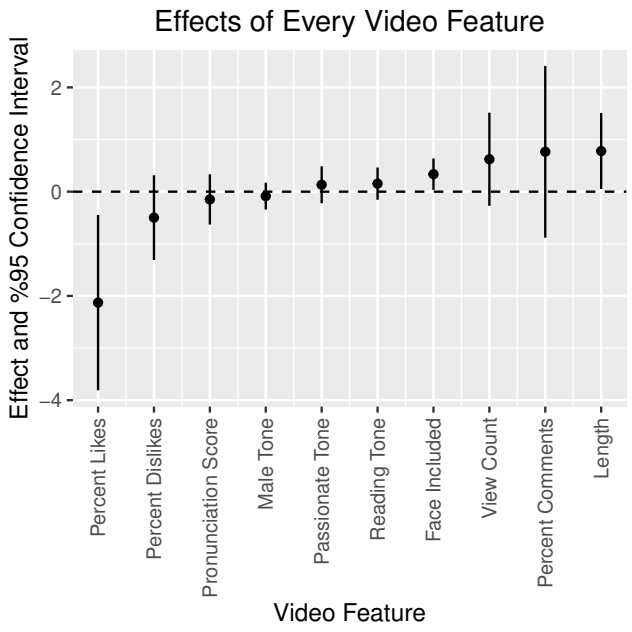


Figure 7: The coefficients and 95% confidence intervals for the impact of each video feature on students' propensity to get the next problem correct.

While these findings are statistically significant (both have $p < 0.001$ after correction), they are only correlational. If all other features of the videos were held constant, and the only difference was the speakers tone or pronunciation, then it would be possible to look for causality, but this is not the case for these skill-related YouTube videos. There are likely many covariates outside of the feature set created in this work that are correlated with pronunciation score and tone that effect these results. However, finding any opportunities to personalize students' learning at scale is rare, and it is interesting that even though so few students seemed to engage with the skill-related videos, there were still significant differences between the effectiveness of certain videos for specific groups of students.

6 DISCUSSION

From this work it seems that students are not interested in engaging with skill-related videos. It is unlikely that students were uninterested in the videos simply because they were videos because prior research in ASSISTments offered students a choice between video-based or text-based problem-specific support and found that about 29% of students chose the videos [5]. The presence of problem-specific support, which is more direct, relevant, and shorter, likely made students see the extra videos as a waste of time. Even though viewing the problem-specific support lowered students' scores while the skill-related videos did not, most students use ASSISTments for in-class work or homework assignments, which are generally low pressure assignments meant to help prepare them for tests that are more impactful to their grades. Students might not care about their homework score and prioritize

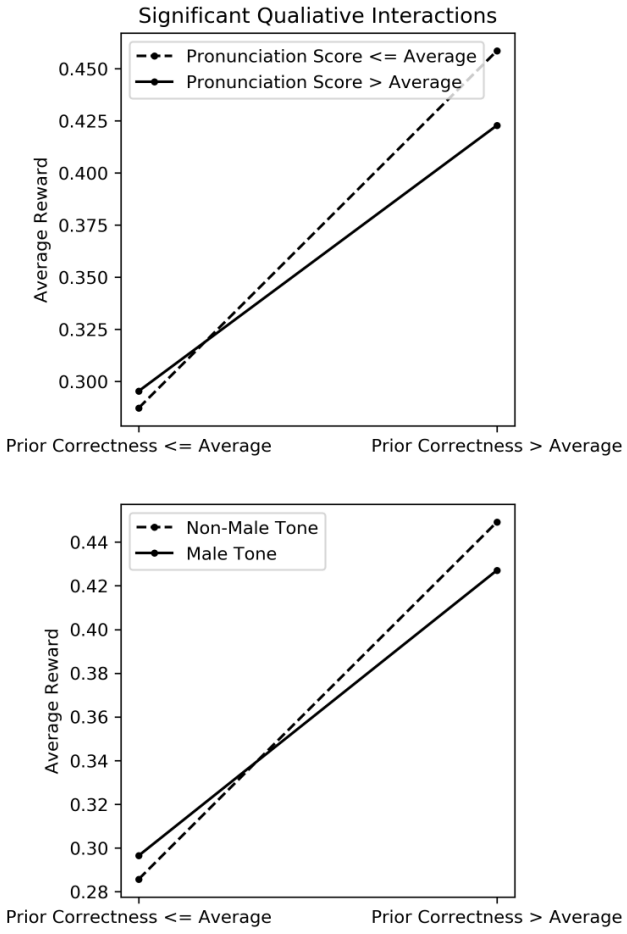


Figure 8: The two significant qualitative interactions between students' prior correctness and video features.

getting the most direct and relevant advice over general advice that may or may not be as helpful. An important distinction between the videos in this work and the videos in MOOCs is that MOOC videos are meant to be the primary instructional material, whereas in this work the videos were supplemental instructional material. This likely had an impact on students motivation to engage with the videos because their teachers were probably providing them with primary instruction in a way they were more familiar and engaged with.

Regarding the analysis, using an intent-to-treat design made it very difficult to observe any effect of skill-related videos or of using BBTS to recommend them. Students only requested a video about 0.7% of the time. Unless seeing that a video is available but not requesting it effects students' propensity to get the next problem correct, 99.3% of the data in the treatment condition was equivalent to the data in the control condition. The amount of noise this adds to the analyses made the confidence intervals too large to see any effects, even on a per-skill basis.

By only including data from instances where students requested a randomly recommended video, this work was able to investigate the impact that different video features had on student performance. This part of the analysis was not an intent-to-treat design, and instead looked only at the impact that the videos had on the treated, i.e., the students that requested them. Interestingly, even though no video feature was a significant predictor of students' next-problem correctness, two video features, Male Tone and Pronunciation Score, had a significant qualitative interaction with students' prior correctness. These findings are almost certainly not causal because other features of the videos were not controlled for. Students with below-average prior correctness benefited more from videos with above-average pronunciation scores and male toned speakers while students with above-average prior correctness benefited more from videos with below-average pronunciation scores and non-male toned speakers. There were a handful of videos in this study in which a woman with a southern accent effectively explained a variety of mathematics skills. It likely that this woman, and similar content creators in the data, happen to explain concepts at a level that was more appropriate for students with higher knowledge, and because this woman has a lower pronunciation score and a non-male tone, the data reflects that these features have qualitative interactions with students' prior knowledge. In reality it is likely not the features themselves that led to these qualitative interactions, but the content creators that happened to correlate with those features.

7 LIMITATIONS AND FUTURE WORK

The results of these studies do not imply that skill-related videos are ineffective, but rather that there was no effect in this particular use case. This work only looked at the impact of skill-related videos on middle-school mathematics students within ASSISTments. It could be that without the problem-specific support that ASSISTments provides, skill-related videos would have a larger effect. It could also be that different age or socioeconomic groups are impacted differently than the population in this study. More studies should be conducted to investigate the impact of skill-related videos in different contexts, and to ensure that if there is an impact in a particular context, that this impact is fairly distributed amongst different groups of students.

While the intent-to-treat analysis was necessary to unbiasedly compare videos to no videos, it was not as necessary to investigate the impact of using BBTS to recommend videos compared to random selection. If BBTS was not allowed to recommend no video, then BBTS could have been updated only when students actually requested videos, and these samples could have been compared to only the times that students requested randomly recommended videos. This would have likely resulted in a larger effect by removing about 99.3% of the data used to updated the BBTS model in which students never requested videos. This would have allowed the BBTS model to learn the trends in the data more easily, and likely led to a larger difference over time between BBTS recommendations and random recommendations. Moving forward, more experiments comparing BBTS to random selection in ways that are more fair to BBTS should be conducted.

Additionally, better covariates for predicting students' next-problem correctness could be created to help remove some of the noise in the intent-to-treat analysis. The covariates used in all the

models in this work for students' prior knowledge and problem difficulty had Pearson correlations [14] with students' next-problem correctness of only around 0.2. Serious work could be done to thoroughly investigate different combinations of student and problem past performance measures in order to create more predictive covariates.

Lastly, the videos in this work were collected from YouTube via algorithmic searches and teacher ratings. If, in the future, one wished to perform a causal analysis of the significance of different video features and their qualitative interactions with students, it would be better to create the videos from scratch. If everything except one video feature of interest was held constant, the analyses in Sections 4.2 and 4.3 could be regarded as causal for that feature.

