Learning to Give Useful Hints: Assistance Action Evaluation and Policy Improvements

Robin Schmucker[1](✉), Nimish Pachapurkar[2], Shanmuga Bala[2], Miral Shah[2], and Tom Mitchell[1]

[1] Carnegie Mellon University, Pittsburgh, PA, USA
{rschmuck,tom.mitchell}@cs.cmu.edu
[2] CK-12 Foundation, Palo Alto, CA, USA
miral.shah@ck12.org

Abstract. We describe a fielded online tutoring system that learns which of several candidate assistance actions (e.g., one of multiple hints) to provide to students when they answer a practice question incorrectly. The system learns, from large-scale data of prior students, which assistance action to give for each of thousands of questions, to maximize measures of student learning outcomes. Using data from over 190,000 students in an online Biology course, we quantify the impact of different assistance actions for each question on a variety of outcomes (e.g., response correctness, practice completion), framing the machine learning task as a multi-armed bandit problem. We study relationships among different measures of learning outcomes, leading us to design an algorithm that for each question decides on the most suitable assistance policy training objective to optimize central target measures. We evaluate the trained policy for providing assistance actions, comparing it to a randomized assistance policy in live use with over 20,000 students, showing significant improvements resulting from the system's ability to learn to teach better based on data from earlier students in the course. We discuss our design process and challenges we faced when fielding data-driven technology, providing insights to designers of future learning systems.

Keywords: intelligent tutoring systems · multi-armed bandits

1 Introduction

Intelligent tutoring systems (ITSs) are part of everyday life for millions of students worldwide. ITSs promote accessible learning experiences that can narrow the educational achievement gap [24] and that, in some cases, can be as effective as human tutoring [13]. In their effort to create effective learning systems, ITS designers are confronted with a plethora of design decisions ranging from specifying general instructional design principles [12] to the creation of individual learning and practice materials. Designers rely on their domain expertise and

O. Viberg et al. (Eds.): EC-TEL 2023, LNCS 14200, pp. 383–398, 2023.
https://doi.org/10.1007/978-3-031-42682-7_26

consider effects of different design choices, but in many cases it is difficult to predict which exact choice will benefit students the most [18], and often thousands of design decisions have to be made on a case by case basis (e.g., which exact hint is most effective for this specific question). In this context, the promise of data-driven design approaches is that they can leverage system usage data to evaluate the effects of different design choices inside the ITS on student learning and can improve learning outcomes by refining the ITS automatically over time.

This work describes an online tutoring system that embraces a data-driven design approach by using large-scale student data to learn which of several candidate assistance actions to provide to students after they answer a practice questions incorrectly. We report results from a study–analysing data from over 190,000 students in a Biology course–evaluating the impact of individual assistance actions and assistance policies on different measures of learning outcomes. We discuss rationales behind our methodology and provide insights for the design of future learning systems. The main contributions of this work include:

- *Quantifying effects of assistance.* We evaluate effects of over 7,000 individual assistance actions on a variety of student learning outcome measures (e.g., practice completion). We study the relationship among different measures and design an assistance policy training algorithm that for each question decides on the most suitable policy training objective to optimize the student's success at the current question as well as their overall session performance.
- *Offline policy optimization.* We compute statistically significant estimates on the effects of multi-armed bandit policies trained to optimize different learning outcome measures. Studying assistance actions selected by these policies, we find that there is no single best assistance type (e.g., hint, vocabulary).
- *Live A/B evaluation.* We evaluate the assistance policy trained using our algorithm in comparison to a randomized assistance policy in live use with over 20,000 students. The system's ability to learn to teach better using data from prior students improves learning outcomes of future students significantly.

2 Related Work

2.1 Evaluating Treatment Effects Inside ITSs

Initially the effects of ITSs on student learning were evaluated at the *system level* by comparing a group of students that uses the ITS to a control group in a post-test [13]. Later research focuses on studying the effects of individual *instructional design choices* [12] and conducts experiments with students that interact with different configurations of the same learning system (e.g., [16,17]). With the ever increasing popularity of online ITSs, large-scale student log data becomes available, which enables investigating the effects of increasingly detailed system design choices, up to the choice of individual *practice questions* and *hints*.

As part of this development, ASSISTments introduced AXIS [27], the E-TRIALS TestBed [19] and the TeacherASSSIST system [20] to allow educators and researchers to create and evaluate the effectiveness of different problem sets and *on-demand* assistance materials. In the context of massive open online courses (MOOCs), DynamicProblem [28] was introduced as a proof-of-concept system that supports bandit algorithms [14] to collect feedback from students regarding the helpfulness of individual assistance materials. Relatedly, the MOOClet framework [23] allows instructors to specify multiple versions of educational resources and to evaluate them in A/B tests using randomization and bandit algorithms. The UpGrade system [8] was introduced as a flexible A/B testing framework designed for easy integration into various learning systems.

