Bandit Algorithms to Personalize Educational Chatbots*

William Cai
Stanford University

Josh Grossman
Stanford University

Zhiyuan (Jerry) Lin
Stanford University

Hao Sheng
Stanford University

Johnny Tian-Zheng Wei
University of Massachusetts Amherst

Joseph Jay Williams
University of Toronto

Sharad Goel
Stanford University

October 15, 2020

Abstract

To emulate the interactivity of in-person math instruction, we developed MathBot, a rule-based chatbot that explains math concepts, provides practice questions, and offers tailored feedback. We evaluated MathBot through three Amazon Mechanical Turk studies in which participants learned about arithmetic sequences. In the first study, we found that more than 40% of our participants indicated a preference for learning with MathBot over videos and written tutorials from Khan Academy. The second study measured learning gains, and found that MathBot produced comparable gains to Khan Academy videos and tutorials. We solicited feedback from users in these two studies to emulate a real-world development cycle, with some users finding the lesson too slow and others finding it too fast. We addressed these concerns in the third and main study by integrating a contextual bandit algorithm into MathBot to personalize the pace of the conversation, allowing the bandit to either insert extra practice problems or skip explanations. We randomized participants between two conditions in which actions were chosen uniformly at random (i.e., a randomized A/B experiment) or by the contextual bandit. We found that the bandit learned a similarly effective pedagogical policy to that learned by the randomized A/B experiment while incurring a lower cost of experimentation. Our findings suggest that personalized conversational agents are promising tools to complement existing online resources for math education, and that data-driven approaches such as contextual bandits are valuable tools for learning effective personalization.

*We thank Keith Shubeck, Carol Forsyth, Ben Nye, Weiwen Leung, Ro Replan, and Sam Maldonado for helpful comments and discussions. This work was supported by the Office of Naval Research.

1 Introduction

Math learners can now turn to a wide variety of freely available online resources, from Khan Academy to Massive Open Online Courses (MOOCs). However, many of these resources cannot completely reproduce features of in-person tutoring, like giving students the sense that they are engaged in a back-and-forth exchange with a tutor, tailored feedback, and guidance about how to allocate their attention between reading explanations and practicing problems. Existing online math platforms have recently moved towards these desiderata with features like tailored feedback and personalized guidance. For example, online math homework tools like ASSISTments [15, 21] give feedback on common wrong answers. Further, online resources like MathTutor [3] build on example-tracing tutors [5], which model the progression of a lesson with a behavior graph that: (1) outlines potential student actions, such as providing common incorrect responses; and (2) specifies the feedback, explanation, or new problem that should follow those actions. That approach aims to reduce development time while achieving some of the benefits of intelligent tutoring systems for mathematics, like personalized selection of problems [12, 14, 31, 46].

One consequence of online math education shifting from static media to adaptive intelligent tutoring systems is the dramatic increase in potential for personalization of the platform. When developing interactive platforms, a content designer must choose an appropriate pedagogical strategy: for example, whether the topic should be conveyed through conceptual lessons or practice problems, and the degree of feedback that should be provided. To choose an optimal pedagogical strategy for every new piece of content, one could turn to cognitive and educational experts and draw from educational theory, such as aptitude treatment interaction [41]. In practice, however, it can be difficult to operationalize such theories to create effective strategies [49]. Furthermore, the large number of avenues for personalization, some of which may not have been previously investigated in the literature, along with the data available in online platforms, suggests a more computational approach for learning personalized pedagogical policies.

The traditional method to compare the efficacy of various policies is to run a randomized A/B experiment. However, running such an experiment may not be feasible or desirable in adaptive education platforms due to high exploration costs: many users may be assigned to a bad pedagogical policy before the experiment is over, leading to deleterious effects on their learning experience. An alternative to traditional randomized experiments is the contextual bandit, a popular technique from the reinforcement learning (RL) literature [27]. Compared to traditional A/B tests, bandit algorithms can often learn personalized strategies with substantially less experimentation, leading to improved user experiences.

Our paper builds upon the aforementioned intelligent tutoring systems, moving from adaptive platforms to an actual conversational interface that closely mimics some key facets of conversation with a human tutor. Specifically, we designed and evaluated a prototype chatbot system, which we call MathBot. To achieve conversational flow and mirror the experience of interacting with a human tutor, we paid close attention to the timing of prompts and incorporated informal language and emoji. As with a human tutor, the MathBot system alternates between presenting material and gauging comprehension. MathBot also provides learners with personalized feedback and guidance in the form of explanations, hints, and clarifying sub-problems. Finally, we built into MathBot the capability of learning personalized pedagogical policies via both contextual bandits and randomized experiments, allowing us to compare the two strategies in a live deployment.

To evaluate MathBot, we carried out three user studies on Amazon Mechanical Turk. The first study sought to determine whether users preferred to use MathBot over comparable online resources and, through qualitative feedback from users, elucidate potential avenues for improving MathBot through personalization. Specifically, 116 participants completed (in a randomized order)

both an abridged lesson about arithmetic sequences with MathBot and a video on Khan Academy covering similar content; these participants then rated their experiences. We found that 42% of users preferred learning with MathBot over the video, with 20% indicating a strong preference. An additional 110 participants completed the same abridged lesson with MathBot along with a written tutorial from Khan Academy containing embedded practice problems. In this case, 47% of these users preferred learning with MathBot over the written tutorial, with 18% stating a strong preference. While MathBot was not preferred by the majority of our participants, our results point to potential demand for conversational agents among a substantial fraction of learners.

The second study sought to determine whether MathBot produced learning gains on par with comparable online resources. We randomized 369 participants to either complete a full-length conversation with MathBot about arithmetic sequences or complete a set of videos and written tutorials from Khan Academy covering similar content. To test their knowledge, each subject took an identical quiz before and after completing their assigned learning module. Under both conditions, participants exhibited comparable average learning gains and learning times: 65% improvement for MathBot, with a mean learning time of 28 minutes (SD = 20), and 60% improvement given Khan Academy material, with a mean learning time of 29 minutes (SD = 22); we note that the difference in learning gain was not statistically significant.

Given that a subset of users indeed preferred MathBot to conventional learning tools, we explored the potential of contextual bandits to learn personalized pedagogical policies for MathBot in the third and main study. For this experiment, we recruited 405 participants to complete a full-length conversation with MathBot about arithmetic sequences. Unlike the first two studies, in which the possible conversation paths were the same for each user, the third study leveraged a version of MathBot that could choose, for each user, whether or not to present certain conceptual lessons and whether or not to provide certain supplemental practice questions. We randomized participants between two experimentation strategies—one in which actions were chosen by a contextual bandit and another in which actions were randomly chosen (i.e., an A/B design)—with the ultimate goal of reducing learning time without reducing learning gains. This goal was motivated by feedback from users in the first two studies, some of whom commented that the pacing of the lesson was too slow and some too fast, suggesting that personalizing the speed of the lesson could be beneficial. We found that, during experimentation, users assigned to the contextual bandit condition took less time (with a 95% confidence interval of [32, 266] seconds) to complete the lesson and were less likely to drop out, despite scoring equivalently on the post-learning assessment as those assigned to the A/B design condition. Finally, we compared the quality of the learned post-experimentation policies using offline policy evaluation techniques, finding no statistically significant difference between the quality of the policies learned by the contextual bandit and the randomized experiment.

In summary, our contributions are threefold: (1) MathBot, a prototype system that adds conversational interaction to learning mathematics through solving problems and receiving explanations; (2) a live deployment of a contextual bandit in a conversational educational system; (3) evidence that a contextual bandit can continuously personalize an educational conversational agent at a lower cost than a traditional A/B design.

2 Related Work

We briefly review past work on building chatbots, conversational tutoring systems, example-tracing tutors, and other intelligent tutoring systems (ITSs). We also survey the use of reinforcement learning algorithms in these systems.

Chatbots. Chatbots have been widely applied to various domains, such as customer service [47], college management [7], and purchase recommendation [22]. One approach to building a chatbot is to construct rule-based input-to-output mappings [2, 48]. One can also embed chatbot dialogue into a higher-level structure [8] to keep track of the current state of the conversation, move fluidly between topics, and collect context for later use [10, 39, 44]. We envisioned MathBot as having an explicit, predefined goal of the conversation along with clear guidance and control of intermediate steps, so we took the approach of modeling the conversation as a finite-state machine [6, 34, 35], where user responses update the conversation state according to a preset transition graph.

Conversational Tutors in Education. Conversational tutors in education often build a complex dialogue, such as asking students to write qualitative explanations of concepts (e.g., *A battery is connected to a bulb by two wires. The bulb lights. Why?*) and initiating a discussion based on the responses [19]. AutoTutor and its derivatives [16, 20, 30, 43] arose from Graesser et al.'s investigation of human tutoring behaviors [18] and modeled the common approach of helping students improve their answers by way of a conversation. These systems rely on natural language processing (NLP) techniques, such as regular expressions, templates, semantic composition [43], LSA [20, 33], and other semantic analysis tools [17]. Nye et al. added conversational routines to the online mathematics ITS ALEKS by attaching mini-dialogues to individual problems but leaving navigation on the website [31]. MathBot aims to have the entire learning experience take place through a text conversation, giving the impression of a single tutor. More broadly, MathBot differs from past work on NLP-based conversational tutors in that it explores the possibility of reproducing part of the conversational experience without handling extensive open-ended dialogue, potentially reducing development time.