8 CONCLUSION

Overall, it did not appear that offering students the option to request skill-related videos had a positive impact on their performance. This mostly stemmed from students' lack of interest in the skill-related videos. Students only requested a skill-related video about 0.7% of the time, compared to the about 15% of the time that they requested problem-specific tutoring, which implies they would much prefer concise advice directly related to the task at hand, regardless of the impact it has on their score. Although this work did not show any significant impact of providing skill-related videos to students, it was able to analyse which features of videos correlated most with students' performance when they did request a video. This analysis found that while there were no video features that significantly predicted students' performance, there were two video features that had qualitative interactions with students' prior knowledge. These qualitative interactions implied that particular content creators created videos that were more helpful for higher-knowledge students, while other content creators made videos that were more effective for lower-knowledge students. Moving forward, the educational research community can take away two main findings from this work. The first is that students are unlikely to be interested in content that they do not see as directly relevant to them. Therefore, when creating or curating tutoring for students, taking the effort to ensure each piece of content is direct and relevant is likely to pay off. Secondly, it seems possible to create videos that are better for higher or lower knowledge students. This should motivate randomized controlled studies to determine which aspects of video based learning specifically influence videos' effectiveness for different groups of students. Uncovering the causal mechanisms behind these qualitative interactions paves the way for more effective forms of personalized learning.

ACKNOWLEDGMENTS

We would like to thank NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, & R305A120125), GAANN (e.g., P200A180088 & P200A150306), EIR (U411B190024 & S411B210024), ONR (N00014-18-1-2768), NHI (R44GM146483), and Schmidt Futures. None of the opinions expressed here are that of the funders.

REFERENCES

[1] Murat Akkus. 2016. The Common Core State Standards for Mathematics. *International Journal of Research in Education and Science* 2, 1 (2016), 49–54.
[2] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57, 1 (1995), 289–300.
[3] Joshua A Dijksman and Salman Khan. 2011. Khan Academy: the world's free virtual school. In *APS March Meeting Abstracts*, Vol. 2011. A14–006.
[4] Andrew Gelman and Jennifer Hill. 2006. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.
[5] Ashish Gurung. 2021. *Examining Student Effort on Hint through Response Time Decomposition*. Ph. D. Dissertation. Worcester Polytechnic Institute.
[6] Aaron Haim and Neil Heffernan. 2022. Student Perception on the Effectiveness of On-Demand Assistance in Online Learning Platforms. In *Educational Data Mining Conference*.
[7] Neil T Heffernan and Cristina Lindquist Heffernan. 2014. The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education* 24, 4 (2014), 470–497.
[8] Nina Hollender, Cristian Hofmann, Michael Deneke, and Bernhard Schmitz. 2010. Integrating cognitive load theory and concepts of human–computer interaction. *Computers in human behavior* 26, 6 (2010), 1278–1288.
[9] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. 2012. Thompson sampling: An asymptotically optimal finite-time analysis. In *International conference on algorithmic learning theory*. Springer, 199–213.
[10] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. 2002. *Logistic regression*. Springer.
[11] Rebecca Mullen and Linda Wedwick. 2008. Avoiding the digital abyss: Getting started in the classroom with YouTube, digital stories, and blogs. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas* 82, 2 (2008), 66–69.
[12] Korinn Ostrow and Neil Heffernan. 2014. Testing the multimedia principle in the real world: a comparison of video vs. Text feedback in authentic middle school math assignments. In *Educational Data Mining 2014*.
[13] Thanaporn Patikorn and Neil T Heffernan. 2020. Effectiveness of crowd-sourcing on-demand assistance from teachers in online learning platforms. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*. 115–124.
[14] Karl Pearson. 1895. VII. Note on regression and inheritance in the case of two parents. *proceedings of the royal society of London* 58, 347-352 (1895), 240–242.
[15] Ethan Prihar, Aaron Haim, Adam Sales, and Neil Heffernan. 2022. Automatic Interpretable Personalized Learning. In *Proceedings of the Ninth ACM Conference on Learning@ Scale*. 1–11.
[16] Ethan Prihar, Thanaporn Patikorn, Anthony Botelho, Adam Sales, and Neil Heffernan. 2021. Toward Personalizing Students' Education with Crowdsourced Tutoring. In *Proceedings of the Eighth ACM Conference on Learning@ Scale*. 37–45.
[17] Ethan Prihar, Manaal Syed, Korinn Ostrow, Stacy Shaw, Adam Sales, and Neil Heffernan. 2022. Exploring Common Trends in Online Educational Experiments. In *Proceedings of the 15th International Conference on Educational Data Mining*. 27.
[18] Anna Rafferty, Huiji Ying, Joseph Williams, et al. 2019. Statistical consequences of using multi-armed bandits to conduct adaptive educational experiments. *Journal of Educational Data Mining* 11, 1 (2019), 47–79.
[19] Jan Renz, Matthias Bauer, Martin Malchow, Thomas Staubitz, and Christoph Meinel. 2015. Optimizing the video experience in moocs. In *EDULEARN15 Proceedings*. IATED, 5150–5158.
[20] Herbert Robbins. 1952. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.* 58, 5 (1952), 527–535.
[21] Jeremy Roschelle, Mingyu Feng, Robert F Murphy, and Craig A Mason. 2016. Online mathematics homework increases student achievement. *AERA open* 2, 4 (2016), 2332858416673968.
[22] Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. 2018. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning* 11, 1 (2018), 1–96.
[23] Daniel T Seaton, Sergiy Nesterko, Tommy Mullaney, Justin Reich, Andrew Ho, and Isaac Chuang. 2014. Characterizing video use in the catalogue of MITx MOOCs. *Proceedings of the European MOOC Stakeholder Summit* (2014), 140–146.
[24] Doraisamy Gobu Sooryanarayan and Deepak Gupta. 2015. Impact of learner motivation on mooc preferences: Transfer vs. made moocs. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 929–934.
[25] Jiayu Yao, Emma Brunskill, Weiwei Pan, Susan Murphy, and Finale Doshi-Velez. 2021. Power Constrained Bandits. In *Machine Learning for Healthcare Conference*. PMLR, 209–259.
[26] Yang Zhi-Han, Shiyue Zhang, and Anna Rafferty. 2022. Adversarial bandits for drawing generalizable conclusions in non-adversarial experiments: an empirical study. In *Proceedings of the 15th International Conference on Educational Data Mining*. 353.

Chapter 3.3

A Bandit you can Trust

A Bandit You Can Trust

ETHAN PRIHAR, Worcester Polytechnic Institute, USA
ADAM SALES, Worcester Polytechnic Institute, USA
NEIL HEFFERNAN, Worcester Polytechnic Institute, USA

This work proposes Dynamic Linear Epsilon-Greedy, a novel contextual multi-armed bandit algorithm that can adaptively assign personalized content to users while enabling unbiased statistical analysis. Traditional A/B testing and reinforcement learning approaches have trade-offs between empirical investigation and maximal impact on users. Our algorithm seeks to balance these objectives, allowing platforms to personalize content effectively while still gathering valuable data. Dynamic Linear Epsilon-Greedy was evaluated via simulation and an empirical study in the ASSISTments online learning platform. In simulation, Dynamic Linear Epsilon-Greedy performed comparably to existing algorithms and in ASSISTments, slightly increased students' learning compared to A/B testing. Data collected from its recommendations allowed for the identification of qualitative interactions, which showed high and low knowledge students benefited from different content. Dynamic Linear Epsilon-Greedy holds promise as a method to balance personalization with unbiased statistical analysis. All the data collected during the simulation and empirical studies is publicly available at https://osf.io/zuwf7/.

CCS Concepts: • **Applied computing** → **Education**; **Distance learning**; **Computer-assisted instruction**.

Additional Key Words and Phrases: Contextual Bandit Algorithms, Online Learning, Empirical Studies

ACM Reference Format:
Ethan Prihar, Adam Sales, and Neil Heffernan. 2018. A Bandit You Can Trust. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 15 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Online learning platforms have become significantly more popular in recent years due to the prevalence of technology in the classroom and the transition to remote learning due to the global pandemic [15]. This has allowed students that would have otherwise been unable to attend class to receive instruction and enabled researchers to perform large-scale investigations into various instructional methods. However, these opportunities have come with challenges.