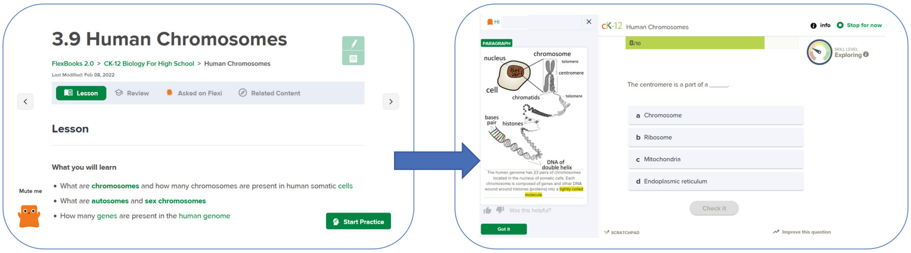
This work describes a fielded online tutoring system at CK12.org that learns to provide effective assistance actions (e.g., choose one of multiple available hints) to support students after they answer practice questions incorrectly. We use offline evaluation techniques [15] to leverage log data capturing over 4,800,000 assistance requests from over 190,000 students in a Biology course. The unprecedented scale of this data enables us to compute statistically significant estimates on the effects of *individual* assistance actions and assistance policies on different measures of learning outcomes. We further evaluate the effectiveness of the learned assistance policy in live use with over 20,000 students.

2.2 Data-Driven Assistance Policies

Here, we provide a concise overview of related research that uses data-driven techniques to support students during the problem solving process via bandit and reinforcement learning (RL) algorithms. For a comprehensive review on RL in the education domain we refer to a survey by Doroudi et al. [6].

Barnes and Stamper [3] induced a Markov decision process (MDP) based on hundreds of student solution paths and used RL to generate new hints inside a logic ITS. Chi et al. [4] modeled a physics tutor via an MDP with 16 states and learned a RL policy to improve student learning outcomes by deciding whether to ask the student to reflect on a problem or to tell them additional information. Georgila et al. [9] used Least-Squares Policy Iteration to learn a feedback policy for an interpersonal skill training system using data describing over 500 features from 72 participants. Ju et al. [10] identified critical pedagogical decisions based on Q-value and reward function estimates derived from logs of 1,148 students inside a probability ITS. Relatedly, Ausin et al. [1,2] explored Gaussian Process- and inverse RL-based approaches to address the credit assignment problem inside a logic ITS. A recent series of works [7,25,26] used a random policy to collect data from 500 students in an operational command course and explored offline RL techniques to learn adaptive scaffolding policies based on the ICAP framework.

A recent study by Prihar et al. [21] compared a multi-armed bandit algorithm based on Thompson Sampling to a random assistance policy with respect to their ability to increase students' *success on the next question*. In a two-month long experiment with 2,923 questions they find the bandit algorithm to be only slightly more effective than the random policy and argue that this is due to sample size limitations (on average 6.5 samples per action). In contrast, this work accurately estimates the impact of individual actions on *different measures of learning outcomes* by leveraging hundreds of samples per action (Table 1).



3.9 Human Chromosomes

FlexBooks 2.0 > CK-12 Biology For High School > Human Chromosomes

Last Modified: Feb 08, 2022

Lesson

What you will learn

- What are chromosomes and how many chromosomes are present in human somatic cells
- What are autosomes and sex chromosomes
- How many genes are present in the human genome

The human genome has 23 pairs of chromosomes located in the nucleus of somatic cells. Each chromosome is composed of genes and other DNA wound around histones (proteins) into a tightly coiled molecule.

The centromere is a part of a ______.

a Chromosome

b Ribosome

c Mitochondria

d Endoplasmic reticulum

Lesson   Adaptive Practice

Fig. 1. Example views from the concept *Human Chromosomes*. [Left] In the *Lesson* section the student interacts with multi-modal learning materials. [Right] During *Adaptive Practice* the student develops and tests their understanding by answering practice questions. In the shown example the system displays a paragraph with illustration to assist the student before they reattempt the question after an initial incorrect response.

Further, in contrast to Prihar et al. [21], we quantify treatment effects by automatically providing assistance in response to incorrect student responses which avoids self-selection effects when assistance is only shown upon student request.

3 CK-12 FlexBook 2.0 System

The CK-12 Foundation is a non-profit organization that provides millions of students worldwide with access to free educational resources. CK-12's Flexbook 2.0 system[1] is a web-based ITS that offers a large variety of courses for different subjects and grade levels. Each course consists of a sequence of *concepts*. Each concept has a *Lesson* section with learning materials and an *Adaptive Practice* (AP) section where students can develop and test their understanding (Fig. 1).

The AP section features item response theory (IRT)-driven question sequencing and tries to select practice questions matching the student's ability level (*Goldilocks* principle [12]). After the system selects a question, the student can request a hint before submitting a first response. If the first response is incorrect, the system provides immediate feedback by displaying one *assistance action* (e.g., a hint or vocabulary) and the student reattempts the question. Afterwards, the system uses the student's first response to update the student's ability estimate and selects the next practice question. This process repeats until the student completes the AP session successfully by achieving 10 correct responses or until the question pool is exhausted in which case the student can try again.

This paper centers on the question of how we can employ data-driven techniques to learn an *assistance policy* that selects the most effective assistance action as feedback for each individual question. We focus on CK-12's *Biology for High School* course which is used by hundreds of thousands of students each year and whose content has been developed and refined for over ten years. The course covers hundreds of concepts and features over 12,000 questions corresponding

[1] https://www.ck12.org.

to five categories: *multiple-choice*, *select-all-that-apply*, *fill-in-the-blank*, *short-answer* and *true-false*. The AP system associates each question with a set of potential assistance actions. An exception are true-false questions which students only attempt once. The average non-true-false question is associated with 4.8 different actions. Each action falls into one of six categories: *hint*, *paragraph* (short text from lesson), *vocabulary* (keyword definitions), *remove distractor* (removes multiple-choice/select-all-that-apply response option), *first letter* (shows initial of fill-in-the-blank/short-answer solution) and *no assistance* (as baseline).