Intelligent Tutoring Systems and Example-Tracing Tutors. A wide range of intelligent tutoring systems in mathematics use precise models of students' mathematical knowledge and misunderstandings [3, 4, 32, 36, 42]. To reduce the time and expertise needed to build ITSs, some researchers have proposed example-tracing tutors [4, 5, 23]. Specifically, example-tracing tutors allow content designers to specify the feedback that should appear after students provide certain answers and then record those action-feedback pairs in a behavior graph [5]. Using the Cognitive Tutor Authoring Tools (CTAT), Aleven et al. built MathTutor, a suite of example-tracing tutors for teaching 6th, 7th, and 8th grade math [3, 4]. Our work draws on insights from example-tracing tutors in that we build a graph encoding rules that determine how MathBot responds to specific student answers, though our approach differs in that we display these responses in a conversational context.

Learning Pedagogical Strategies with Bandits. To allow MathBot to personalize elements during live deployment, we incorporate a contextual multi-armed bandit algorithm [24, 27], a tool from reinforcement learning for discovering which actions are effective in different situations (contexts). Other reinforcement learning approaches have been applied in education, typically for offline learning. Ruan et al. [37] increase student performance by combining adaptive question sequencing with an NLP-based conversational tutor for teaching factual knowledge, but use a combination of random selection and a probabilistic model of learners' knowledge of particular items to order questions. Lee et al. [26] develop a framework to learn personalized pedagogical policies for DragonBox Adaptive, a K–12 math puzzle platform, without the support of an expertly-designed cognitive model. Chi et al. [9] use another popular technique from RL, a Markov decision process model, to learn an effective pedagogical strategy for making micro-decisions, such as eliciting the next step of the problem versus revealing it, in an NLP-based ITS teaching college-level physics. Lan and Baraniuk [25] develop a

contextual bandit framework to assign students to an educational format and optimize performance on an immediate follow-up assessment, but evaluate the performance of the framework offline and do not personalize the actual lessons. A key difference between these studies and MathBot is that it is rare to use these strategies online in a live educational deployment. Only a handful of studies have begun to explore live deployments for sequencing problems [11, 38], and none that we are aware of does so to learn which actions to take in a conversation.

3 MathBot System Design & Development

MathBot allows users to learn math topics through conversation-style interaction, rather than simply browsing online resources like videos, written lessons, and problems. Below we give an illustrative example of a learner interacting with MathBot, describe MathBot's front-end of an interactive chat, and outline its back-end of a conversation graph which specifies a set of rules, such as how to progress through concepts and which actions to take based on user responses.

Sample Learner Interaction with MathBot

Suppose a student, Alice, wants to learn about arithmetic sequences by interacting with MathBot. To start the interaction, MathBot greets Alice and asks her to extend the basic sequence "2, 4, 6, 8 ...". Alice correctly answers "10", so MathBot provides positive feedback (e.g., "Good work! 🎖") and begins a conceptual explanation of recognizing patterns in sequences. MathBot asks Alice if she is ready to complete a question to check her understanding, and Alice responds affirmatively. Alice progresses successfully through a series of additional explanations and questions.

Following an explanation of common differences, Alice is asked a new question: "What's the common difference of 2, 8, 14, 20, ...?". Figure 1 displays the conversation rules that underlie Alice's current question. When asked the new question, Alice confuses the term "common difference" with "greatest common factor", a topic she recently reviewed, so she answers "2". MathBot recognizes that Alice has made a mistake and subsequently checks that she knows how to identify terms in a sequence and subtract them, a prerequisite task for finding the common difference (Figure 1, ii). Alice answers correctly, so MathBot begins to ask her a series of additional sub-questions to further clarify the concept of common differences (Figure 1, iii). Alice successfully completes these sub-questions, so MathBot directs her back to the original question. Alice remembers learning that the common difference is the difference between consecutive terms, though she mistakenly subtracts 8 from 2 and answers "I think it's -6". Rather than have Alice finish a redundant series of sub-questions, MathBot recognizes that Alice has made a common mistake, subsequently provides specific feedback to address that mistake, and then allows Alice to retry the original question (Figure 1, iv). Alice answers the original question correctly and proceeds to a new question on identifying decreasing arithmetic sequences (Figure 1, v).

MathBot's Front-End Chat and Back-End Conversation Graph

The front-end of MathBot is a text chat window between MathBot and the student (Figures 2a and 2b). Students type replies to MathBot to give answers to problems, providing responses like "I'm not sure". Students can freely scroll through the chat history to review explanations or questions.

Drawing inspiration from example-tracing tutors [4, 5, 23], the MathBot back-end consists of a conversation graph that specifies a set of if-then rules for how learner input (e.g., "I'm ready" or "The answer is 6") leads to MathBot's next action (e.g., give a new problem or provide feedback). In this rule-based system, the state of the conversation is represented as a finite state machine (FSM).

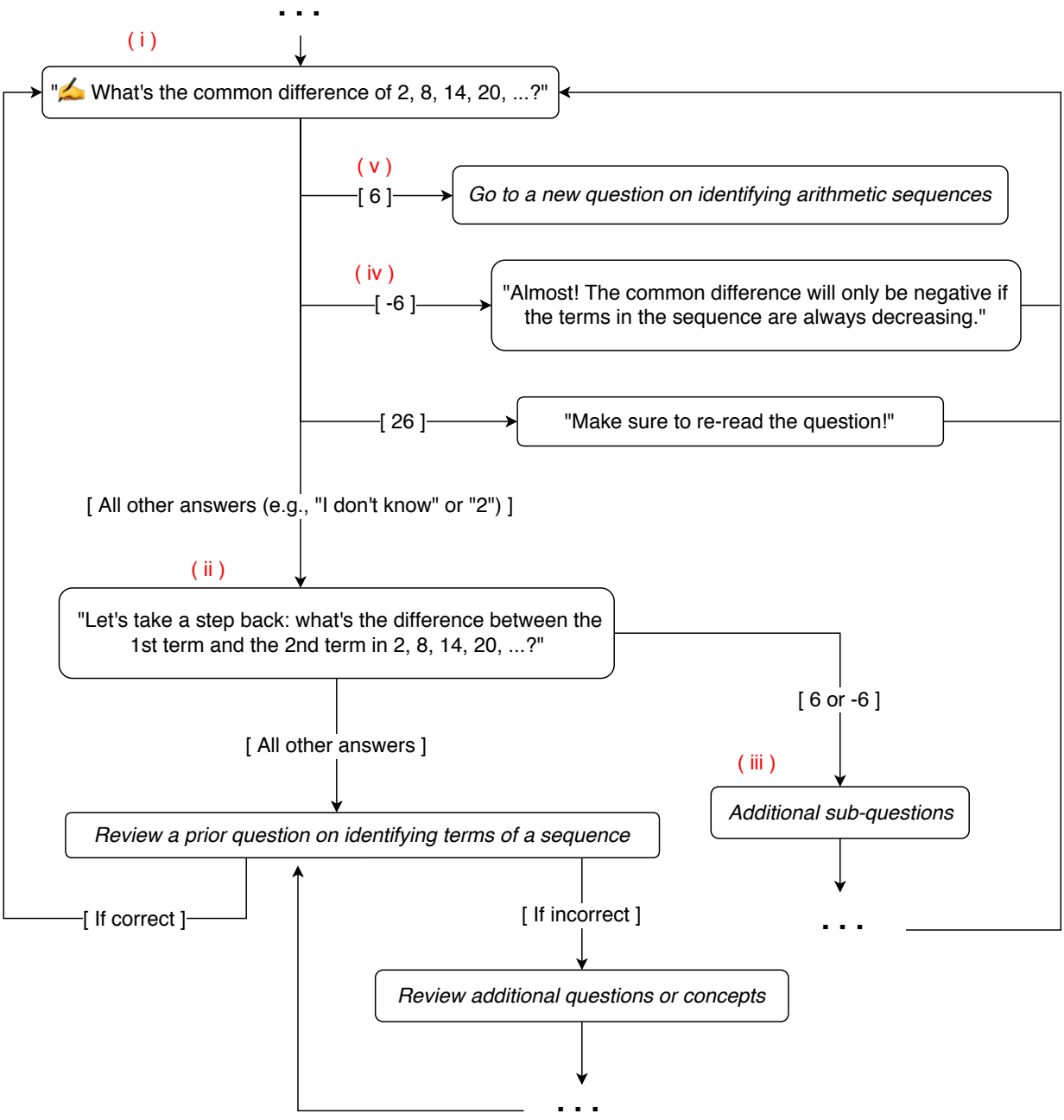
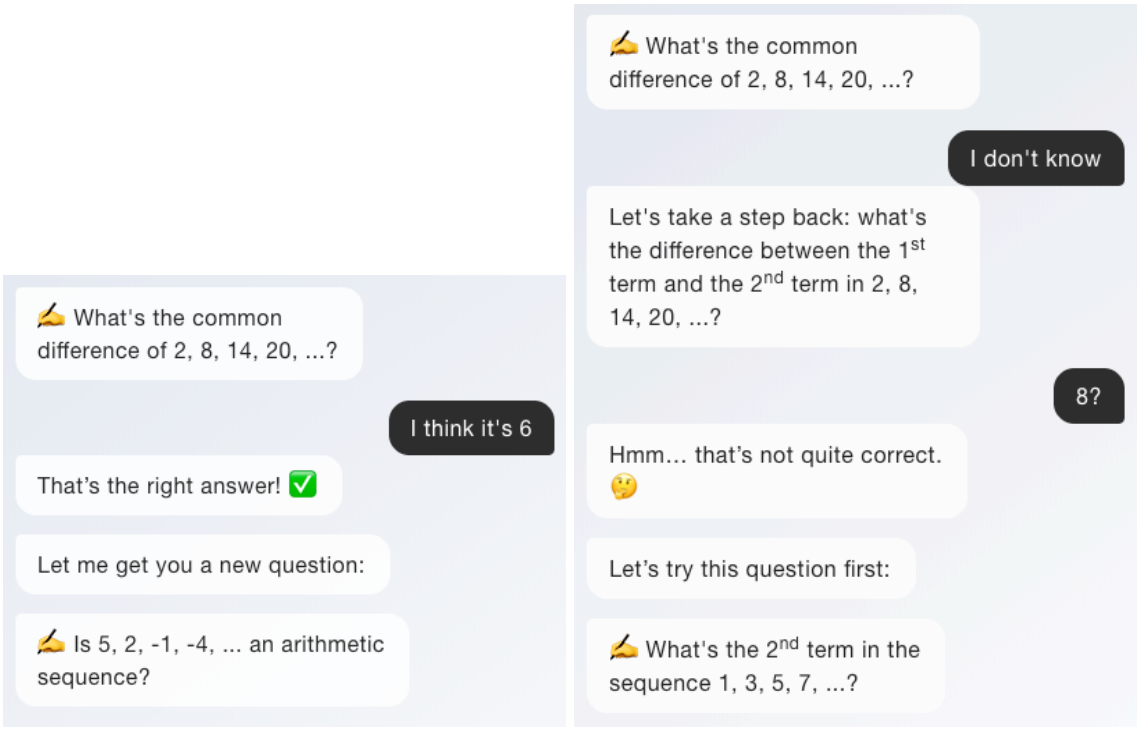