There are countless choices to be made when structuring online instruction. Should lessons be student-pace or teacher-paced? Should the assignments have multiple-choice or open-ended questions? What criteria should be used to determine when a student has mastered the material? When students are struggling, what kind of assistance should be provided?

Researchers have attempted to answer many of these questions using randomized experiments (A/B testing) integrated into online learning platforms [20, 25], but these learning platforms must balance scientific inquiry with social responsibility. If researchers are experimenting with new and potentially beneficial instructional interventions, then the control students who do not receive the beneficial intervention are being treated unfairly. In an attempt to counteract this unfair treatment of students, researchers have proposed using multi-armed bandit algorithms (MABs) to mediate

Manuscript submitted to ACM

which interventions are given to students [18, 21, 26]. MABs learn over time which interventions are most effective, and transition from assigning interventions uniformly to recommending the most effective interventions.

Using MABs has the potential to remedy the unfair treatment of students, but doing so causes other problems. MABs adjust which interventions they assign based on prior assignments. Therefore, assignments are not independent of each other, which prevents statistical methods such as t-tests or ANOVAs from being used because they require samples to be independent of each other. Some researchers have proposed modifications to MABs that make the data they collect more similar to an experiment [29, 30], but these modifications only help to identify the most effective treatment for students on average.

To personalize students' learning, the algorithm used to assign treatments must be able to learn qualitative interactions between students and interventions. A qualitative interaction exists when different groups of students each benefit from different interventions [19]. Qualitative interactions can exist for individual students and interventions, e.g., Student A benefits most from Intervention 1, or on a student and intervention feature basis, e.g., Students that take longer than average to answer questions benefit more from multiple-choice problems. Researchers are particularly interested in these feature-based qualitative interactions because they can generalize beyond a specific experiment and have a much greater impact on the pedagogue of online learning.

In order to find qualitative interactions while still gaining the advantages of using MABs, contextual MABs (CMABs) can be used. Unlike MABs, which learn the average effectiveness of each intervention, CMABs learn how to estimate the effectiveness of an intervention given information on a student, their learning environment, and the interventions itself. CMABs are capable of personalizing students' experiences, but, like MABs, bias common statistical methods by creating dependence between samples.

In this paper, we propose Dynamic Linear Epsilon-Greedy (DLEG), a novel adaptation of established CMAB methods that allows for students to receive personalized interventions while identifying valid, unbiased, generalizable qualitative interactions between features of students and the interventions available to them. We first demonstrate in simulation the effects of using DLEG compared to the most widely used CMABs. Then, we evaluate DLEGs ability to improve student learning while discovering generalizable qualitative interactions in a three month long empirical study on 3,602 real students during regular instruction within an online learning platform.

In this work, we make the following contributions.

(1) We propose Dynamic Linear Epsilon-Greedy (DLEG), a novel contextual multi-armed bandit algorithm (CMAB) designed balance the needs of students and researchers.
(2) We compared DLEG to the most well established existing CMABs in simulation.
(3) We empirically evaluated DLEG's ability to help students in a large-scale study.
(4) We empirically evaluated DLEG's ability to discover opportunities to personalize students' learning at-scale within this study.

2 BACKGROUND

2.1 Multi-Armed Bandit Algorithms

Multi-Armed bandit algorithms (MABs) are a class of reinforcement learning algorithm in which the algorithm, or agent, is presented with multiple actions it can take. The agent takes one of the possible actions, and is given a numeric reward based on criteria defined by the researcher. The agent learns over time the relationship between the actions it can take and the reward it receives, and uses this knowledge to try and maximize the reward it receives by taking

actions it thinks will lead to a high reward [23]. MABs differ from other more complicated reinforcement learning algorithms because they assume that the reward received for an action is independent of the sequence of actions taken.

In previous work, researchers have shown that MABs were able to increase students' learning during randomized experiments performed within an online learning platform, but that MABs added bias and increased the false positive rate of the following experiment analyses [21]. Some researchers have developed methods of bounding the behavior of MABs [29, 30] in order to make them behave more like a randomized experiment. However, this prior work focused on making MABs more interpretable, but not on identifying opportunities to personalize students' learning.

2.1.1 Contextual Multi-Armed Bandit Algorithms. In this work we focus on contextual multi-armed bandit algorithms (CMABs). CMABs expand upon MABs by incorporating information about the agent's environment, or context, into its decision of what action to take. This context allows users' recommendations to be personalized [4] by learning the relationship between users context and the expected reward.

One challenge when designing a CMAB is to choose a model that can accurately identify relationships between features of the context, the actions, and the reward. Some models, like neural networks, can be very powerful but difficult to interpret. A detailed look at various neural-network based CMABs can be found in [22]. Other models, like linear regressions, are easier to interpret but must have non-linear interactions explicitly engineered into the model. Two of the most well known CMABs, LinUCB [11] and Linear Thompson Sampling [1], both use a ridge regression. A major advantage of using a ridge regression is that it can be updated from a stream of data, i.e., these CMABs do not need a complete history of all the contexts, actions, and rewards they have observed to update their models.

Another challenge is to balance learning about the relationships between the context, actions, and reward with taking the actions that the CMAB expects will lead to the highest reward. This balance is often referred to as the exploration-exploitation trade-off [2]. A naive approach to addressing this balance is to take a random action a predetermined percent of the time, and otherwise take the action with the highest expected reward. This method is called ϵ-greedy, where ϵ is the percent of time a random action is taken, and the greedy action is the action with the highest expected reward. The ϵ-greedy method is not optimal because theoretically, the CMAB will eventually collect enough data to know with certainty which actions will lead to the highest reward at which point it is unnecessary to take any more random actions. Often, the exploration-exploitation trade-off is addressed using a variant of an Upper Confidence Bound (UCB) [10], or Thompson Sampling (TS) [24, 27] algorithm.

Both UCB and TS use the estimated reward for each possible action as well as a measure of the uncertainty of the estimate to determine which action to take. UCB adds to the estimated reward of an action inversely proportional to how many times previously the action was taken, and calls this value the upper confidence bound of that action. UCB then takes the action with the highest upper confidence bound [10]. TS uses the estimated reward and the variance of this estimate for each action to randomly sample from each possible action's prior reward distribution. TS then takes the action corresponding to the highest-valued random sample [24]. Both UCB and TS start by making mostly random decisions, but as the error of their estimates decreases, they converge to selecting the action with the highest estimated reward.

The downside of using UCB or TS is that actions are always taken based on prior observations, which biases the data collected during these algorithms use, making it unsuitable for typical statistical analyses to compare the effects of the actions. For this reason, in this work we modify the ϵ-greedy method such that it behaves similarly to UCB and TS while still collecting some independently sampled data for statistical analysis.

2.2 ASSISTments

In this work, both studies were performed using data from, or within the ASSISTments online learning platform. ASSISTments is an online learning platform with over 100,000 active student users that focuses on middle-school mathematics. In ASSISTments, teachers assign problem sets from open source mathematics curricula. Students then complete the assignments in the ASSISTments Tutor [8]. When students are struggling they can request to view a video relevant to the skills required to solve the problem, or they can request a hint or explanation directly relevant to the specific problem.

2.2.1 Skill-Level Videos. When a student requests a skill-level video, they are shown a YouTube video related to the skills required to solve their problem. In ASSISTments, each problem is tagged with its most relevant Common Core State Standards for Mathematics Skill Code [13], and five videos are available for each skill code. The student will receive the same video for a specific problem even if they press the button multiple times, but can receive different videos on other problems of the same skill.

2.2.2 Problem-Level Support. Between two and four problem-level supports are available for most of the mathematics question in ASSISTments [16] in the form of sets of hints or explanations. Sets of hints are composed of multiple small pieces of advice that the student must request one at a time and do not reveal the answer. Explanations contain a complete solution to the problem and the correct answer. Based on what is available, the student can request hints or an explanation, but never both for the same problem. Sets of hints and explanations will impact a student's score when they are requested, but hints remove a fraction of a student's score for each hint requested, and explanations remove all of a student's score upon request [16].