4 Methodology

4.1 Formal Problem Statement

We denote the set of practice questions inside the system as $Q = \{q_1, \ldots, q_k\}$. Each question $q \in Q$ is associated with a set of n_q assistance actions $A_q = \{a_{q,1}, \ldots, a_{q,n_q}\}$ that the system can use to support students after their first incorrect response. In this work, we approach the problem of learning one effective assistance policy for the entire practice system by learning one question-specific multi-armed bandit policy π_q for each practice question $q \in Q$. During deployment, π_q responds to each assistance query for question q by selecting one assistance action $a_q \in A_q$ and in return receives a real-valued reward $r_q \in \mathbb{R}$ which is assumed to be sampled from an action-specific and time-invariant distribution R_{a_q} with mean μ_{a_q}. The optimal question-specific assistance policy π_q^* maximizes the expected reward by always selecting action $a_q^* = \arg\max_{a_q \in A_q} \mu_{a_q}$.

For us, multi-armed bandits are a framework that enables our system to automatically make design decisions by learning from the observed behavior of earlier students. It is difficult for experts to predict the most effective design ahead of time [18] and the bandit framework enables the system to estimate the effects of potential design choices using student data to refine the ITS automatically over time. Section 6 discusses benefits and limitations of our bandit formulation.

4.2 Data Collection

This work focuses on an online high-school Biology course that has been in continuous refinement for over ten years. Because of this, its content base features *multiple* assistance actions for individual questions. This raises the question of *what type* of assistance action is most effective for a particular question (e.g., should one provide hints or keyword definitions?). Even if the domain experts decide on a specific type of assistance, it is still unclear *what action* from the reduced action set is most effective (e.g., which exact hint should one show?).

To address these questions, we conduct an experiment to quantify the impact of *individual* assistance actions on different measures of learning outcomes. Starting from Aug 23rd, 2022, a randomized assistance policy was deployed. Each time this policy is queried to provide assistance for a question $q \in Q$, it uniformly (with same chance) chooses one action at random from the set A_q. An overview of the data collected up to Jan 11th, 2023, is provided by Table 1.

4.3 Measures of Learning Outcomes

One key question in this work is how to define a reward function that takes as input information about a student practice session and that outputs a reward value that quantifies the degree to which the assistance provided by the system led to successful learning. This reward function is central as it serves as objective during policy training and thus directly affects the experience of future students.

Table 1. Data collection overview. The *Overall* column shows statistics on the raw data collected for all content in the Biology course. The *Offline/Online Evaluation* columns show statistics on the data that went into the offline/online evaluation experiments.

	Overall	Offline Eval.	Online Eval.
# of questions	12,496	1,336	4,521
# of assistance actions	36,354	7,707	11,406
# of concepts	470	166	166
# of students	191,554	164,516	27,268
# of practice sessions	1,274,072	1,007,850	62,464
# of student responses	20,425,691	17,081,054	953,185
# of shown actions	4,842,856	3,266,171	234,178
average correctness	63.3%	60.0%	61.1%
average completion	75.1%	77.1%	69.4%
collection period (days)	142	142	9

The designers of the practice system want to promote growth in *student knowledge* as well as *student engagement*. Unfortunately, student knowledge and engagement are both unobservable variables and the system is limited in that it can only access data that describes the student's observable interactions with the website interface. Because of this, we compiled a list specifying different measures of learning outcomes that can be computed from the observed log data:

- *Reattempt correct*: Binary indicator ($\{0, 1\}$) of whether the student is correct on the reattempt directly after the assistance action.
- *Student Ability*: 3-Parameter item response theory (IRT)-based ability estimate using all first attempt responses computed at end of session. IRT is a logistic model that explains response correctness by fitting student- (ability) and question-specific (difficulty, discrimination, guessing) parameters [5].
- *Session success*: Binary indicator ($\{0, 1\}$) of whether the student achieves 10 correct responses in the practice session.
- *Future correct rate*: Proportion of student's correct responses on first attempts on other practice questions following the assistance action.
- *Next question correct*: Binary indicator ($\{0, 1\}$) of whether the student is correct on the next question following the assistance action (used in [21]).

- *Future response time*: Measures the student's average response time on questions after an assistance action in seconds (individual question response time values are capped at 60 s (95% percentile) to mitigate outliers).
- *Student confidence*: Tertiary indicator ($\{1, 2, 3\}$) of the student's self-reported confidence level at the end of the practice session.

In the experiments we study the relationships between these individual outcome measures (Sect. 5.1) which leads us to defining our final reward function R as

$$R(s, q) = 0.4 \cdot \text{reattempt\_correct}(s, q) + 0.6 \cdot \text{student\_ability}(s). \tag{1}$$

Here, s represents information about a student's entire practice session and q indicates the question for which the student received assistance. The reward value is computed as a weighted sum that considers the student's success at reattempting question q as well as their overall practice session performance.