Figure 1: Example section of MathBot's conversation graph. Ellipses (...) denote excised sections of the full conversation graph. Marked blocks (i) – (v) denote actions taken by a hypothetical user, Alice, in Section 3.



(a) Correct response (b) Incorrect response

Figure 2: Example snippets of MathBot conversations.

In this FSM, each state is a response provided by MathBot, and user responses route the user along different paths in the conversation graph. For example, the question asked at the top of Figure 1 is a state, and responses to that question (e.g., "I don't know" or "6") route users to a new state. MathBot uses fuzzy matching and basic string equivalence to parse responses and route users appropriately.

4 Evaluating MathBot

We first validate MathBot in two studies comparing it to Khan Academy, a high-quality, free, and widely-used online resource for math tutorials and problems that delivers content in a non-conversational format. In the first study, we investigate user preferences between the two platforms and solicit qualitative feedback on what users liked and disliked about MathBot. In the second study, we compare the learning efficacy of the two platforms. In the third and main study, we leverage qualitative feedback from the first two studies to design personalized improvements to MathBot's pedagogical policy

Design of Study 1

In the first part of this within-subject study, we ask participants to interact with MathBot and watch a six-minute Khan Academy video, and then solicit feedback on the two learning methods (Figure 3). Despite their lack of interactivity, Khan Academy videos are competitive baselines, as they are carefully tailored by expert instructors and are demonstrably effective for teaching mathematical content [45].

We conduct the second part of the study identically, except we recruit new users and replace the

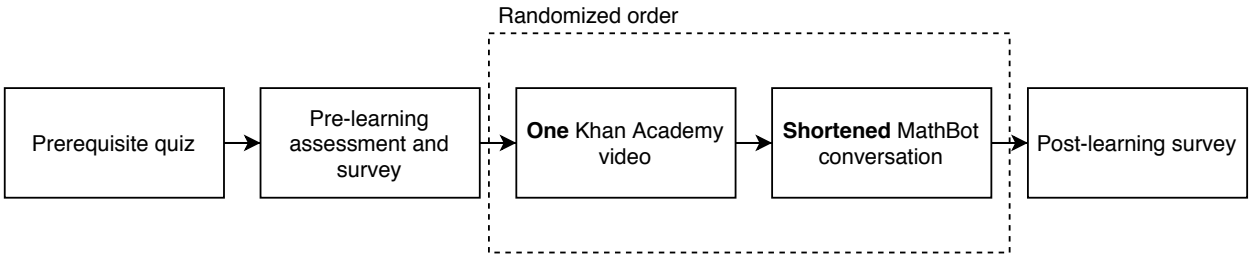


Figure 3: Study design of the first part of Study 1, which measured preferences for video-based instruction versus instruction via MathBot.

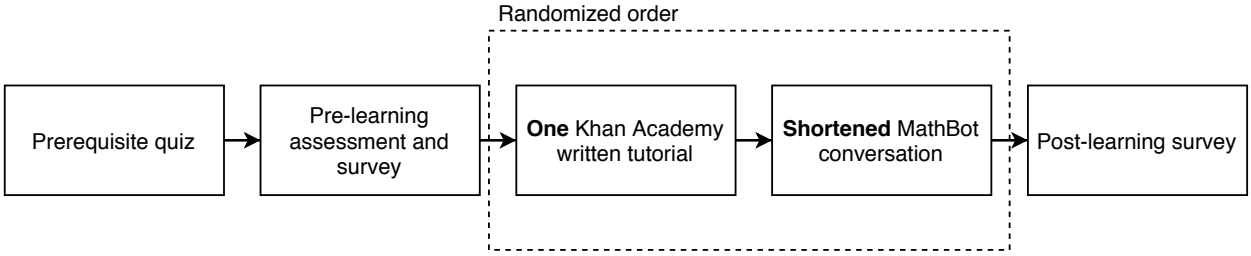


Figure 4: Study design of the second part of Study 1, which measured preferences for instruction via written tutorial versus instruction via MathBot.

video with a written tutorial from Khan Academy containing embedded practice problems (Figure 4). This second comparison provides an additional layer of insight, as one might conjecture that any result favoring MathBot over video instruction may simply be the result of MathBot providing an interface to work through problems.

To limit the length of the study, we use an abridged version of our developed MathBot content that covers only explicit formulas for arithmetic sequences, and pair that with either a Khan Academy video or a written tutorial that covers similar material. To avoid ordering effects—including anchoring bias and fatigue—we randomized the order in which participants saw MathBot and the Khan Academy video or written tutorial.

Our study was conducted on Amazon Mechanical Turk and was restricted to adults in the United States. To qualify for the study, we required that participants pass two screening quizzes. The first was a brief, 5-question quiz to ensure participants had sufficient algebra knowledge to understand sequences, but did not already have advanced knowledge of arithmetic sequences. The second screening quiz consisted of a more in-depth set of 12 questions selected from a Khan Academy quiz on arithmetic sequences. We excluded participants who answered more than 50% of the questions correctly, reasoning that these individuals already had substantial knowledge of sequences. Users were paid a bonus proportional to their score on a post-learning quiz. This performance-based payment scheme was disclosed to participants at the start of the study to incentivize active engagement with MathBot, attentive watching of the Khan Academy video, and dutiful completion of the written tutorial. Finally, we excluded participants who spent less than one minute on either MathBot or the Khan Academy learning module, reasoning that these individuals did not seriously engage with the material.

Figures 12 and 13 in the Appendix summarize user attrition and filtering, which were similar across conditions. After accounting for user attrition and the filtering criteria, 116 participants remained in the first part of the study and 111 participants in the second part. Figures 14 and 15 in

the Appendix summarize the demographics of the filtered set of users. Our analysis is restricted to this filtered set of users.

Quantitative Results

After study participants completed the MathBot and Khan Academy learning modules, we asked them a series of questions to quantify their experiences. In particular, we asked participants to answer the following question on a 7-point scale ranging from "strongly prefer" MathBot to "strongly prefer" the Khan Academy material: *"If you had 60 minutes to learn more about arithmetic sequences and then take a quiz for a large bonus payment, which of these two scenarios would you prefer? 1. Interact with an expanded version of the conversational computer program, then take the quiz. 2. [Watch more videos / Complete more interactive tutorials] about arithmetic sequences, then take the quiz."* We note that the ordering of options 1 and 2 was randomized for each user.

The responses to this question for the first part of the study are presented in Figure 5a. We found that 42% of participants stated at least a weak preference for MathBot, 53% stated at least a weak preference for Khan Academy videos, and 5% indicated a neutral preference. The corresponding results for the second part of the study are displayed in Figure 5b. In that case, we found that 47% of the 110 participants who answered the question stated at least a weak preference for MathBot, 44% stated at least a weak preference for Khan Academy interactive tutorials, and 9% stated a neutral preference. Figures 16 and 17 in the Appendix summarize the experiential ratings and time-on-task of participants in Study 1.

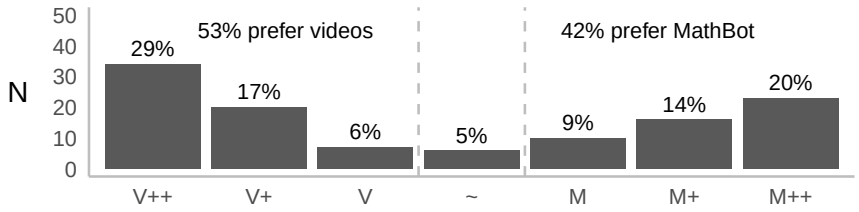
Overall, more of our participants preferred Khan Academy materials to MathBot—a testament to the quality of Khan Academy. The highly polarized response distribution, however, also illustrates the promise of new forms of instruction to address heterogeneous learning preferences. Indeed, 20% of users in the first part of the study and 18% of users in the second part expressed a "strong preference" for MathBot over Khan Academy material.