2.2.3 The Automatic Personalized Learning Service. The ASSISTments platform has developed the Automatic Personalized Learning Service (APLS) in order to use MABs to recommend both skill-level videos and problem-level supports to students [18]. The APLS operates in real-time by responding to requests from the ASSISTments Tutor. In these requests, the tutor provides the APLS with unique identifiers for the student, the problem, and the available content. The APLS uses these identifiers to look up features of the student, problem, and content, compiles these features into context, and then uses a recommendation algorithm to select content for the student. The APLS randomly chooses from multiple recommendation algorithms each time it makes a recommendation, which enables randomized experiments between algorithms [18]. In this work, we used the APLS to compare random recommendations to recommendations made by Dynamic Linear ϵ-Greedy.

In the APLS, each recommendation algorithm receives a reward of 1 when the student gets the next problem correct without any additional support after viewing the algorithm's recommended content, and 0 when they do not. When no information on the student's next-problem correctness is available, the recommendation is not used to update the algorithm. The APLS calculates these rewards every day in the evening during low load periods in order to not interrupt users' experience. After updating each algorithm with the rewards it received for each recommendation it made since the last update, the APLS uses logs of students' actions within the ASSISTments Tutor to update the features of the students, problems, and content. A complete list and descriptions of all the context calculated by the APLS can be found at https://osf.io/zuwf7. The subset of this context used during the empirical study in this work is discussed later in Section 4.1.

3 DYNAMIC LINEAR EPSILON-GREEDY

This work presents the Dynamic Linear ϵ-Greedy (DLEG) algorithm, shown in Algorithm 1. DLEG is a contextual multi-armed bandit algorithm that addresses the exploration-exploitation trade-off in a way that enables statistically reliable, generalizable insight to be gleaned from the data collected during its use. DLEG uses a modification of the ϵ-greedy method, because the data collected from random decisions is akin to data collected during a randomized experiment, and is thus unbiased, and available for use in common statistical analyses.

DLEG estimates the reward from context using a ridge regression, similarly to other linear CMABs [1, 11]. After a short period of random recommendations used to give the regression initial data to fit on, with probability ϵ, DLEG will randomly select from the possible actions it can take, observe a reward, and then update the ridge regression with this sample. After updating the ridge regression, the regression is used to estimate the reward of the random recommendation that was just made. The error in this estimate is used to track the mean squared error of the model's reward estimates for its random recommendations, mse_r.

After a short period of random recommendations used to give the regression initial data to fit on, with probability $1 - \epsilon$, DLEG will use the ridge regression to estimate the reward for each possible action, and then take the action with the highest estimated reward, i.e., the greedy action. DLEG observes the reward for this greedy recommendation, but *does not* update the ridge regression after a greedy recommendation. The error of the greedy recommendation's reward estimate is used to track the mean squared error of the model's reward estimates for its greedy recommendations, mse_g.

The data collected from DLEG's random recommendations are independent of each other, and therefore can be used to analyze the qualitative interactions in the data without inducing any bias from dependence between samples. However, if ϵ never changes, then once the ridge regression has learned all it can from the data, DLEG will be wasting opportunities to exploit these qualitative interactions by continuing to make random recommendations. To avoid this, ϵ is updated dynamically on Line 31 of Algorithm 1 based on mse_r and mse_g, as long as a small amount of data exists for the calculation of mse_r and mse_g. If these two mean squared errors are equal, it means that the regression is just as good at estimating the reward given context it was not trained on as it is given context it was trained on, which implies that the model has captured the underlying trends in the data. If this is the case, then the model will stop making random recommendations. On the other hand, the worse the model is at estimating the reward given context it was not trained on compared to context it was trained on, the higher ϵ will be, resulting in more random recommendations. This allows the model to improve its predictive accuracy by collecting more training data. This method is also robust to changes in the relationship between context and reward, because if the accuracy of the reward estimates for greedy recommendations was very high, but started getting worse, DLEG would begin to make more random recommendations and continue to fit the regression. This simple trick of adjusting ϵ based on the ratio of the standard errors allows this variant of the ϵ-greedy method to be competitive with more optimal methods, while allowing for unbiased statistical analysis on the random recommendations.

In Algorithm 1:

- λ is the L2 penalty of the ridge regression, used during the initialization of the regression.
- α is the number of random recommendations that must be made first before DLEG can begin to make greedy recommendations.
- ϵ is the probability that the model will make a random recommendation after α random recommendations.
- n_r and n_g track the number of random and greedy recommendations made by DLEG respectively.

- mse_r and mse_g track the mean squared error of the ridge regression's reward predictions for random and greedy recommendations respectively.
- $\mathbf{A}$ and $\mathbf{b}$ are the $X^\top X + \lambda I$ and $X^\top Y$ components of the ordinary least squares solution for ridge regressions: $\hat{\beta} = (X^\top X + \lambda I)^{-1} X^\top Y$ [28]. $\mathbf{A}$ and $\mathbf{b}$ can be updated iteratively as more samples are collected.
- $\hat{\boldsymbol{\beta}}$ is the vector of coefficients of the ridge regression, which can be calculated each time a prediction needs to be made from $\mathbf{A}$ and $\mathbf{b}$.

Algorithm 1 Dynamic Linear ϵ-Greedy

Inputs: $\lambda \in \mathbb{R}_+,\ \alpha \in \mathbb{N},\ \epsilon = 0.5,\ n_r = 0,\ n_g = 0,\ mse_r = 0,\ mse_g = 0$
$\mathbf{A} \leftarrow \lambda \mathbf{I}_d$ (d x d dimensional diagonal matrix where all values on the principle diagonal are λ)
$\mathbf{b} \leftarrow \mathbf{0}_{d \text{ x } 1}$ (d x 1 dimensional zero matrix)
$\hat{\boldsymbol{\beta}} \leftarrow \mathbf{A}^{-1}\mathbf{b}$
for t = 1, 2, 3, ..., T **do**
  $R \leftarrow \mathcal{U}(0, 1)$
  Observe features of state s and all actions $a \in A_t$: $\mathbf{x}_{t,s,a} \in \mathbb{R}^{1 \text{ x } d}$.
  for all $a \in A_t$ **do**
    if $R \leq \epsilon$ **or** $n_r < \alpha$ **then**
      $p_{t,s,a} \leftarrow \mathcal{U}(0, 1)$
    else
      $p_{t,s,a} \leftarrow \mathbf{x}_{t,s,a}\hat{\boldsymbol{\beta}}$
    end if
  end for
  Choose arm $a_t = \arg\max_{a \in A_t} p_{t,s,a}$ with ties broken arbitrarily.
  Observe reward $r_t \in \mathbb{R}$.
  if $R \leq \epsilon$ **or** $n_r < \alpha$ **then**
    $\mathbf{A} \leftarrow \mathbf{A} + \mathbf{x}_{t,s,a_t}^\top \mathbf{x}_{t,s,a_t}$
    $\mathbf{b} \leftarrow \mathbf{b} + \mathbf{x}_{t,s,a_t}^\top r_t$
    $\hat{\boldsymbol{\beta}} \leftarrow \mathbf{A}^{-1}\mathbf{b}$
  end if
  $e \leftarrow \mathbf{x}_{t,s,a_t}\hat{\boldsymbol{\beta}} - r_t$
  if $R \leq \epsilon$ **or** $n_r < \alpha$ **then**
    $n_r \leftarrow n_r + 1$
    $mse_r \leftarrow mse_r + \frac{e^2 - mse_r}{n_r}$
  else
    $n_g \leftarrow n_g + 1$
    $mse_g \leftarrow mse_g + \frac{e^2 - mse_g}{n_g}$
  end if
  if $n_r \geq \alpha$ **and** $n_g \geq \alpha$ **then**
    $\epsilon \leftarrow 1 - \sqrt{\frac{mse_r}{mse_g}}$
  end if
end for

3.1 Design Constraints

For DLEG to operate at-scale within the ASSISTments APLS, its model was required to 1) have a limited, fixed memory cost, i.e., DLEG could not grow in size over time, nor could it be too big to begin with, and 2) be able to train from one

sample at a time, i.e., not require the entire history of recommendations to fit the model. Some CMABs like LinUCB [11] can be trained from one sample at a time, but fit one model for each action the CMAB can take. Within ASSISTments, new content is constantly being added to the system. If DLEG created an additional model each time new content was added, the system would quickly run out of memory. Additionally, separate models for different actions prevents the insight learned about the effectiveness of an action from being transferable to other actions.