4.4 Offline Policy Optimization and Evaluation

Preprocessing. Before policy optimization and evaluation we perform the following preprocessing steps: (i) To avoid early dropouts, we only consider practice sessions in which students respond to at least five different questions. (ii) To avoid memorization effects, we only consider each student's first practice attempt for each concept. (iii) To avoid confounding, we estimate the effects of individual assistance actions using only practice sessions in which the student did not request a hint before their first attempt. (iv) To achieve high confidence in our effect estimates we focus on practice questions with at least 100 samples per assistance action. As a result, we consider a set of 1,336 unique questions from 166 concepts associated with 7,707 assistance actions and draw from over 3,200,000 assistance queries occurring in over 1,000,000 different practice sessions (Table 1).

Optimization. To train and evaluate the effects of different assistance policies without conducting repeated live experiments we rely on offline policy optimization [15] and leverage log data collected by the randomized exploration policy. First, we estimate the effectiveness of individual assistance actions by computing the mean value for each learning outcome measure across all relevant practice sessions. From there, our experiments study various multi-armed bandit policies trained to optimize different outcome measures. In preliminary experiments, we found that when using measures with high variance as training objectives (i.e., *student ability* and *session success*), the conventional policy optimization approach–that for each question selects the assistance action estimated to be optimal–struggles to reliably identify actions that perform well in the evaluation on separate test data. For the average question we found optimizing policies for *reattempt correctness*–a measure with focus on a single question and thus lower variance–to be the most effective way to also boost *student ability* and *session success* due to its positive correlations to the other measures (Fig. 3 left).

Still, for a sizeable number of questions the conventional approach yielded better policies when directly optimizing for the measure of interest (Sect. 5.2). These tended to be questions with more available data or with larger differences in the effects of individual assistance actions. This motivated the design of a training algorithm that for each question automatically decides whether we have sufficient data to optimize the measure of interest (e.g., *reward*) directly or whether we should use the low variance *reattempt correctness* measure. We first use the training data to identify the two actions that optimize the measure of interest and *reattempt correctness*. We then conduct a one-sided Welch T-Test to decide whether the former has a significantly larger effect on the measure of interest than the *reattempt correctness* action and if not select the low variance *reattempt correctness* measure as the question-specific training objective.

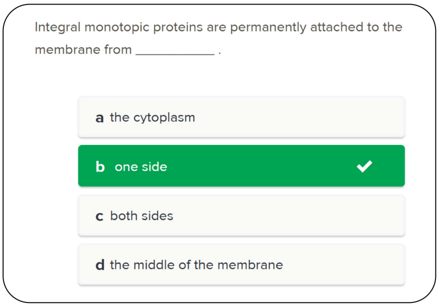
Evaluation. In the offline evaluation experiments we report mean performance estimates derived from a 20 times repeated 5-fold cross validation. In each fold 80% of practice sessions are used for policy training and the remaining 20% are used for testing. This process yields a statistically unbiased estimate of the bandit policy's performance as it simulates a series of interactions with different students inside the system and avoids overfitting effects of sampling with replacement-based approaches [15]. For the significance test we determine a suitable p-value for each individual outcome measure by evaluating $p \in \{0.01, 0.02, \ldots, 0.1\}$ via cross-validation. The final policy used in live A/B evaluation is trained using data from all practice sessions and optimizes our reward function (Eq. 1).

5 Results

5.1 Assistance Action Evaluation

We estimate the effects of individual assistance actions on different measures of learning outcomes by leveraging the student log data collected by the randomized assistance policy (Sect. 4.2). One example of the results of this evaluation process is provided by Fig. 2. It shows the question text, the set of available assistance actions and estimates on how each action affects different outcome measures. We can see how the *paragraph* that provides detailed information leads to the highest reattempt correctness rate. In comparison, *hint 1* leads to a lower reattempt correctness, but conveys insights that improve overall session performance as captured by the final student ability score. We can also identify actions that are not helpful. For example, *hint 2* and *vocabulary* both lead to worse outcomes than showing *no assistance*. Overall, these estimates are very compelling for the content creators, as they allow them to reflect on how the individual resources they designed affect the student experience in different ways.

To study the relationships between the different learning outcome measures we analyse average within question correlations across the 1,336 questions (Fig. 3). We focus on within question correlation instead of total correlation to be more robust towards effects caused by systematic differences between individual questions (e.g., difficulty). We observe that reattempt correctness is most correlated with the IRT-based student ability estimates ($r = 0.27$) and that it is mostly uncorrelated with next question correctness ($r = 0.04$). This shows that while assistance actions can improve students' overall session performance, due to differences between individual questions, it is not enough to focus only on the next question. Matching our intuition, ability estimates correlate with session success ($r = 0.35$), future correctness ($r = 0.64$) and next question correctness rates ($r = 0.36$). This is because these measures all consider first attempt response correctness. Student response time has a low positive correlation to student ability ($r = 0.23$) and self-reported student confidence shows very low correlations with the other considered measures.



Integral monotopic proteins are permanently attached to the membrane from ___________ .

a the cytoplasm

b one side ✓

c both sides

d the middle of the membrane

(a) Practice Question.