Qualitative Results

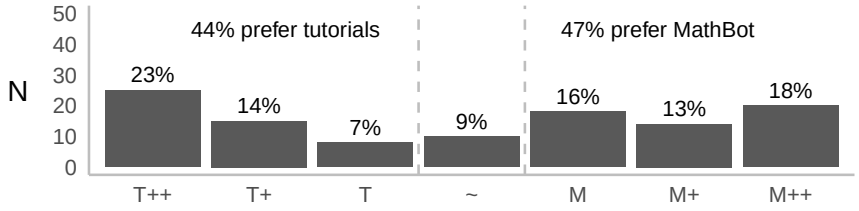
After each part of the study, we asked users to respond to the following prompt: *"Please compare your experience with the conversational computer program and the [video / interactive tutorial]. In what scenarios could one learning method be more effective or less effective than the other?"* We analyzed the resulting comments to identify themes and understand users' perspectives on MathBot and the Khan Academy videos and written tutorials. One author conducted open coding to identify common themes addressed by each response. Another author verified the coded labels and resolved conflicts with discussion. We discuss the coded categories at length in the Appendix, but highlight one theme in particular, that of pacing, here. We found that different users expressed different sentiments about the pacing of the lessons. For example, one participant noted, *"as it gets more complicated, the lesson should slow down a bit,"* while another indicated, *"I felt like the teaching went too slow for me."* We return to this feedback later on, seeking to address it via personalization, slowing down or speeding up the conversation for each learner as appropriate.

Design of Study 2

We next sought to evaluate whether MathBot produced comparable learning outcomes to Khan Academy material. To assess educational gains, we randomly assigned participants to learn about arithmetic sequences via: (1) a full-length MathBot conversation; or (2) a combination of Khan Academy videos *and* written tutorials covering the same content as the MathBot conversation. We assessed learning outcomes with a 12-question quiz, giving the same quiz before and after each



(a) MathBot vs. Khan Academy Video



(b) MathBot vs. Khan Academy Written Tutorial

Figure 5: Distributions of user preferences among the participants of Study 1. "M" denotes MathBot, "V" denotes video, and "T" denotes tutorial. Each "+" indicates a stronger preference, and "∼" indicates a neutral choice. Preferences for MathBot and Khan Academy are highly polarized, suggesting that the needs of learners could be better met by offering both modes of instruction.

participant completed the learning module. Users assigned to Khan Academy had access to seven videos and four written tutorials with embedded practice problems, and they were informed that completing either the videos or the tutorials would sufficiently prepare them for the post-learning quiz. Users were incentivized to complete the learning module to the best of their ability with a bonus payment proportional to their performance on the post-learning quiz. Similar filtering criteria to Study 1 resulted in our analyzing 182 subjects assigned to MathBot and 187 assigned to Khan Academy materials. Figure 18 in the Appendix summarizes user attrition and filtering in Study 2, and Figure 19 summarizes user demographics.

Results

We start by computing the proportional learning gain (PLG) for each subject. To calculate PLG, we first determine the raw learning gain by subtracting the pre-learning quiz score from the post-learning quiz score. We divide this result by the maximum possible score increase, defined as the difference between the maximum possible post-learning score (12) and the user's pre-learning score. Figure 7 shows the distribution of the PLG. We find the average PLG for MathBot users is 65%, with a 95% confidence interval of [58%, 72%]; the corresponding average PLG for Khan Academy users is 60%, with a 95% confidence interval of [53%, 67%]. The gains from MathBot are slightly higher than those from Khan Academy, but the difference is not statistically significant (two-mean t-test, $p = 0.15$, 95% CI: [-2%, 12%]). MathBot and Khan Academy users spent comparable time completing the learning modules—28 minutes on average for MathBot (SD = 20) and 29 minutes for the Khan Academy videos and written tutorials (SD = 22). Figure 20 in the Appendix summarizes raw learning outcomes of participants in Study 2, and Figure 21 summarizes performance on individual questions

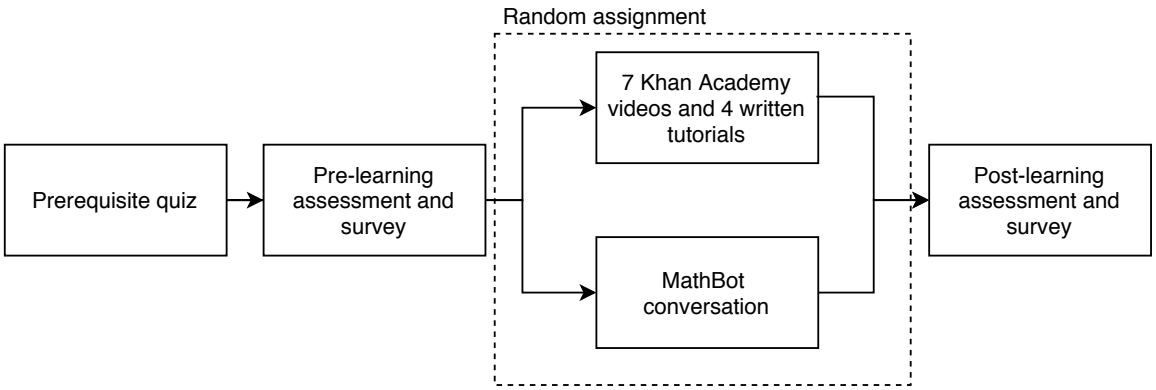


Figure 6: Experimental design of Study 2, which measured learning gains achieved by instruction via MathBot versus instruction via Khan Academy videos and written tutorials.

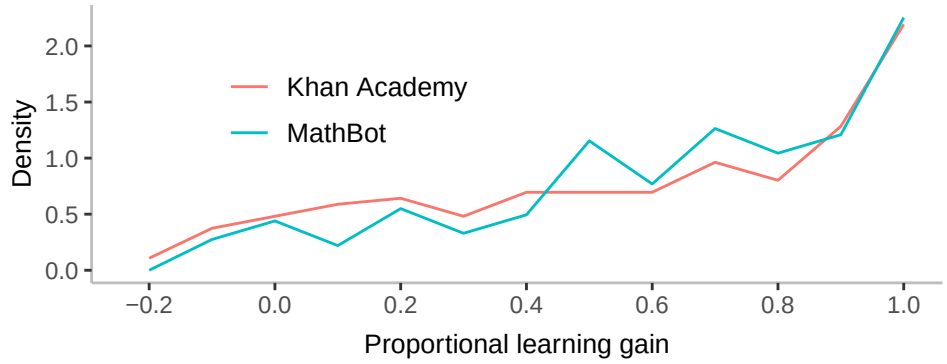


Figure 7: Distributions of proportional learning gain (PLG) for users of MathBot and Khan Academy in Study 2. The distributions are similar for users in both conditions.

in the pre- and post-learning assessments.

5 Learning a Pedagogical Policy

Here we return to feedback from users in Study 1 who expressed mixed sentiments about the pacing of MathBot and address their concerns by learning a personalized pedagogical policy for pacing. Given that the MathBot conversation is structured as a series of lessons, each consisting of a conceptual explanation followed by an assessment question, we could potentially adjust pacing of a lesson in one of four ways: (1) show the conceptual explanation and show an isomorphic practice question before the assessment question (slowest); (2) show the conceptual explanation but skip the isomorphic practice question; (3) skip the conceptual explanation but show the isomorphic practice question; and (4) skip the conceptual explanation and skip the isomorphic practice question (fastest). Figure 8 illustrates these four actions.

We took a data-driven approach to learning a personalized pedagogical strategy that selects between these four actions for each user and question. We specifically chose to use a *contextual bandit*, a tool from the reinforcement learning literature which balances exploring actions whose payoffs are unclear with exploiting actions whose payoffs are believed to be high [27]. For each user and question, the bandit selects one of the four above actions based on the user's pre-learning quiz score

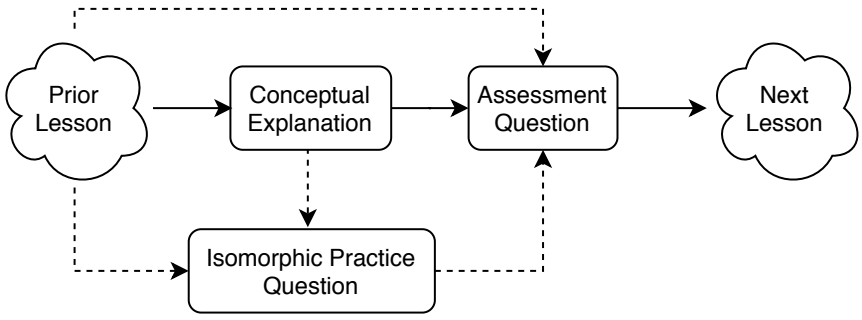


Figure 8: Potential actions taken by the contextual bandit before each assessment question. The bandit chooses whether or not to show a conceptual explanation and whether or not to show an isomorphic practice question.

(the *context*). For example, the algorithm might learn to speed up the conversation for users with high pre-learning quiz scores and slow it down for those with low scores. We note that we had access to many more contextual features than the pre-learning quiz score, such as scores on individual quiz items and self-reported academic history of study participants. However, to best mimic a real-life learning scenario where a tutor has access to only a coarse measure of prior knowledge, such as a grade in a prior course, we choose to use the pre-learning quiz score as the sole context.