Some CMABs use more complicated models like random forests [7] or deep neural networks [22] to learn the relationship between context and reward, but these models not only take up a large memory cost due to their structure, but they must also be re-fit using previous data as new data is collected, making these methods unsuitable for use within the APLS.

In order for DLEG to fit within the imposed constraints, a single ridge regression predicting reward using the context of the students, problems, and available content as input was used as DLEG's model. The ridge regression in DLEG is very similar to the model used in LinUCB [11], but instead of fitting a separate regression for each action, one regression that includes context of the actions was fit. This single regression allows DLEG to identify transferable insight into opportunities to personalize content provided to students based on features of the students, problems, and content in ASSISTments.

4 METHODOLOGY

4.1 Feature Selection

4.1.1 Simulation Study. Before conducting an empirical study of DLEG using the ASSISTments Automatic Personalized Learning Service (APLS), a simulation study was done comparing DLEG to similar variants of existing CMABs. The simulation study was performed using the ASSISTments Student Support Dataset (SSD) [18]. This dataset contains samples from thousands of experiments in which students were randomized between different problem-level supports. The features used from the SSD were chosen to be as similar as possible to the features chosen for the empirical study. For students, the user_avg_correctness, user_avg_support_requested, and user_med_ln_first_response_time features were used. While these features are not calculated identically to the features in the APLS, they attempt to measure the same thing. The difference being that the features in the APLS are normalized versions of the features included in the SSD. For problems, the problem_avg_correctness, problem_avg_support_requested, problem_med_ln_first_response_time, problem_type_1, problem_subject_g, problem_subject_rp, problem_subject_ns, problem_subject_ee, problem_subject_f, and problem_subject_sp features were used. The problem_type_1 feature in the SSD is similar to the problem_type_choice feature in the APLS, which is an indication of whether the question is of any type that requires the user to choose from options, as opposed to problem_type_1, which is an indication of whether or not the question is a multiple-choice question. For the problem-level supports, the student_support_is_explanation, student_support_message_count, student_support_contains_image, and student_support_contains_video features were included. The student_support_is_explanation feature in the SSD is equivalent to the answer_given feature in the APLS. The SSD provides the next problem correctness for each sample, which the APLS uses as the CMAB reward, Therefore, the simulation also used this as the reward. A complete description of the features in the SSD is available through [18]. In total, 1 constant, i.e., the intercept, 17 features, and 52 interactions between features of the supports and features of the users and problems were included in DLEG's regression for the problem-level support simulation study.

4.1.2 Empirical Study. Prior to this work, no CMABs had been evaluated using ASSISTments' APLS. Prior research has shown the negative impact that including too many features in a CMAB has on the CMAB's ability to benefit users [12].

Therefore, for the study in this work, the CMAB used a smaller subset of the features available in the APLS, as well as the interactions between the features of the content and features of the student and problem. The interactions between features was a necessary inclusion because without interactions, the ridge regression used by DLEG to estimate reward would not be able to find opportunities for personalization. For students, the correctness, support_requested, and ln_first_response_time features were chosen. For problems, the correctness, support_requested, ln_first_response_time, type_choice, subject_g, subject_rp, subject_ns, subject_ee, subject_f, and subject_sp features were chosen. For the skill-level videos, only percent_likes, percent_dislikes, and percent_comments were included in the context provided to DLEG. The definitions for all the above features can be fount at https://osf.io/zuwf7/. In total, 1 constant, i.e., the intercept, 16 features, and 39 interactions between features were included in DLEG's regression for the skill-level video empirical study.

4.2 Study Design

4.2.1 Simulation Study Design. The simulation study was conducted identically to previous simulation studies done using medical and educational data from randomized studies [18, 21]. To simulate how effectively CMABs would have recommended support to students in the SSD, samples from the SSD were randomly selected with replacement using the following strategy [18].

(1) Initialize a CMAB.
(2) Randomly sample with replacement a single instance of a student receiving support from the SSD.
(3) Provide context from the sample to the CMAB algorithm for all possible supports the student could have received.
(4) Given this context, receive a support recommendation from the CMAB.
(5) If the support recommended by the CMAB matches the support that was actually given to the student, update the bandit algorithm using the next problem correctness value in the SSD, otherwise ignore the recommendation and go back to step 2.
(6) Repeat steps 2-5 to simulate the CMAB making a series of recommendations.

This study ran for 1,000,000 recommendations to observe the long-term effects of the different algorithms. In the simulation study, DLEG was compared to random selection, Linear Thomson Sampling [1], and Pooled-LinUCB, which is similar to LinUCB [11] but with only one regression that shares context across actions. These CMABs were selected for comparison because they are well established algorithms that meet the memory and time requirements of the ASSISTments APLS.

4.2.2 Empirical Study Design. Once the simulation study demonstrated the effectiveness of DLEG compared to existing CMAB algorithms, the next step was to evaluate DLEG in a real setting, at-scale, within an online learning platform. Both a random selection model and a DLEG model were created in the APLS for recommending skill-level videos. Each time a student requested a video, the student's request was randomly sent to either the random model or DLEG with equal probability. The random model randomly recommended one of the available videos with equal probability, and DLEG recommended a video using Algorithm 1. Essentially, this study is a randomized experiment between two conditions (Random vs. CMAB recommendations), and the random selection model performed a randomized experiment between the different videos.

4.3 Study Analysis

4.3.1 Recommendation Algorithm Comparison. To compare the different recommendation algorithms to each other in both the simulation study and the empirical study, a logistic regression was used to predict the reward given the following inputs:

(1) A constant.
(2) Three covariates: student, problem, and next-problem prior correctness.
(3) A binary feature for each model except random selection indicating if that model made this recommendation.
(4) The number of recommendations made thus far by the algorithm that made this recommendation.
(5) A feature for the interactions between each of Input 3's features and Input 4.

If any of Input 5's features were positive and statistically significant, then the corresponding algorithm out-performed random selection, because over time, the chance of receiving a high reward increased for that algorithm more than it did for random selection. Additionally, if any of Input 5's features were statistically significantly different from each other, then one non-random model out performed another. This analysis was used instead of just comparing the distribution of reward between the algorithms because the distribution of reward is not expected to be different at the beginning of the algorithms' use, when mostly random recommendations are being made. However, once the non-random models have learned something, the reward distributions should be different.

4.3.2 Identifying Effective Content. To determine if DLEG was capable of identifying any significant relationships between features of the videos and students' performance at-scale, a logistic regression was fit to estimate students' next-problem correctness using all the video features available in the APLS as well as covariates for student, problem, and next-problem prior correctness. To ensure there was no bias in the estimates due to dependence between samples, only the data from DLEG's random recommendations during the empirical study was used to fit the model. This model was also fit using data from the random selection model used during the study to see how much difference there was between what DLEG's random recommendations revealed and what a randomized experiment revealed. The p-values of the models' coefficients were corrected for multiple hypothesis testing using the Benjamini-Hochberg procedure [3].

It is important to note that a lack of bias from dependent samples does not mean that the results of this regression can be interpreted as causal relationships. To identify causal relationships in the data, all but one feature of the content provided to students would have needed to be controlled [9]. However, the skill-level videos came from publicly available YouTube videos. No efforts were made to control for different features across videos, nor to make sure each skill had a similar distribution of features in the videos available for it. As such, the coefficients of this regression can only be interpreted as correlations in the data. However, there is nothing preventing the use of DLEG in a causal setting, as long as the content is appropriate for causal inference.