- *hint 1*: The prefix **mono-** means **single** or **alone**.
- *hint 2*: The word **integral** means **essential** or **necessary to make whole**.
- *paragraph*: Transmembrane proteins span the entire plasma membrane. Transmembrane proteins are found in all types of biological membranes. Integral monotopic proteins are permanently attached to the membrane from only one side.
- *vocabulary*:
  - cytoplasm: Material inside the cell membrane, including the watery cytosol and other cell structures except the nucleus if one is present.
  - protein: A peptide that is greater than one hundred amino acids in length.

(b) Assistance Actions.

	n	Reward	Reatt. Cor.	Stud. Abil.	Sess. Succ.
no assist.	872	0.025 ±.079	0.313 ±.031	-0.167 ±.125	0.925 ±.017
hint 1	986	**0.394** ±.077	0.693 ±.029	**0.195** ±.121	0.934 ±.015
hint 2	940	-0.029 ±.081	0.255 ±.028	-0.219 ±.130	0.920 ±.017
paragraph	940	0.363 ±.078	**0.774** ±.027	0.088 ±.121	**0.935** ±.016
vocabulary	936	-0.033 ±.078	0.259 ±.028	-0.228 ±.126	0.929 ±.016

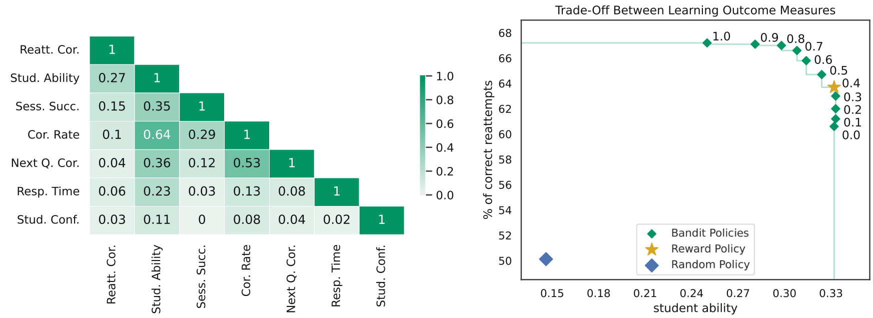
(c) Action Performance Estimates.

Fig. 2. Example of assistance action evaluation for one individual question.

Before moving on to training assistance policies we quantify the degree to which we can differentiate the effects of assistance actions for individual questions based on the available log data via analysis of variance (ANOVA). Compared to the bandit problem which tries to identify the single most effective action, ANOVA focuses on the simpler question of whether there are statistically significant differences in mean effects between individual actions. For a p-value of 0.05 ANOVA rejects the null hypothesis for *reattempt correctness* for 83.2% ($n = 1,111$), for *student ability* for 13.3% ($n = 178$) and for *session success rate* for 9.6% ($n = 128$) of questions. We can explain this by studying sample variance and the effect size gaps between the most and least effective assistance action for each outcome measure. By only focusing on the current question, *reattempt correctness* exhibits on average across the 1,336 question a better ratio between action effect gaps and sample variance ($\delta = 0.230$, $\sigma^2 = 0.229$) compared to the *student ability* ($\delta = 0.302$, $\sigma^2 = 3.665$) and *session completion rate* ($\delta = 0.042$, $\sigma^2 = 0.085$) measures which describe overall session performance.

5.2 Offline Policy Evaluation

While ANOVA finds significant differences in mean action effects on *reattempt correctness* for most questions, it only detects differences on *student ability* and *session completion* for a smaller subset of questions. For our offline policy evaluation process this suggests that it is difficult to reliably identify the optimal



	Reatt. Cor.	Stud. Ability	Sess. Succ.	Cor. Rate	Next Q. Cor.	Resp. Time	Stud. Conf.
Reatt. Cor.	1						
Stud. Ability	0.27	1					
Sess. Succ.	0.15	0.35	1				
Cor. Rate	0.1	0.64	0.29	1			
Next Q. Cor.	0.04	0.36	0.12	0.53	1		
Resp. Time	0.06	0.23	0.03	0.13	0.08	1	
Stud. Conf.	0.03	0.11	0	0.08	0.04	0.02	1

Fig. 3. [Left] Average within question correlations between individual measures of learning outcomes across 1, 336 questions. [Right] Pareto front visualizing the estimated average performance of policies optimized to increase the final student ability estimates (x-axis) and reattempt correctness rate (y-axis) across 178 questions. Each bandit policy is marked with a number that indicates how it weights the two objectives.

Table 2. Offline evaluation of various policies across the 178 questions for which ANOVA indicated significant differences ($p < 0.05$) in mean action effects on student ability. The first two rows show no assistance and randomized policies as baselines. The following four rows are bandit policies optimized directly for different outcome measures and the reward function. We report mean values and 95% confidence intervals.

policy/measure	Reward	Reatt. Cor.	Stud. Abil.	Sess. Succ.
no assistance	0.241 ±.064	0.433 ±.023	0.113 ±.106	0.801 ±.039
random	0.288 ±.067	0.501 ±.021	0.146 ±.109	0.806 ±.039
reattempt correct	0.419 ±.071	**0.672** ±.020	0.250 ±.115	0.816 ±.038
student ability	0.442 ±.068	0.606 ±.025	**0.332** ±.109	0.819 ±.038
session success	0.371 ±.068	0.573 ±.025	0.237 ±.109	0.817 ±.037
reward	**0.454** ±.068	0.637 ±.023	**0.332** ±.108	**0.820** ±.038

assistance actions for the latter two measures even when having access to hundreds of samples per action. Indeed, in preliminary experiments we found that action effect rankings based on training data often deviate from rankings based on separate test data. For the average question we found training assistance policies based on *reattempt correctness* estimates to be the most effective way to boost all three outcome measures. This is due to its lower variance and the fact that improvements in *reattempt correctness* are positively correlated with improvements in *student ability* and *session completion rates* (Fig. 3 left).