To train a contextual bandit, we must not only specify the actions but also the objective function (the *reward*) over which the algorithm will optimize.[1] Recall that our motivation for improving our pedagogical strategy was to personalize the pacing of the lesson, with the goal of either slowing down the chat to boost comprehension or speeding it up without sacrificing learning. These dual desiderata suggest defining our reward as a linear combination of the total time spent on a lesson and an indicator of whether the user gets the assessment question correct on their first try:

$$150 \cdot \mathbf{1}_{\text{correct}} - \text{seconds spent on lesson.}$$

In other words, we assume it is worth 150 seconds of extra time spent on a lesson to turn a student who would have answered the assessment question incorrectly into a student who answered the question correctly. In particular, we expected the lesson to take around 30 minutes for 12 concepts, giving 2.5 minutes (or 150 seconds), for each concept. It bears emphasis that the precise form of the reward function should be set by domain experts and depends on the situation. For example, in a setting where a chatbot was augmented by a human tutor, we might increase the relative worth of time compared to correctness to account for the opportunity cost of having the concept explained by the tutor.

Design of Study 3

Our goal is to assess the value of using a contextual bandit to learn a personalized pedagogical strategy for students. We benchmark the bandit to a common alternative: a regression fit on data from users who were randomly assigned to one of the four possible actions before each assessment question. That is, in the benchmark approach, we first conduct an *exploration* phase, in which

[1]To train the bandit, we used a linear model with Thompson sampling, a technique known to have strong empirical performance and theoretical guarantees [1]. We model the reward using ordinary least squares (OLS) regression where the covariates are the contextual variables, the actions, and the two-way interaction terms between the contextual variables and the actions. Then, we simply choose action a with a probability proportional to its posterior likelihood of being the best action. See Appendix for more details.

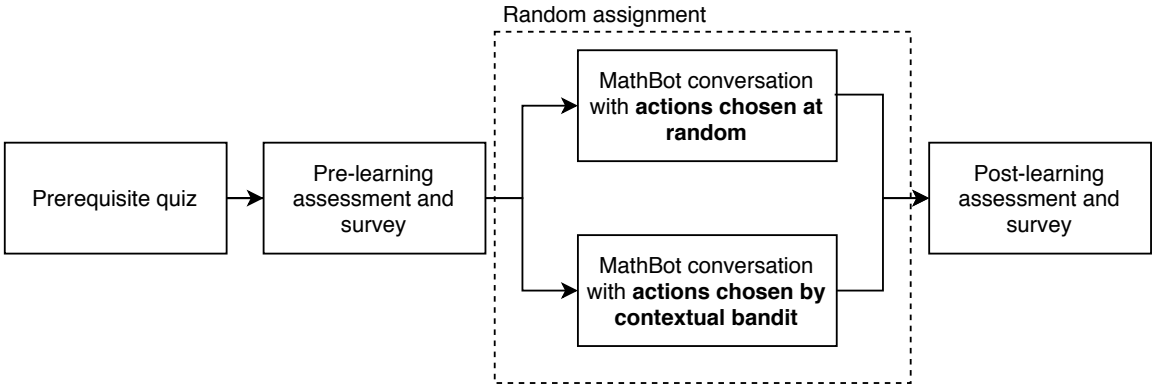


Figure 9: Experimental design of Study 3, which investigated whether a contextual bandit could learn a personalized pedagogical policy for MathBot at a lower cost than a randomized A/B design.

we assign users to the four actions uniformly at random; then, we fit a regression on the collected data to learn a personalized policy. The bandit, in contrast, aims to better manage exploration by down-weighting actions that are learned to be ineffective.

To carry out this comparison, we first recruited 30 participants from Amazon Mechanical Turk and assigned them to each of the four actions at random, independently for each question. Data from this pilot phase were used to provide the bandit a warm start. We then randomly assigned the remaining participants to either: (1) the contextual bandit condition; or (2) the uniform random condition.

We use the same criteria as in Study 2 to filter participants before they interact with MathBot. These filtering criteria resulted in 228 subjects assigned to MathBot with a uniform random policy and 239 assigned to MathBot with a contextual bandit policy. We note that both groups include the 30 participants from the pilot phase: they are included in the uniform random group since their actions were given uniformly at random, and they are included in the bandit group as the bandit learned its initial policy from those individuals. Figure 22 in the Appendix summarizes user dropout during experimentation.

Results

We examine the behavior of the contextual bandit algorithm along three dimensions: (1) its degree of personalization; (2) the quality of the final learned pedagogical policy; and (3) the cost of exploration. We found that the bandit learned a personalized policy comparable in quality to the one learned on the uniform random data but, importantly, did so while imposing less burden on users.

Personalization. We begin by examining the pedagogical policy ultimately learned by the contextual bandit (i.e., the policy the bandit believed to be the best at the end of the experiment, after seeing 239 participants). Averaged over all questions, the final, learned policy assigns approximately 30% of users to each of the concept-only, isomorph-only, and no-concept-no-isomorph conditions; the remaining 10% are assigned to the concept-plus-isomorph condition.[2] In Figure 10, we disaggregate the action distribution by question, showing the result for 4 representative questions out of the 11 total. The plot shows that the bandit is indeed learning a policy that differs substantially across users and questions. For only 3 of the 11 questions does the bandit determine that it is best to use

[2]These numbers do not represent the distribution of actions actually assigned during the experiment, but rather the distribution of actions under the policy the bandit ultimately learned.

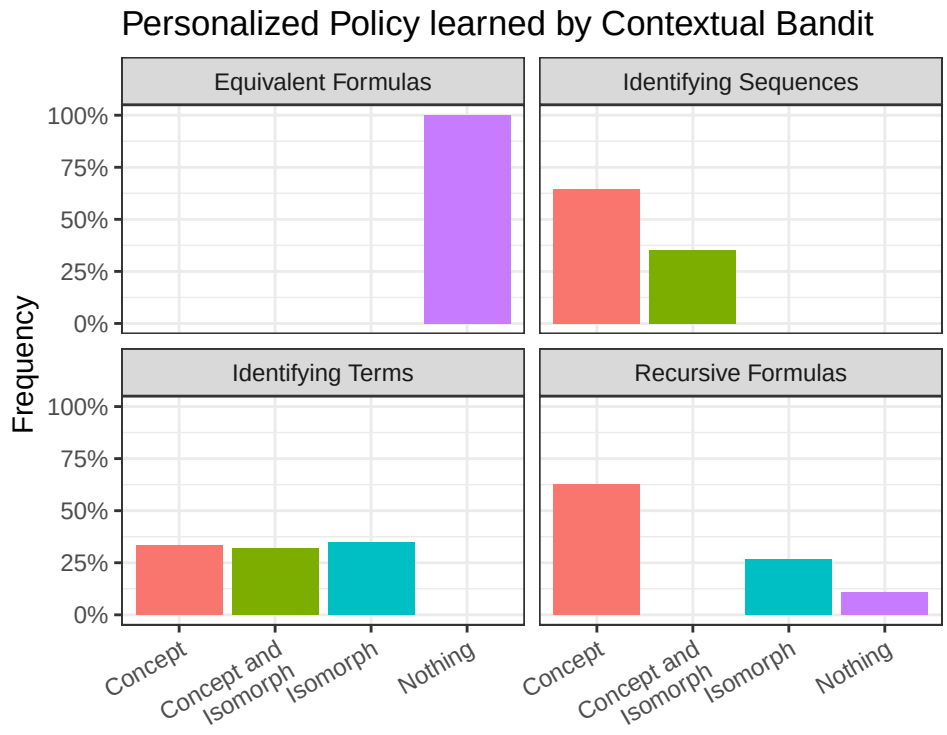


Figure 10: For four representative assessment questions out of the eleven total, the proportion of users for which the final policy learned by the bandit would use each action. The policy chooses different actions based on each user's pre-learning quiz score.

the same action for every user—though even in these cases, each of the 3 questions have different selected actions.

Quality of learned solution. Next, we compare the expected reward of the learned policy from the bandit to that of the learned policy from the uniform random condition.[3] For the uniform random condition, we consider three different regression models: (1) a model with two-way interactions between actions and questions (effectively learning a constant policy per question); (2) a model with the same specification as the bandit, which is able to personalize based on pre-learning quiz score; and (3) a lasso regression that includes eight contextual covariates: pre-learning quiz score, accuracy on the previous question, time since starting the learning session, whether in the previous concept they were shown the conceptual explanation and/or isomorphic practice question, the time they spent on the previous concept, and the speed at which they set MathBot to send responses. Of these three models, the first performs the best.

We find that the bandit learned a policy which is comparable to the most successful policy

[3]We never observe the actual outcomes of implementing these policies, as that would require running another costly experiment. We instead make use of standard offline policy evaluation techniques to compare the pedagogical strategies learned by the bandit and the uniform random experiment [28]. The specific quantity we are interested in is the expected average reward for a random user drawn from the population distribution. To compute the expected average reward for this policy on question i, we evaluate the average reward on question i in our uniform random condition where the user was randomly assigned into the action our bandit policy would have chosen. We then average these rewards over all the questions and compute standard errors by bootstrapping the uniform random data. We perform a similar method for the policies trained on the uniform random data, except we choose an action for person p using a model trained on all the uniform random data except those of person p to avoid overfitting.