4.3.3 Identifying Qualitative Interactions. The greatest value of DLEG is in its ability to identify opportunities to personalize students' learning. For these opportunities to exist, qualitative interactions must be present in the data. Using the data collected from DLEG's random recommendations, the same method used in [18] to identify statistically significant qualitative interactions between users and the content available to them was used. In order to identify generalizable interactions, students were binned into high and low knowledge groups based on whether or not they had a higher than average correctness feature in the APLS. Each video feature was also binned into above and below average groups. The regression $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 (x_1 \oplus x_2)$ was then fit, where x_1 is a binary variable for a binned video feature, x_2 is a binary variable for a student's binned prior correctness, and y is the student's next problem

correctness. Using this model, a qualitative interaction exists if β_3^2 is greater than β_1^2, which is derived with more detail in [18]. p-values for the statistical significance of these qualitative interactions were calculated using a bootstrapping approach in which a regression for each video feature was fit 10,000 times on subsets of equal size to the original data sampled from the original data with replacement. The distribution of $\beta_3^2 - \beta_1^2$ was used to perform a one-sample t-test to determine the p-value of the null hypothesis: $\beta_3^2 - \beta_1^2 \leq 0$. p-values were corrected for multiple hypothesis testing using the Benjamini-Hochberg procedure [3].

5 RESULTS

5.1 Simulation Study

Figure 1 shows the cumulative reward received by the three CMABs compared to random selection during the simulation. In Figure 1, the total reward received through random selection was subtracted from the total reward received by each algorithm after the same number of recommendations were made. The random selection line is a horizontal line at $y = 0$ because the cumulative reward received through random selection was subtracted from itself. By comparing each CMAB to random selection, we can see more clearly how each CMAB compares to selecting at random from the available content.

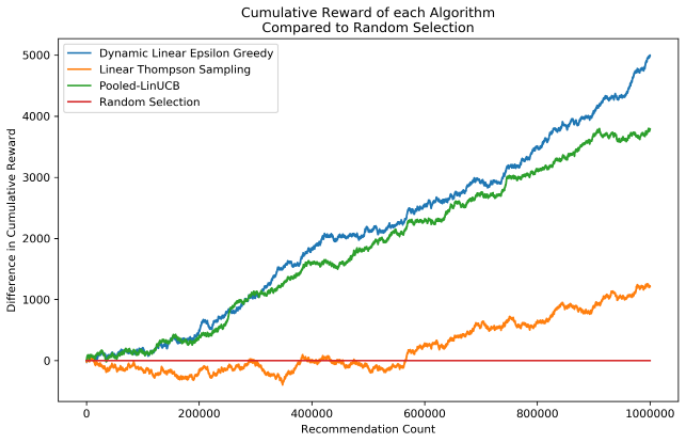


Fig. 1. The cumulative reward of each algorithm as a function of how many recommendations they have made compared to the cumulative reward received through random selection.

The regression described in Section 4.3.1 found that DLEG and Linear Thompson Sampling statistically significantly out-performed random selection ($p < 0.001$ and $p = 0.006$ after correction respectively), but Pooled-LinUCB did not. Although Figure 1 indicates that DLEG and Pooled-LinUCB are the best, this is not the case after adjusting for prior-knowledge covariates. Additionally, DLEG statistically significantly out-performed Pooled-LinUCB ($p = 0.012$ after correction). Based on the simulation, we could expect DLEG to perform better than random selection and at least as well as existing CMABs, while enabling further statistical analysis of the data.

5.2 Empirical DLEG Performance Analysis

From October 3rd 2022 to December 30th 2022, 3,602 students participated in the skill-level video empirical study. Each time a student requested support, they were randomized at a problem-level between receiving a randomly selected

video, chosen from 5 skill-related videos, or receiving the video recommended by DLEG from the same set of 5 videos. 6,035 total recommendations were made, 2,982 of them made by DLEG, and 3,053 of them made by random selection. 817 videos were shown to students across 217 skills. On average, when DLEG was used to make recommendations, about 2.8 different videos were shown per skill, and each video was viewed an average of 5.5 times. When random selection was used to make a recommendation, about 3.3 different videos were shown per skill, and each video was viewed around 4.5 times.

Figure 2 shows the trends in the recommendations made by DLEG. As shown by left graph, DLEG made fewer random recommendations over time, which indicates that it was able to learn the relationship between context and reward. The right graph shows that after an initial learning period, DLEG began to consistently out-perform random selection. Using the regression described in Section 4.3.1, no statistically significant differences between DLEG and random selection were found. With a longer study, it is likely that DLEG's video recommendations would have a statistically significant positive effect on students' propensity to get the next problem correct.

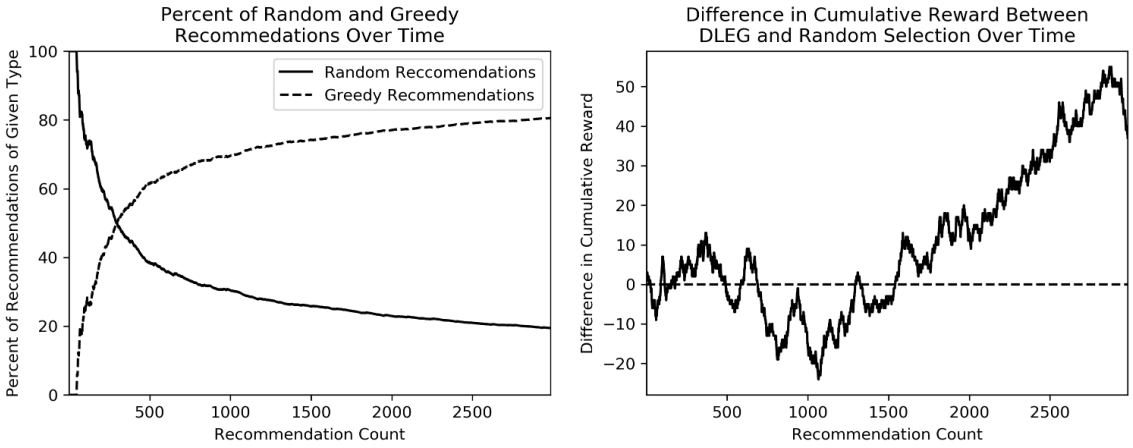


Fig. 2. The total percent of random and greedy recommendations made by DLEG (left) and the cumulative reward received by DLEG compared to random selection (left) for the skill-level video empirical study.

Figure 3 shows how the state of DLEG changed as more recommendations were made during the study. The left graph shows how at the beginning of the study, the standard error of DLEG's predictions of the reward for the random recommendations was very low, this was because there were few recommendations made, and the random recommendation data was used to train the regression, which caused DLEG's regression to over-fit on the random recommendation data. Due to this over-fitting, the standard error of DLEG's predictions of the reward for the greedy recommendations was very high. This resulted in a high initial ϵ. This is ideal because a CMAB should explore more at the beginning of its use in order to learn the trends in the data.

As DLEG made more recommendations, the standard error of the random recommendation reward predictions climbed and the standard error of the greedy recommendation reward predictions fell. This is an indication that DLEG's regression was trending away from over-fitting. As a result of these shifts in standard error, ϵ decreased. This is preferred because as a CMAB learns more about the relationship between context and reward, it should explore less and make more exploitative choices. At the end of the study, DLEG was making random recommendations about 7% of the time.

One can observe that a sudden drop in reward around recommendation 3,000 caused ϵ to slightly increase. This is desired because as trends in the data change, DLEG should explore more to learn about these new trends.

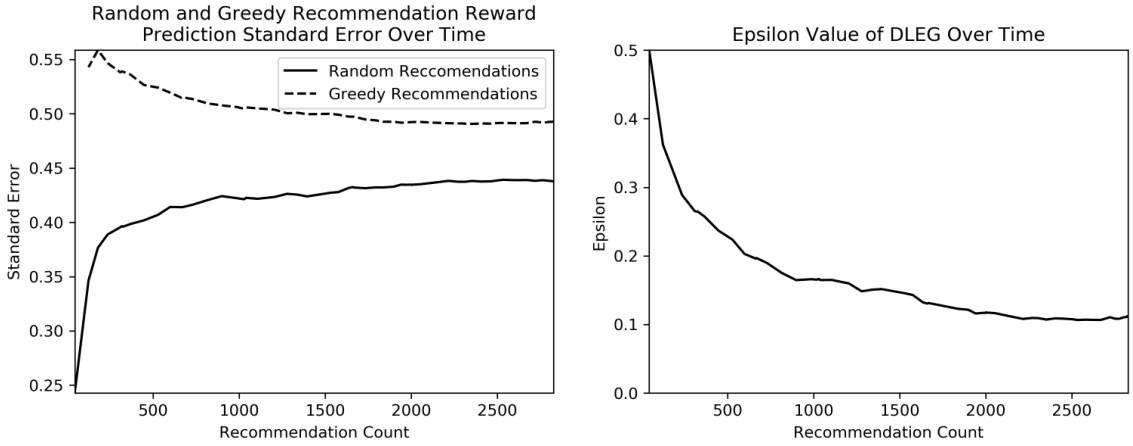


Fig. 3. The standard error of the predictions made for random and greedy recommendations (left) and the resulting ϵ (right) calculated by DLEG for the skill-level video study.