Still, for 178 (13.3%) of the 1,366 questions ANOVA detected significant differences in action effects on *student ability* which is a core measure of interest. To study the relationship between *reattempt correctness* and *student ability* for these 178 questions, we train bandit policies for different objectives. Here, analog to the reward function (Eq. 1), we assign each policy a weight $w_1 \in \{0, 0.1, \ldots, 1.0\}$ and compute its reward values by linearly weighting *reattempt correctness* with w_1 and *student ability* with $1 - w_1$. We visualize the Pareto front defined by the resulting policies (Fig. 3 right) and observe performance estimates that range in *reattempt correctness* rates from 60.6% to 67.2% and in *student ability* from

Table 3. Offline evaluation of various policies across 1, 336 questions. The first two rows show no assistance and randomized policies as baselines. The following four rows are bandit policies optimized with our algorithm for different learning outcome measures and the reward function. We report mean values and 95% confidence intervals.

policy/measure	Reward	Reatt. Cor	Stud. Abil.	Sess. Succ.
no assistance	0.218 ±.026	0.498 ±.008	0.032 ±.042	0.815 ±.014
random	0.255 ±.026	0.551 ±.007	0.058 ±.042	0.820 ±.013
reattempt correct	0.327 ±.026	**0.666** ±.007	0.101 ±.043	**0.827** ±.013
student ability	0.327 ±.026	0.660 ±.007	**0.105** ±.043	**0.827** ±.013
session success	0.326 ±.026	0.663 ±.007	0.101 ±.043	**0.827** ±.013
reward	**0.328** ±.026	0.664 ±.007	0.104 ±.043	**0.827** ±.013

Table 4. Types of assistance actions selected by the multi-armed bandit policy learned using our reward function for all 1,336 questions. The individual columns show how the policy focuses on different types of assistance actions for different types of questions.

action/question	Mult.-Choice	All-That-Apply	Fill-Blank	Short-Answ.
no assistance	5.0%	7.3%	1.8%	3.4%
hint	11.4%	15.6%	6.6%	5.6%
paragraph	51.2%	43.1%	57.9%	55.1%
vocabulary	5.7%	16.5%	1.8%	3.4%
hide distractor	26.8%	17.4%	-	-
first letter	-	-	31.7%	32.6%

0.250 to 0.332. All learned policies outperform the *random* policy significantly. In collaboration with domain experts we select $w_1 = 0.4$ as reward function to train the assistance policy for live evaluation as it improves both measures substantially. Table 2 provides detailed performance statistics for policies trained to optimize different outcome measures across the 178 questions.

To train an assistance policy for all 1,366 questions we designed an algorithm that for each question decides whether we have sufficient data to optimize the measure of interest (e.g., *reward*) directly or whether we should use the low variance *reattempt correctness* measure (Sect. 4.4). Table 3 shows average performance metrics across 1,336 questions for a policy that always selects the *no assistance* action, the *random* policy, and four policies trained using our algorithm to optimize *reattempt correctness* rates, *student ability*, *successful session completion* rates, and *reward* function. The algorithm resolves the variance issue and the trained policies enhance the student experience in different ways.

Lastly, we study for which types of questions the final policy offers which types of assistance actions to maximize the *reward* objective. Table 4 shows for each question type for what proportion of questions the policy finds a certain assistance type to be most effective. We find that the policy utilizes a diverse blend of different assistance types for each type of question and that paragraph actions are selected most frequently overall. Because of this, we compare the effects of a policy that always selects *paragraph* actions to the trained *reward* policy in an additional experiment. Across the 1,175 questions with paragraphs, we find that the *reward policy* outperforms the *paragraph* policy in all outcome measures (*reward:* 0.336/0.299, *reattempt correctness:* 67.3%/61.5%, *student ability:* 0.112/0.089, *session success:* 84.0%/83.7%). Thus, the data-driven approach benefits by selecting effective teaching actions on a question-by-question basis.

Table 5. Live policy evaluation. We randomly assign student practice sessions to the randomized policy ($n = 31,527$) and the learned bandit policy ($n = 30,937$) condition, track various outcome measures and report mean values and 95% confidence intervals.

policy/measure	Reward	Reatt. Cor.	Stud. Abil.	Sess. Succ.
random	0.753 ±.016	0.585 ±.004	0.866 ±.026	0.676 ±.005
bandit	**0.881** ±.015	**0.683** ±.004	**1.013** ±.024	**0.712** ±.005

5.3 Online Policy Evaluation

To evaluate the policy optimized using our training algorithm we compare its ability to provide students with effective assistance actions to the randomized assistance policy. For this a nine-day long A/B evaluation (Apr 5th to Apr 13th, 2023) was conducted in which practice sessions for the 166 studied concepts were randomly assigned to the bandit and the randomized policy condition. During this period we collected log data describing over 62,000 sessions from over 20,000 different students (Table 1). Even though the learned assistance policy implemented only 1,336 question-specific bandit policies, it was able to provide learned actions for 87,721 (74.9%) of the 117,180 queries and only needed to default to random action selection in 29,459 (25.1%) cases. This is because the majority of incorrect responses occur on a smaller number of questions.