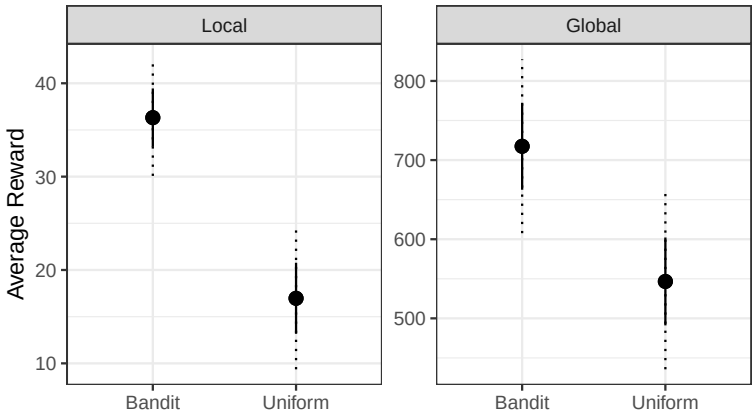


Figure 11: Average local (left) and global (right) rewards during the experiment for the bandit and uniform random conditions with 2 standard errors (1 SE solid, 2 SE dashed). In both cases, the contextual bandit obtains higher rewards, suggesting that it provides improved user experience while learning an optimal pedagogical policy.

learned from the uniform random condition, and further, that both the bandit and uniform random strategies learned a policy which outperformed the original policy from Studies 1 and 2 of always showing the concept without an isomorphic practice question. In Figure 24 in the Appendix, we display the average expected rewards of the two learned policies, along with the four policies which use the same action constantly. In particular, we find no statistically significant difference between the average reward obtained by the final bandit policy and the policy learned from the uniform random data. A 95% confidence interval for this difference in average rewards is [-11, 28], slightly in favor of the policy learned in the uniform random condition.

The cost of exploration. The above results indicate that one can indeed learn a personalized pedagogical policy using a contextual bandit that is on par with one learned from uniform random data. The primary value of a bandit, however, is that it incurs lower costs of exploration by quickly learning which actions are unlikely to be beneficial. We thus now directly compare the average rewards obtained under the bandit and uniform random conditions *during* the model-learning period. Higher average reward during model-learning suggests users are having a better experience, as they receive sub-optimal actions less often.

We first compute the average reward for each lesson in each condition, and then average that quantity over all the lessons for each condition. This gives us the the average reward per lesson per user for each condition. As shown in Figure 11 (left panel), the average reward in the contextual bandit condition is substantially higher than in the uniform random condition. A 95% confidence interval on the difference is [9.6, 29.1].[4]

As another way to assess the cost of exploration, we compute the average value of a *global* reward function across users in our two conditions—bandit and uniform random. Analogous to the local reward function, the global reward is defined as:

$$150 \cdot \textbf{Post-learning Quiz Score} - \text{seconds spent on MathBot.}$$

In contrast to the local reward function, the global reward considers the total post-learning quiz

[4]We compute the standard errors for our uniform random condition through the bootstrap. The standard errors for the bandit condition are obtained by fitting a response surface model on the uniform random data and running simulations.

score and total time spent on the entire MathBot conversation, rather than correctness and time spent during individual lessons.

Figure 11 (right panel) shows the average global rewards of participants between the two conditions. We find that the bandit obtains considerably higher average global rewards than the uniform random condition. A 95% confidence interval on the difference is [18, 324]. We note further that the difference is mostly driven by users in the bandit condition taking far less time to finish the MathBot conversation. Figure 23 in the Appendix breaks down the average learning gains and lesson times for users in the two conditions. A 95% confidence interval of the difference is [32, 266] seconds, while users in both conditions scored roughly the same on the post-learning quiz, with a 95% confidence interval on the difference being [-0.5, 0.9], slightly in favor of the bandit. The bandit was only designed to optimize for *local* rewards, so this result offers further evidence that the bandit is learning a *generally* effective policy.

As a final way to assess user satisfaction during exploration, we examine the difference in dropout rates between the two conditions. A user is said to "drop out" if they complete the first MathBot lesson but not the final lesson, either skipping to the post-learning quiz or leaving the experiment. Out of the participants in the bandit condition, 9% dropped out, compared to 15% of participants in the uniform random condition—a statistically significant gap of 6 percentage points (two-proportion z-test, $p < 0.05$). This result again suggests the bandit provides an improved user experience while learning a pedagogical policy.

6 Discussion

Limitations

One potential shortcoming of MathBot and similar conversational tutoring systems is the time needed to develop and test the underlying conversation graph. On the other hand, since it does not require researchers to develop NLP algorithms and models for conversation, it has one of the strengths of example-tracing tutors: those without extensive machine-learning expertise, including high-school instructors, could feasibly participate in development. In addition, Study 3 demonstrated the successful use of contextual bandit algorithms to learn how to personalize elements of the conversation graph—for example, when to skip conceptual explanations and when to give additional practice problems. This result provides one demonstration of how such components could be learned via a data-driven process after deployment, further minimizing the development time.

It is worth discussing whether our success in using a contextual bandit to learn a pedagogical policy might generalize to other learning scenarios. One of the main theoretical concerns of using a contextual bandit in learning scenarios is that it may not be able to optimally handle long-term dependencies (e.g., skipping the first conceptual explanation hurts performance on the eighth concept). Much work has thus explored more complicated strategies for learning personalized pedagogical strategies which require more data [9, 37]. In particular, we point out two features of our setting which are actually encouraging in this respect: (1) the lesson contains many concepts, most of which build upon one another, and (2) our bandit, despite being designed with a local reward function, was still able to learn more effectively than a uniform random policy even when evaluated with a global reward function. These two points of evidence suggest that bandits, despite theoretical concerns, may still have value in learning pedagogical policies even in complex and path-dependent learning scenarios.

An important limitation of our study is that we evaluated MathBot using a convenience sample of adults from Amazon Mechanical Turk. While Mechanical Turk workers have been shown to exhibit similar video-watching behavior and quiz performance as MOOC learners [13], it would be valuable

to test our system with a population actively exposed to algebra instruction, such as high school students or remedial adult learners in college. Our study also does not address the implications of using MathBot as a major component of a full-length course. For example, we did not investigate knowledge retention, and we do not know whether students would enjoy using MathBot less or more if they used it to learn over the course of several weeks or months. Since MathBot's applicability to a classroom setting is yet to be explored, future work can consider how this approach would be received and used by teachers. For example, would MathBot be most useful as homework, as an optional supplementary resource, or as in-class practice?

Additionally, our system taught a single algebra topic, arithmetic sequences, with a conversation intended to last approximately 30 minutes (Studies 2 and 3) and could be less than 10 minutes (Study 1). Furthermore, because Khan Academy is an independent platform, we were unable to deeply investigate video-watching and tutorial-completing behavior of participants in Studies 1 and 2. Further investigation is necessary to understand exactly which of our insights might generalize to other learning scenarios, including longer interaction periods, different topics in mathematics, and different learning formats such as games [26].

Conclusion

In this work, we developed and studied the effect of an interactive math tutoring system: MathBot. Although the content of MathBot closely matched that of the Khan Academy materials, we found evidence of heterogeneous learning preferences. MathBot produced learning gains that were somewhat higher than those of Khan Academy, though the gap was not statistically significant. Finally, we found that a contextual bandit was able to efficiently learn a personalized pedagogical policy for showing extra practice problems and skipping explanations to appropriately alter the pace of the MathBot conversation, outperforming a randomized experiment.

We note several directions for further work. We found that the bandit was able to learn as effective a policy as the randomized A/B experiment while requiring less time: however, we did not study what might happen in a setting where the time of the lesson was fixed and the bandit instead had to optimize learning gains given the fixed time allotted for the lesson. Additionally, given the challenge of fully exploring a substantial number of actions and a sizable context space with only a limited number of interactions with real users, the contextual bandit used in Study 3 had access to only four actions with one contextual variable. If a future iteration of MathBot were released to a larger audience, the bandit could explore additional actions, such as entirely skipping topics or providing more than one additional practice question, and could leverage additional contextual variables, such as users' stated preferences for learning via conceptual explanations versus example problems, or individual pre-quiz answers. Furthermore, the choice of learning media itself could be personalized with either a contextual bandit or another technique from the reinforcement learning literature: one could certainly imagine specific students or concepts being better suited for conversation than video or vice-versa.

Several users in Study 1 noted the benefit of interacting with multiple learning modules, and past work has demonstrated that prompting users with relevant questions periodically during a video may improve learning outcomes [40]. Accordingly, one could explore integrating brief conversations with MathBot into educational videos or, conversely, video elements could be used in the MathBot conversation. Though MathBot interactively guides learners through explanations and relevant questions, it does not provide a platform for extensive rote practice after finishing the conversation. An adaptive question sequencing model such as DASH [29] could be used to guide students through an optimized sequence of practice problems by accounting for student performance during the MathBot conversation. We hope that future work will investigate the potential of intelligent tutoring

systems that incorporate multiple modes of teaching and learn to personalize themselves to individual student needs.

References

[1] S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, 2013.

[2] R. Al-Rfou, M. Pickett, J. Snaider, Y.-h. Sung, B. Strope, and R. Kurzweil. Conversational contextual cues: The case of personalization and history for response ranking. *arXiv preprint arXiv:1606.00372*, 2016.

[3] V. Aleven, B. M. McLaren, and J. Sewall. Scaling up programming by demonstration for intelligent tutoring systems development: An open-access web site for middle school mathematics learning. *IEEE Transactions on Learning Technologies*, 2(2):64–78, 2009.