5.3 Empirically Identifying Effective Content

The purpose of using DLEG was not only to positively impact students' learning, but to also reveal statistically reliable relationships between features of the context and reward. Using the methodology discussed in Section 4.3.2, two logistic regressions were fit. One using DLEG's random recommendations and the other using the random selection algorithm's recommendations. The confidence intervals of the coefficients of the logistic regression fit using data from DLEG were about 43% larger on average than the confidence intervals of the coefficients of the logistic regressions fit using data from the random selection algorithm. The difference in confidence intervals is likely due to DLEG only making random recommendations about 20% of the time. However, even though the confidence intervals were larger, neither regression had any statistically significant coefficients, meaning that the lack of data did not result in DLEG missing any significant correlations.

5.4 Empirically Identifying Qualitative Interactions

Even though there were no features of the videos that were statistically significantly predictive of student performance, there could still be opportunities for personalization. The coefficients in the previous regressions only indicated how predictive each feature was of student performance on average, but it could be that higher knowledge students benefited from different things than lower knowledge students. To investigate for these qualitative interactions, the approach discussed in Section 4.3.3 was used to determine if there was a qualitative interaction between each feature of the videos and students' prior knowledge. The results of this analysis revealed 5 statistically significant qualitative interactions, shown in Table 1, all of which had p-values < 0.001 after correcting for multiple hypotheses.

These results indicate that despite little evidence that features of the content were predictive of students' average performance, many features had qualitative interactions with students' prior knowledge. These interactions can be used

Table 1. Some Typical Commands

Feature	High Knowledge Benefits From	Lower Knowledge Benefits From
Percent Likes	Above Average	Below Average
Length	Above Average	Below Average
Face Included	Yes	No
Reading Tone	Below Average	Above Average
Male Tone	Below Average	Above Average

to personalize the videos provided to students to help each student achieve their maximum potential. This analysis is possible because of the independent random recommendations that DLEG made during the study, without which the coefficients and confidence intervals of these analyses would be biased.

6 DISCUSSION

Using DLEG to recommend content to students had promising results. DLEG performed slightly, but significantly better than random selection and another CMAB when recommending problem-level supports in simulation, and also slightly out-performed random selection at-scale within ASSISTments, though not significantly.

Overall it seems the CMABs explored in this work struggled to have a large benefit on students' learning. Most likely, this lack of significant improvement was caused by the constraints placed upon the algorithms. DLEG, Pooled-LinUCB, and Linear Thompson Sampling all used a single ridge regression to model the relationship between context and reward. Many models used in learning sciences to understand students' performance do not reduce all content to a set of features, and instead model students or problems as individuals [5, 6, 17]. Additionally, the relationship between context and reward changes over time. In the skill-level video study, after an initial learning period, DLEG appeared to steadily out-perform random selection, but near the end of the study, DLEG's performance dropped. Around this time, students were preparing for winter break, and may have felt rushed to finish their work. This could have changed what kind of videos were most effective. Perhaps longer videos, which were previously more informative, were now ignored because students were unwilling to spend time watching them. This is just one hypothetical example of a change in students' preferences over time, but any number of factors could have led to this shift. DLEG will eventually re-learn trends, but if the trends in the data are often shifting, temporal features should be included in the model so a CMAB can learn to anticipate these trends. Lastly, it could be that DLEG had a difficult time significantly out-performing random selection because all the content in ASSISTments was equally good. Even if there were slight differences in quality, all the content was written or validated by mathematics teachers. In domains where there are fewer consequences for low-quality material, DLEG would likely have a larger benefit. However, in education, there is a significant negative impact when students are shown low-quality material. Therefore, all the content DLEG could recommend was likely similarly high-quality.

Although DLEG had only a small benefit to students, its purpose was not solely to benefit students, but to also glean statistically reliable and unbiased insight into the relationship between the context and the reward. Although there were no features of educational videos that were significantly predictive of students' average performance, multiple qualitative interactions between students' prior knowledge and features of the videos were significantly predictive of students' performance. Although these are only correlations, we can look at the interactions, theorize why they

occurred, and see if there are causal studies to support our theories. For example, this work found that higher knowledge students benefited more from videos that were above average in length. Studies have shown that students' attention span is a key factor in their academic success [14]. Therefore, it could be that students' attention spans help them to both achieve more academically and watch longer videos.

6.1 Limitations and Future Work

While the results of this study promising, there are some limitations to the scope of our analysis. Currently, DLEG has only been tested on data from the ASSISTments platform. While this has provided the opportunity to evaluate the effectiveness of DLEG at scale in a real-world environment, it also put strong restrictions on the memory and time requirements of DLEG. A version of DLEG more akin to LinUCB, where each action has a separate regression, could be even more powerful while still allowing for some statistically reliable insight. In the future, exploring how DLEG performs in other domains could both reveal interesting insight into the relationship between the context and reward of those new domains, as well as provide further opportunities to refine DLEG.

In addition to the limited scope, the empirical study ran for only about three months, and DLEG was only able to make 6,035 recommendations. While this may seem like a lot, many CMAB studies allow the algorithm to make millions of recommendations before interpreting the results. The limited time available for this study likely impacted the discovery of more significant results. Longer versions of this study should be repeated both to gather more data, and evaluate if the results are consistent.

7 CONCLUSION

In this work, we introduced DLEG, a CMAB algorithm that enables personalized content recommendations by learning and leveraging the statistical relationships between context and reward. We demonstrated through simulation and empirical studies that DLEG can slightly improve student performance within an online learning platform. Additionally, we found that unbiased random samples from DLEG's recommendations can reveal interesting qualitative interactions between content features and students' prior knowledge. These results have implications for both DLEG's ability to enhance student performance and for researchers seeking to design further studies or build upon existing pedagogy. In any domain where reliable, generalizable insights from recommendations are desired, DLEG can be employed to identify opportunities for personalization that benefit both researchers and recipients.

ACKNOWLEDGMENTS

We would like to thank NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, & R305A120125), GAANN (e.g., P200A180088 & P200A150306), EIR (U411B190024 & S411B210024), ONR (N00014-18-1-2768), NHI (R44GM146483), and Schmidt Futures. None of the opinions expressed here are that of the funders.

REFERENCES

[1] Shipra Agrawal and Navin Goyal. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*. PMLR, 127–135.

[2] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. 2009. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science* 410, 19 (2009), 1876–1902.

[3] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57, 1 (1995), 289–300.

[4] Djallel Bouneffouf, Irina Rish, and Charu Aggarwal. 2020. Survey on applications of multi-armed and contextual bandits. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

[5] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.

[6] Susan E Embretson and Steven P Reise. 2013. *Item response theory*. Psychology Press.

[7] Raphaël Féraud, Robin Allesiardo, Tanguy Urvoy, and Fabrice Clérot. 2016. Random forest for the contextual bandit problem. In *Artificial intelligence and statistics*. PMLR, 93–101.

[8] Neil T Heffernan and Cristina Lindquist Heffernan. 2014. The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education* 24, 4 (2014), 470–497.

[9] Guido W Imbens and Donald B Rubin. 2010. Rubin causal model. In *Microeconometrics*. Springer, 229–241.

[10] Tze Leung Lai, Herbert Robbins, et al. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6, 1 (1985), 4–22.

[11] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.

[12] ZhaoBin Li, Luna Yee, Nathaniel Sauerberg, Irene Sakson, Joseph Jay Williams, and Anna N Rafferty. 2020. Getting Too Personal (ized): The Importance of Feature Choice in Online Adaptive Algorithms. *International Educational Data Mining Society* (2020).

[13] William McCallum. 2015. The common core state standards in mathematics. In *Selected regular lectures from the 12th international congress on mathematical education*. Springer, 547–560.