Table 5 reports average performance for the two different policies. The trained assistance policy outperforms the randomized policy significantly in all outcome measures, achieving on average a 9.8% improvement in reattempt correctness rate and a 0.147 higher student ability estimate. The session success rate improvement from 67.6% to 71.2% corresponds to a 11.1% reduction in sessions in which students did not achieve the practice target. We note that in contrast to the offline evaluation (Sect. 5.2) where we estimate effects based on individual assistance queries, here we compute metrics based on the session level.

6 Discussion

The results show how the offline evaluation approach can leverage large-scale student log data to quantify the impact of individual assistance actions (e.g., hints and keyword definitions) for each question on different measures of student learning outcomes (e.g., reattempt correctness, practice completion). This allows ITS designers to monitor and reflect on fine-grained design decisions inside the system (e.g., which assistance action for which question) and enables a data-driven design process in which the designers can specify a reward function to train an assistance policy that promotes the desired student learning experience. The live use evaluation confirms that this process provides the system with the ability to learn to teach better automatically over time, by showing how the actions selected by the learned multi-armed bandit policies lead to significant improvements in learning outcomes compared to a randomized assistance policy.

By studying the assistance actions selected by our optimized policy (Fig. 4) we observe that there is no single best type of assistance that is always most effective. This emphasizes the importance of algorithms that can identify the most effective teaching action for each individual practice question based on observational data. Interestingly, the policy blends more informative (e.g., paragraphs) with less informative assistance actions (e.g., hints) and decides for some questions to provide no additional help at all. This indicates a trade-off between giving and withholding information during the learning process which is a phenoma that has been described as *assistance dilemma* in prior research [11].

Our methodology combines multi-armed bandit and offline policy evaluation techniques [15] with large-scale student log data to compute high confidence estimates on the effects of individual assistance actions. One inherent property of our multi-armed bandit formulation of the problem is that it focuses on selecting the teaching action that is most effective for the *average* student and does not attempt to provide assistance conditioned on the *individual* student, and does not capture synergies that could occur when certain combinations of assistance actions are shown to a student in the same practice session. While a reinforcement learning approach could be used to address both of these shortcomings, the volume of training data required for such an approach would increase dramatically, and it would be much harder to compute statistically significant estimates on the effects of individual policies before deployment. We will explore the potential of personalized assistance policies via contextual bandit and reinforcement learning algorithms [7,22] in future work. Another future direction is the integration of online bandit algorithms [14] into the current system to keep enhancing the assistance policies by adaptively sampling individual actions based on evolving effect estimates in live deployment. Adaptive sampling is of particular interest to us as the pool of questions and assistance actions is continuously refined.

7 Conclusion

In this paper we discussed a large-scale online tutoring system that uses student log data to learn which of several candidate assistance actions (e.g., hints and paragraphs) to provide to students when they answer a particular practice question incorrectly. We used offline policy evaluation to leverage data from over 1,000,000 student practice sessions to evaluate the effects of individual assistance actions and multi-armed bandit policies on various measures of learning outcomes. We studied relationships among outcome measures and designed an algorithm to train an assistance policy that optimizes the student's success at answering the current question, as well as their overall practice session performance. In a live evaluation with over 20,000 students we compared the trained assistance policy to a randomized assistance policy finding that the system's ability to learn to select more effective teaching actions automatically over time enables significant improvements in learning outcomes of future students. The trained policy now supports thousands of students practicing Biology each day.

Acknowledgements. This research was supported by the AFOSR under award FA95501710218.

References

1. Ausin, M.S., Azizsoltani, H., Barnes, T., Chi, M.: Leveraging deep reinforcement learning for pedagogical policy induction in an intelligent tutoring system. In: Proceedings of the 12th International Conference on EDM, pp. 168–177. EDM, Montréal, Canada (2019)
2. Ausin, M.S., Maniktala, M., Barnes, T., Chi, M.: Tackling the credit assignment problem in reinforcement learning-induced pedagogical policies with neural networks. In: Roll, I., McNamara, D., Sosnovsky, S., Luckin, R., Dimitrova, V. (eds.) AIED 2021. LNCS (LNAI), vol. 12748, pp. 356–368. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78292-4_29
3. Barnes, T., Stamper, J.: Toward automatic hint generation for logic proof tutoring using historical student data. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 373–382. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69132-7_41