[4] V. Aleven, B. M. Mclaren, J. Sewall, and K. R. Koedinger. A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors. Technical report, 2009.

[5] V. Aleven, B. M. McLaren, J. Sewall, M. van Velsen, O. Popescu, S. Demi, M. Ringenberg, and K. R. Koedinger. Example-Tracing Tutors: Intelligent Tutor Development for Non-programmers. *International Journal of Artificial Intelligence in Education*, 26(1):224–269, Mar 2016.

[6] P. Andrews, M. De Boni, S. Manandhar, and M. De. Persuasive Argumentation in Human Computer Dialogue. In *AAAI Spring Symposium: Argumentation for Consumers of Healthcare*, pages 8–13, 2006.

[7] K. Bala, M. Kumar, S. Hulawale, and S. Pandita. Chat-bot for college management system using ai. *International Research Journal of Engineering and Technology*, 2017.

[8] D. G. Bobrow and T. Winograd. An overview of krl, a knowledge representation language. *Cognitive science*, 1(1):3–46, 1977.

[9] M. Chi, K. VanLehn, D. Litman, and P. Jordan. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. In *User Modeling and User-Adapted Interaction*, pages 137–180, 2011.

[10] J. Chu-Carroll and M. K. Brown. Tracking initiative in collaborative dialogue interactions. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 262–270. Association for Computational Linguistics, 1997.

[11] B. Clement, P.-Y. Oudeyer, D. Roy, and M. Lopes. Multi-armed bandits for intelligent tutoring systems. 7(2):20–48, 2015.

[12] S. D. Craig, X. Hu, A. C. Graesser, A. E. Bargagliotti, A. Sterbinsky, K. R. Cheney, and T. Okwumabua. The impact of a technology-based mathematics after-school program using ALEKS on student's knowledge and behaviors. *Computers & Education*, 68:495–504, Oct 2013.

[13] D. Davis, C. Hauff, and G.-J. Houben. Evaluating crowdworkers as a proxy for online learners in video-based learning contexts. In *Proceedings of the ACM on Human-Computer Interaction*, pages 42:1–42:16. ACM, 2018.

[14] J.-C. Falmagne, D. Albert, C. Doble, D. Eppstein, and X. Hu. *Knowledge spaces: Applications in education*. Springer Science & Business Media, 2013.

[15] M. Feng, N. Heffernan, and K. Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction*, 19(3):243–266, 2009.

[16] A. C. Graesser, S. Lu, G. T. Jackson, H. H. Mitchell, M. Ventura, A. Olney, and M. M. Louwerse. AutoTutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, 36(2):180–192, May 2004.

[17] A. C. Graesser, P. Penumatsa, M. Ventura, Z. Cai, and X. Hu. Using lsa in autotutor: Learning through mixed initiative dialogue in natural language. *Handbook of latent semantic analysis*, pages 243–262, 2007.

[18] A. C. Graesser, N. K. Person, and J. P. Magliano. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9(6):495–522, Dec 1995.

[19] A. C. Graesser, K. VanLehn, C. P. Rosé, P. W. Jordan, and D. Harter. Intelligent tutoring systems with conversational dialogue. *AI magazine*, 22(4):39–39, 2001.

[20] A. C. Graesser, K. Wiemer-Hastings, P. Wiemer-Hastings, and R. Kreuz. AutoTutor: A simulation of a human tutor. *Cognitive Systems Research*, 1(1):35–51, Dec 1999.

[21] 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.

[22] A. Horzyk, S. Magierski, and G. Miklaszewski. An intelligent internet shop-assistant recognizing a customer personality for improving man-machine interactions. *Recent Advances in intelligent information systems*, pages 13–26, 2009.

[23] K. R. Koedinger, V. Aleven, N. Heffernan, B. Mclaren, and M. Hockenberry. Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. Technical report, 2004.

[24] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[25] A. S. Lan and R. G. Baraniuk. A contextual bandits framework for personalized learning action selection. In *Proceedings of the 9th International Conference on Educational Data Mining*, pages 424–429, 2016.

[26] S. J. Lee, Y.-E. Liu, and Z. Popovic. Learning individual behavior in an educational game: A data-driven approach. *Proceedings of the 7th International Conference on Educational Data Mining (EDM)*, pages 114–121, 2014.

[27] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW 2010*, pages 661–670, 2010.

[28] L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *JMLR*, pages 19–36, 2012.

[29] R. V. L. Michael C. Mozer. Predicting and improving memory retention:psychological theory matters in the big data era. *Big Data in Cognitive Science*, pages 43–73, 2016.

[30] B. D. Nye, A. C. Graesser, and X. Hu. AutoTutor and Family: A Review of 17 Years of Natural Language Tutoring. *International Journal of Artificial Intelligence in Education*, 24(4):427–469, Dec 2014.

[31] B. D. Nye, P. I. Pavlik, A. Windsor, A. M. Olney, M. Hajeer, and X. Hu. SKOPE-IT (Shareable Knowledge Objects as Portable Intelligent Tutors): overlaying natural language tutoring on an adaptive learning system for mathematics. *International Journal of STEM Education*, 5(1):12, Dec 2018.

[32] E. O'Rourke, E. Andersen, S. Gulwani, and Z. Popović. A Framework for Automatically Generating Interactive Instructional Scaffolding. 2015.

[33] N. K. Person. AutoTutor improves deep learning of computer literacy: Is it the dialog or the talking head. *Artificial intelligence in education: Shaping the future of learning through intelligent technologies*, 97:47, 2003.

[34] S. Quarteroni and S. Manandhar. A chatbot-based interactive question answering system. *Decalog 2007*, page 83, 2007.

[35] A. Raux and M. Eskenazi. A finite-state turn-taking model for spoken dialog systems. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 629–637. Association for Computational Linguistics, 2009.

[36] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett. Cognitive Tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, 14(2):249–255, Apr 2007.

[37] S. Ruan, L. Jiang, J. Xu, B. J.-K. Tham, Z. Qiu, Y. Zhu, E. L. Murnane, E. Brunskill, and J. A. Landay. Quizbot: A dialogue-based adaptive learning system for factual knowledge. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

[38] A. Segal, Y. B. David, J. J. Williams, K. Gal, and Y. Shalom. Combining difficulty ranking with multi-armed bandits to sequence educational content. In *International Conference on Artificial Intelligence in Education*, pages 317–321, 2018.

[39] S. Seneff. Tina: A natural language system for spoken language applications. *Computational linguistics*, 18(1):61–86, 1992.

[40] H. Shin, E.-Y. Ko, J. J. Williams, and J. Kim. Understanding the effect of in-video prompting on learners and instructors. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 319. ACM, 2018.

[41] R. E. Snow. Aptitude-treatment interaction as a framework for research on individual differences in learning. In *A series of books in psychology. Learning and individual differences: Advances in theory and research*, pages 13–59. 1989.

[42] K. VanLehn. Conceptual and meta learning during coached problem solving. pages 29–47, 1996.

[43] K. VanLehn, P. W. Jordan, C. P. Rosé, D. Bhembe, M. Böttner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, et al. The architecture of why2-atlas: A coach for qualitative physics essay writing. In *International Conference on Intelligent Tutoring Systems*, pages 158–167. Springer, 2002.

[44] M. Walker and S. Whittaker. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, pages 70–78. Association for Computational Linguistics, 1990.

[45] B. Weeraratne and B. Chin. Can khan academy e-learning video tutorials improve mathematics achievement in sri lanka?. *International Journal of Education and Development using Information and Communication Technology*, 14(3):93–112, 2018.

[46] R. Winkler, S. Hobert, A. Salovaara, M. Söllner, and J. M. Leimeister. Sara, the lecturer: Improving learning in online education with a scaffolding-based conversational agent. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.

[47] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju. A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3506–3510. ACM, 2017.

[48] R. Yan, Y. Song, and H. Wu. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 55–64. ACM, 2016.

[49] G. Zhou, J. Wang, C. Lynch, and M. Chi. Towards closing the loop: Bridging machine-induced pedagogical policies to learning theories. In *Proceedings of the 10th International Conference on Educational Data Mining*, pages 112–119, 2017.

7 Appendix

Additional Qualitative Results, Study 1

Self-pacing versus guidance. In the first part of the study, 8 out of 116 users noted the benefits of freely navigating the video: *"I can rewind them and fast forward if I already know the concept."* Similarly, 22 out of 111 users in the second part of the study indicated value in freely scrolling through the written tutorial. These users frequently indicated frustration with the inability to freely navigate the material in the MathBot conversation.

On the other hand, 6 users in the first part of the study preferred that MathBot adapted its speed to their progression through concepts and questions, unlike the video. Similar sentiments were echoed by 15 users in the second part of the study, who preferred that MathBot explicitly guided them through concepts, unlike the written tutorial. Furthermore, 8 users in the first part of the study noted value in being able to scroll through earlier parts of the MathBot conversation to review concepts.

Human elements and interactivity. In the first part of the study, 7 out of 116 users found MathBot to be more agentic than the video. However, 9 users reported the opposite: *"Even though it was a video, it felt like a more personal experience because it was a human voice talking versus just reading on the screen."* In the second part of the study, 12 out of 111 users indicated that MathBot provided a greater sense of interaction than the written tutorial.

Requiring users to evaluate their knowledge. The video asked users to pause and think about problems; however, unlike MathBot, answering these questions correctly was not required. 22 out of 116 users in the first part of the study noted the value of MathBot holding them accountable for understanding concepts before progressing: *"When watching the video, I wasn't sure if I was*

actually understanding the concepts correctly." Similarly, although the Khan tutorial embedded problems between text, users could easily skip them, and 21 out of 111 users in the second part of the study found that being held accountable aided their learning. 10 learners also valued that MathBot provided more specific feedback on their answers than the tutorial.