[14] Megan M McClelland, Alan C Acock, Andrea Piccinin, Sally Ann Rhea, and Michael C Stallings. 2013. Relations between preschool attention span-persistence and age 25 educational outcomes. *Early childhood research quarterly* 28, 2 (2013), 314–324.

[15] Kyndra V Middleton. 2020. The Longer-Term Impact of COVID-19 on K–12 Student Learning and Assessment. *Educational Measurement: Issues and Practice* (2020).

[16] Thanaporn Patikorn and Neil T Heffernan. 2020. Effectiveness of crowd-sourcing on-demand assistance from teachers in online learning platforms. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*. 115–124.

[17] Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. 2009. Performance Factors Analysis–A New Alternative to Knowledge Tracing. *Online Submission* (2009).

[18] Ethan Prihar, Aaron Haim, Adam Sales, and Neil Heffernan. 2022. Automatic Interpretable Personalized Learning. In *Proceedings of the Ninth ACM Conference on Learning@ Scale*. 1–11.

[19] Ethan Prihar, Thanaporn Patikorn, Anthony Botelho, Adam Sales, and Neil Heffernan. 2021. Toward Personalizing Students' Education with Crowdsourced Tutoring. In *Proceedings of the Eighth ACM Conference on Learning@ Scale*. 37–45.

[20] Ethan Prihar, Manaal Syed, Korinn Ostrow, Stacy Shaw, Adam Sales, and Neil Heffernan. 2022. Exploring Common Trends in Online Educational Experiments. In *Proceedings of the 15th International Conference on Educational Data Mining*. 27.

[21] Anna Rafferty, Huiji Ying, Joseph Williams, et al. 2019. Statistical consequences of using multi-armed bandits to conduct adaptive educational experiments. *Journal of Educational Data Mining* 11, 1 (2019), 47–79.

[22] Carlos Riquelme, George Tucker, and Jasper Snoek. 2018. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127* (2018).

[23] Herbert Robbins. 1952. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.* 58, 5 (1952), 527–535.

[24] Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. 2018. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning* 11, 1 (2018), 1–96.

[25] Alexander O Savi, Joseph Jay Williams, Gunter KJ Maris, and Han van der Maas. 2017. The role of A/B tests in the study of large-scale online learning. (2017).

[26] Adish Singla, Anna N Rafferty, Goran Radanovic, and Neil T Heffernan. 2021. Reinforcement learning for education: Opportunities and challenges. *arXiv preprint arXiv:2107.08828* (2021).

[27] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3-4 (1933), 285–294.

[28] Wessel N van Wieringen. 2015. Lecture notes on ridge regression. *arXiv preprint arXiv:1509.09169* (2015).

[29] Jiayu Yao, Emma Brunskill, Weiwei Pan, Susan Murphy, and Finale Doshi-Velez. 2021. Power Constrained Bandits. In *Machine Learning for Healthcare Conference*. PMLR, 209–259.

[30] Yang Zhi-Han, Shiyue Zhang, and Anna Rafferty. 2022. Adversarial bandits for drawing generalizable conclusions in non-adversarial experiments: an empirical study. In *Proceedings of the 15th International Conference on Educational Data Mining*. 353.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

References for Included Papers

1.1: Identifying Struggling Students by Comparing Online Tutor Clickstreams

- AIED 2021 Short Paper:
    - Prihar, E., Moore, A., & Heffernan, N. (2021, June). Identifying Struggling Students by Comparing Online Tutor Clickstreams. In Artificial Intelligence in Education (pp. 290-295). Springer, Cham.

1.2: The Effects of Socioeconomic Status and Teachers' Pedagogy on Student Engagement During Remote Learning

- LAK 2021 Poster:
    - Prihar, E., Botelho, A., Yuen, J., Corace, M., Shanaj, A., Dia, Z., & Heffernan, N. (2021). Student Engagement During Remote Learning. In Companion Proceedings 11th International Conference on Learning Analytics & Knowledge (LAK21) (pp. 49-51).

1.3: The Effect of an Intelligent Tutor on Performance on Specific Posttest Problems

- EDM 2021 Full Paper:
    - Sales, A., Prihar, E., Heffernan, N., & Pane, J. F. (2021, June). The Effect of an Intelligent Tutor on Performance on Specific Posttest Problems. In Proceedings of the 14th International Conference on Educational Data Mining (pp. 206-215). International Educational Data Mining Society.

1.4: Classifying Math Knowledge Components via Task-Adaptive Pre-Trained BERT

- AIED 2021 Full Paper:
    - Shen, J. T., Yamashita, M., Prihar, E., Heffernan, N., Wu, X., McGrew, S., & Lee, D. (2021, June). Classifying Math Knowledge Components via Task-Adaptive Pre-Trained BERT. In Artificial Intelligence in Education (pp. 408-419). Springer, Cham.

1.5: Deep Learning or Deep Ignorance? Comparing Untrained Recurrent Models in Educational Contexts

- AIED 2022 Full Paper:
    - Botelho, A., Prihar, E., & Heffernan, N. (2022, July). Deep Learning or Deep Ignorance? Comparing Untrained Recurrent Models in Educational Contexts. Artificial Intelligence in Education (pp. 281-293). Springer, Cham.

1.6: Using Auxiliary Data to Boost Precision in the Analysis of A/B Tests on an Online Educational Platform: New Data and New Results

- AIED 2022 Poster:
    - Sales, A., Prihar, E., Gagnon-Bartsch, J., Gurung, A., & Heffernan, N. T. (2022). More Powerful A/B Testing using Auxiliary Data and Deep Learning. In Artificial Intelligence in Education (pp. 524-527). Springer, Cham.

1.7: Effective Evaluation of Online Learning Interventions with Surrogate Measures

- Submitted to EDM 2023:
    - Prihar, E., Vanacore, K., Sales, A., & Heffernan, N. (2023, July). Effective Evaluation of Online Learning Interventions with Surrogate Measures. Submitted to the 16th International Conference on Educational Data Mining.

2.1: Toward Personalizing Students' Education with Crowdsourced Tutoring

- L@S 2021 Full Paper:
    - Prihar, E., Patikorn, T., Botelho, A., Sales, A., & Heffernan, N. (2021, June). Toward Personalizing Students' Education with Crowdsourced Tutoring. In Proceedings of the Eighth ACM Conference on Learning at Scale (pp. 37-45).

2.2: A Novel Algorithm for Aggregating Crowdsourced Opinions

- EDM 2021 Short Paper:
    - Prihar, E., & Heffernan, N. (2021, June). A Novel Algorithm For Aggregating Crowdsourced Opinions. In Educational Data Mining (pp. 547-552). International Educational Data Mining Society.

2.3: Exploring Common Trends in Online Educational Experiments

- EDM 2022 Full Paper:
    - Prihar, E., Syed, M., Ostrow, K., Shaw, S., Sales, A., & Heffernan, N. (2022, July). Exploring Common Trends in Online Educational Experiments. In Proceedings of the 15th International Conference on Educational Data Mining. International Educational Data Mining Society.

2.4: Comparing Different Approaches to Generating Mathematics Explanations Using Large Language Models

- Submitted to AIED 2023:
    - Prihar, E., Lee, M., Hopman, M., Kalai, A., Vempala, S., Wang, A., Wickline, G., & Heffernan, N. (2023, July). Comparing Different Approaches to Generating Mathematics Explanations Using Large Language Models. Submitted to Artificial Intelligence in Education.

3.1: Automatic Interpretable Personalized Learning

- L@S 2022 Full Paper:
    - Prihar, E., Haim, A., Sales, A., & Heffernan, N. (2022, June). Automatic Interpretable Personalized Learning. In Proceedings of the Ninth ACM Conference on Learning at Scale (pp. 1-11).

3.2: Investigating the Impact of Skill-Related Videos on Online Learning

- Submitted to L@S 2023:
    - Prihar, E., Haim, A., Shen, T., Sales, A., Lee, D., Wu, X., & Heffernan, N. (2023, July). Investigating the Impact of Skill-Related Videos on Online Learning. Submitted to the Tenth ACM Conference on Learning at Scale.

3.3: A Bandit you can Trust

- Submitted to UMAP 2023:
    - Prihar, E., Sales, A., & Heffernan, N. (2023, June). A Bandit you can Trust. Submitted to The 31st ACM Conference On User Modeling, Adaptation And Personalization.