4. Chi, M., VanLehn, K., Litman, D.: Do micro-level tutorial decisions matter: applying reinforcement learning to induce pedagogical tutorial tactics. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 224–234. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13388-6_27
5. De Ayala, R.J.: The Theory and Practice of Item Response Theory. Guilford, New York, NY, USA (2013)
6. Doroudi, S., Aleven, V., Brunskill, E.: Where's the reward? Int. J. AIED **29**(4), 568–620 (2019)
7. Fahid, F.M., Rowe, J.P., Spain, R.D., Goldberg, B.S., Pokorny, R., Lester, J.: Adaptively scaffolding cognitive engagement with batch constrained Deep Q-networks. In: Roll, I., McNamara, D., Sosnovsky, S., Luckin, R., Dimitrova, V. (eds.) AIED 2021. LNCS (LNAI), vol. 12748, pp. 113–124. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78292-4_10
8. Fancsali, S., Murphy, A., Ritter, S.: Closing the loop in educational data science with an open source architecture for large-scale field trials. In: Proceedings of the 15th International Conference on EDM, pp. 834–838. EDM, Durham, UK (July 2022)
9. Georgila, K., Core, M.G., Nye, B.D., Karumbaiah, S., Auerbach, D., Ram, M.: Using reinforcement learning to optimize the policies of an intelligent tutoring system for interpersonal skills training. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pp. 737–745. AAMAS, Richland, SC (2019)
10. Ju, S., Zhou, G., Barnes, T., Chi, M.: Pick the moment: identifying critical pedagogical decisions using long-short term rewards. In: Proceedings of the 13th International Conference on EDM, pp. 126–136. EDM, Virtual (2020)
11. Koedinger, K.R., Aleven, V.: Exploring the assistance dilemma in experiments with cognitive tutors. Educ. Psychol. Rev. **19**(3), 239–264 (2007)
12. Koedinger, K.R., Booth, J.L., Klahr, D.: Instructional complexity and the science to constrain it. Science **342**(6161), 935–937 (2013)
13. Kulik, J.A., Fletcher, J.D.: Effectiveness of intelligent tutoring systems: a meta-analytic review. Rev. Educ. Res. **86**(1), 42–78 (2016)
14. Lattimore, T., Szepesvári, C.: Bandit Algorithms. Cambridge University Press, Cambridge, UK (2020)
15. Li, L., Chu, W., Langford, J., Wang, X.: Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In: Proceedings of the 4th International Conference on WSDM, pp. 297–306. WSDM '11, ACM, New York, NY, USA (2011)
16. McLaren, B.M., Richey, J.E., Nguyen, H., Hou, X.: How instructional context can impact learning with educational technology: Lessons from a study with a digital learning game. Comput. Educ. **178**, 1–20 (2022)
17. Nagashima, T., et al.: How does sustaining and interleaving visual scaffolding help learners? a classroom study with an intelligent tutoring system. In: Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 44 (2022)
18. Nathan, M.J., Koedinger, K.R., Alibali, M.W.: Expert blind spot: When content knowledge eclipses pedagogical content knowledge. In: Proceedings of the 3rd International Conference on Cognitive Science, vol. 3, pp. 644–648. USTC Press, Beijing, China (2001)
19. Ostrow, K., Heffernan, N., Williams, J.J.: Tomorrow's edtech today: establishing a learning platform as a collaborative research tool for sound science. Teach. Coll. Rec. **119**(3), 1–36 (2017)

20. Patikorn, T., Heffernan, N.T.: Effectiveness of crowd-sourcing on-demand assistance from teachers in online learning platforms. In: Proceedings of the 7th International Conference on Learning @ Scale, pp. 115–124. L@S '20, ACM, New York, NY, USA (2020)
21. Prihar, E., Haim, A., Sales, A., Heffernan, N.: Automatic interpretable personalized learning. In: Proceedings of the 9th International Conference on Learning @ Scale, pp. 1–11. L@S '22, ACM, New York, NY, USA (2022)
22. Prihar, E., Patikorn, T., Botelho, A., Sales, A., Heffernan, N.: Toward personalizing students' education with crowdsourced tutoring. In: Proceedings of the 8th International Conference on Learning @ Scale, pp. 37–45. L@S '21, ACM, New York, NY, USA (2021)
23. Reza, M., Kim, J., Bhattacharjee, A., Rafferty, A.N., Williams, J.J.: The MOOClet framework: unifying experimentation, dynamic improvement, and personalization in online courses. In: Proceedings of the 8th ACM Conference on Learning @ Scale, pp. 15–26. L@S '21, ACM, New York, NY, USA (2021)
24. Roschelle, J., Feng, M., Murphy, R.F., Mason, C.A.: Online mathematics homework increases student achievement. AERA Open **2**(4), 1–12 (2016)
25. Spain, R., Rowe, J., Goldberg, B., Pokorny, R., Lester, J., Rockville, M.: Enhancing learning outcomes through adaptive remediation with gift. In: Proceedings of the I/ITSEC, pp. 1–11. I/ITSEC, Orlando, Florida (2019)
26. Spain, R., et al.: A reinforcement learning approach to adaptive remediation in online training. J. Defense Model. Simul. **2**(19), 173–193 (2021)
27. Williams, J.J., et al.: AXIS: generating explanations at scale with learner sourcing and machine learning. In: Proceedings of the 3rd ACM Conference on Learning @ Scale, pp. 379–388. L@S '16, ACM, New York, USA (2016)
28. Williams, J.J., Rafferty, A.N., Tingley, D., Ang, A., Lasecki, W.S., Kim, J.: Enhancing online problems through instructor-centered tools for randomized experiments. In: Proceedings of the CHI Conference on Human Factors in Computing Systems, pp. 1–12. CHI '18, ACM, New York, NY, USA (2018)