Combining learning modules. 42 users in the first part of the study and 57 users in the second part of the study suggested that both tools could be particularly valuable in specific learning scenarios. For example, 8 users in the first part of the study thought the video was superior for learning concepts, whereas MathBot was better for learning how to apply those concepts: *"The best option for me would be to watch the video first, and then take part in the conversational computer program so that I could verify my understanding."* Similarly, 16 users indicated that, like the video, the written tutorial introduced concepts more effectively. 25 users in the first part of the study and 15 users in the second part of the study found that videos and written tutorials (respectively) were superior for learning concepts that required complex or detailed explanations.

Description of Thompson Sampling

As described in the text, we use a linear model with Thompson sampling to train the contextual bandit. Every time we get a new data point from a user answering a question, we run a linear regression on all previously recorded contexts and rewards. The linear regression contains all of the interactions between the context (a question identifier denoted by the indicator 1_j, which is 1 if the context is question j and 0 otherwise, and the pre-quiz score p) and the actions (whether an isomorph was shown, denoted by the indicator 1_i, and whether the explanation of the concept was skipped, denoted by the indicator 1_s). The reward is denoted by r.

$$r = \beta_0 + \sum_{j=1}^{n} \beta_j 1_j + \beta_p p + \beta_i 1_i + \beta_s 1_s + \sum_{j=1}^{n} \beta_{j2} 1_j 1_i + \sum_{j=1}^{n} \beta_{j3} 1_j 1_s + \beta_{p2} p 1_i + \beta_{p3} p 1_s + \varepsilon$$

Given the results of the regression, we have a posterior distribution over the coefficients, which is multivariate normal. This distribution is

$$\mathcal{N}(\hat{\beta}, \hat{\sigma}(X^T X)^{-1})$$

We sample a random β_t from this distribution. Then, we iterate over every possible action a and compute the action which would maximize the reward given β_t:

$$\arg\max_a \beta_t * x(a, c)$$

where c are the contextual variables and x are the covariates, which are computed from the contextual variables and the action terms, as described previously. This action is then given to the learner, and the reward is recorded.

Started modules	Finished modules	Module dropout (%)	Finished experiment	Post-module dropout (%)	Learning time > 1 min	Learning time <= 1 min (%)
143	134	6.3	134	0	116	13.4

Figure 12: Dropout summary statistics for the first part of **Study 1**, which measured participant preferences for video-based instruction versus instruction via MathBot. Participants who spent less than 1 minute on MathBot or the video were excluded from the analysis.

Started modules	Finished modules	Module dropout (%)	Finished experiment	Post-module dropout (%)	Learning time > 1 min	Learning time <= 1 min (%)
172	155	9.9	154	0.6	111	27.9

Figure 13: Dropout summary statistics for the second part of **Study 1**, which measured preferences for instruction via written tutorial versus instruction via MathBot. Participants who spent less than 1 minute on MathBot or the tutorial were excluded from the analysis.

N	Female (%)	Educ. > HS (%)	Mean age (years)	SD age (years)
116	69.6	81.9	35.4	10.5

Figure 14: Demographics of participants in the first part of **Study 1**, which measured preferences for video-based instruction versus instruction via MathBot.

N	Female (%)	Educ. > HS (%)	Mean age (years)	SD age (years)
111	76.4	82	34.6	8.7

Figure 15: Demographics of participants in the second part of **Study 1**, which measured preferences for instruction via written tutorial versus instruction via MathBot.

N	Mean MathBot rating	SD MathBot rating	Mean video rating	SD video rating	Mean MathBot time (min)	SD MathBot time (min)	Mean video time (min)	SD video time (min)
116	5.3	1.7	6.1	1.2	8.3	12.1	8.3	6.9

Figure 16: Experiential ratings and learning times of participants in the first part of **Study 1**, which measured preferences for video-based instruction versus instruction via MathBot. Experiential ratings were measured on a 1–7 scale, with 4 as the neutral option.

N	Mean MathBot rating	SD MathBot rating	Mean tutorial rating	SD tutorial rating	Mean MathBot time (min)	SD MathBot time (min)	Mean tutorial time (min)	SD tutorial time (min)
111	5.8	1.3	5.9	1.2	7.3	4.1	10.7	13.6

Figure 17: Experiential ratings and learning times of participants in the second part of **Study 1**, which measured preferences for instruction via written tutorial versus instruction via MathBot. Experiential ratings were measured on a 1–7 scale, with 4 as the neutral option.

Condition	Started module	Completed module	Module dropout (%)	Completed experiment	Post-module dropout (%)	Learned $>$ 2 min.	Learned $<=$ 2 min. (%)
Khan Video	251	231	8.0	221	4.3	187	15.4
MathBot	237	221	6.8	213	3.6	182	14.6

Figure 18: Dropout summary statistics for **Study 2**, which measured learning gains achieved by instruction via MathBot versus instruction via Khan Academy videos and written tutorials. Participants who spent less than 2 minute on their randomly-assigned learning module were excluded from the analysis.

Condition	N	Female (%)	Educ. > HS (%)	Mean age (years)	SD age (years)
Khan Video	187	58.8	85.0	33.6	9.7
MathBot	182	71.7	79.7	34.0	10.7

Figure 19: Demographics of participants in **Study 2**, which measured learning gains achieved by instruction via MathBot versus instruction via Khan Academy videos and written tutorials.

Condition	N	Mean time (min)	SD time (min)	Mean pre-score	SD pre-score	Mean post-score	SD post-score	Mean gain	SD gain	Mean PLG	SD PLG
Khan Video	187	28.9	21.6	2.3	1.7	8.1	3.7	5.7	3.6	0.6	0.4
MathBot	182	28.4	20.3	2.5	1.9	8.6	3.3	6.1	3.2	0.7	0.3

Figure 20: Learning outcomes of participants in **Study 2**, which measured learning gains achieved by instruction via MathBot versus instruction via Khan Academy videos and written tutorials. The pre- and post-learning assessments were on a 12-point scale.

Test question	Pre-MathBot (%)	Post-MathBot (%)	Pre-Khan (%)	Post-Khan (%)	Gain Math-Bot (pp)	Gain Khan (pp)	PLG Math-Bot (%)	PLG Khan (%)
find term explicit 1	26.9	74.7	22.5	74.3	47.8	51.9	65.4	66.9
find term explicit 2	26.4	73.6	25.7	73.3	47.3	47.6	64.2	64.0
find term recursive 1	4.4	61.0	2.7	60.4	56.6	57.8	59.2	59.3
find term recursive 2	4.9	60.4	6.4	62.6	55.5	56.1	58.4	60.0
verify explicit mc 1	32.4	69.2	37.4	57.2	36.8	19.8	54.5	31.6
verify explicit mc 2	53.3	69.8	44.4	72.7	16.5	28.3	35.3	51.0
write explicit 1	1.1	49.5	2.7	49.7	48.4	47.1	48.9	48.4
write explicit 2	1.1	51.1	2.1	50.3	50.0	48.1	50.6	49.2
write recursive 1a	28.6	86.8	19.8	73.3	58.2	53.5	81.5	66.7
write recursive 1b	24.7	87.4	26.2	78.1	62.6	51.9	83.2	70.3
write recursive 2a	24.7	89.6	21.4	79.1	64.8	57.8	86.1	73.5
write recursive 2b	18.7	85.7	23.0	74.9	67.0	51.9	82.4	67.4

Figure 21: Learning outcomes for each quiz question in **Study 2**. The first four columns show the percentage of participants who answered the quiz question correctly. PLG denotes proportional learning gain, which is calculated by dividing the pre/post percentage point gain by the maximum possible percentage point gain. Users in the MathBot condition performed substantially better on the "write recursive" questions than users in the Khan Academy condition. The "verify explicit" questions exhibit conflicting performance gains across conditions. For all other questions, performance is similar across conditions.

Experimentation Strategy	Dropout	Confidence Interval
Bandit	0.087	[0.048, 0.125]
Uniform	0.152	[0.102, 0.201]

Figure 22: For **Study 3**, dropout rates for the two learning experimentation strategies, along with 95% confidence intervals.

Experimentation Strategy	Mean pre-score	SD pre-score	Mean post-score	SD post-score	Mean learning gains	SD learning gains	Mean lesson time (min)	SD lesson time (min)
Bandit	3.87	2.47	12.2	3.82	8.29	4.00	18.4	9.7
Uniform	4.05	2.49	12.0	4.11	7.96	4.17	20.1	11.2

Figure 23: For **Study 3**, average pre- and post-learning quiz scores, learning gains, and total time spent on the lesson for the two learning experimentation strategies, along with standard deviations.

Policy	Expected Reward Per Question	95% Confidence Interval
Bandit	46.3	[41.7, 51.0]
Uniform random	55.0	[46.3, 63.6]
Concept only (original)	35.9	[30.9, 40.8]
Concept + Isomorph	1.0	[-6.7, 8.7]
Isomorph only	14.4	[8.8, 20.1]
Skip concept, no Isomorph	21.2	[15.0, 26.5]

Figure 24: For **Study 3**, average expected reward per question with 95% confidence intervals for the final policy learned from the bandit and uniform random conditions, along with the original strategy used by MathBot in Studies 1 and 2 (only show concept).