Improving Student-System Interaction Through Data-driven Explanations of Hierarchical Reinforcement Learning Induced Pedagogical Policies

Guojing Zhou
gzhou3@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Xi Yang
yxi2@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Hamoon Azizsoltani
h.azizsoltani@gmail.com
SAS
Raleigh, North Carolina, USA

Tiffany Barnes
tmbarnes@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Min Chi
mchi@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

ABSTRACT

Motivated by the recent advances of reinforcement learning and the traditional grounded Self Determination Theory (SDT), we explored the impact of hierarchical reinforcement learning (HRL) induced pedagogical policies and data-driven explanations of the HRL-induced policies on student experience in an Intelligent Tutoring System (ITS). We explored their impacts first independently and then jointly. Overall our results showed that 1) the HRL induced policies could significantly improve students' learning performance, and 2) explaining the tutor's decisions to students through data-driven explanations could improve the student-system interaction in terms of students' engagement and autonomy.

CCS CONCEPTS

• **Human-centered computing → Empirical studies in HCI**.

KEYWORDS

Hierarchical Reinforcement Learning; Pedagogical Policy; Data-driven Explanations; Intelligent Tutoring System

ACM Reference Format:
Guojing Zhou, Xi Yang, Hamoon Azizsoltani, Tiffany Barnes, and Min Chi. 2020. Improving Student-System Interaction Through Data-driven Explanations of Hierarchical Reinforcement Learning Induced Pedagogical Policies. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '20), July 14–17, 2020, Genoa, Italy.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3340631.3394848

1 INTRODUCTION

E-learning environments such as intelligent tutoring systems (ITSs) have become more and more prevalent in educational settings [1, 16]. Most existing ITSs are *tutor-centered* in that *the tutor* decides what to do next in each step when it interacts with the student. For example, the tutor can *elicit* the subsequent step from the student or to *tell* him/her the next step directly. *Pedagogical policies* are used for the tutor to decide what action to take next in the face of alternatives. Each of these decisions affects the student's successive actions and performance. But its impact on student learning often cannot be observed immediately and the effectiveness of one decision often depends on how subsequent decisions are made. Ideally, an effective learning environment should craft and adapt its decisions to users' needs [1, 27]. However, there is no existing well-established theory on how to make these system decisions effectively. On the other hand, Reinforcement Learning (RL) offers one of the most promising data-driven decision-making approaches for improving student learning in ITSs. RL algorithms are designed to induce decision-making policies that specify the best action to take in any given situation so as to maximize a cumulative reward. A number of researchers have studied applying existing RL algorithms to improve the effectiveness of ITSs (e.g. [5, 6, 10, 17, 22, 31, 32, 40, 41, 47, 49, 49]). While promising, this work has three key limitations.

The first limitation is one of granularity. In ITSs, there are many decisions to make at different levels of granularity. For example, we may decide to give a student a problem to solve, to show him/her the next step to take, or to give immediate feedback (e.g., "Good Job!"). All of these actions may have compatible goals but some are more important or impactful than others. Human decision-makers treat these distinct levels of granularity differently and are capable of selecting among them [11, 20]. However, most existing RL approaches treat all decisions equally or independently and do not take into account the long-term impact of higher-level actions. In this work, we use Hierarchical Reinforcement Learning (HRL) to handle decisions at different levels of granularity.

The second limitation is one of interpretability. RL-induced policies are often large, cumbersome, and difficult to understand. For example, RL policies are often represented by complicated computational models that consider a lot of features to make decisions. It is therefore difficult for us to understand how such decisions are made. The opacity raises a major open question: *How can we identify the key features RL used to make pedagogical decisions?* In this work, we used Random Forest (RF) to shed some light on the key features.

UMAP '20, July 14–17, 2020, Genoa, Italy

ACM ISBN 978-1-4503-6861-2/20/07…$15.00
https://doi.org/10.1145/3340631.3394848

The third limitation is one of communication in that the tutor rarely explains to students why certain pedagogical decisions are made. Prior research in Self-Determination Theory (SDT) suggests that explanations could be a powerful tool to enhance student engagement and autonomy during learning. For example, it was shown that explaining the benefits of learning specific knowledge could increase students' sense of control over their own learning [7, 15, 18, 37, 42, 46], which could, in turn, enhance their motivation and engagement [7, 15] and improve their learning experience [37, 46]. However, as far as we know, no prior research has investigated the impact of explaining pedagogical decisions to students. In this work, we generated data-driven explanations based on the key features identified from the HRL-induced policies and investigated how the explanations may impact student-system interaction in terms of engagement and autonomy.

In short, we 1) investigated the effectiveness of HRL induced pedagogical policies that make decisions at different levels of granularity and 2) examined the impact of explaining the tutor's pedagogical decisions using data-driven explanations of the HRL polices. In three classroom studies, we first examined the two factors independently and then jointly. Overall, the results suggest that the HRL-induced policies could improve students' learning performance and data-driven explanations could enhance the student-system interaction in terms of engagement and autonomy.

2 BACKGROUND AND RELATED WORK

2.1 Applying RL to ITSs

Generally speaking, RL approaches can be categorized into online where the agent learns a policy in real-time by interacting with the environment and offline where the agent learns a policy from a pre-collected training corpus. Online applications of RL for pedagogical policy induction often relied on simulations. As a consequence, the success of these approaches depends heavily on the quality or accuracy of the simulation. Many researchers have applied online RL to induce pedagogical policies [4, 13, 28]. Offline RL approaches, on the other hand, "take advantage of previously collected samples and generally provide robust convergence guarantees" [38]. Thus, the success of these approaches depends heavily on the quality of the training data. One common convention is to collect exploratory corpus by training students on ITSs that make random yet reasonable decisions and then induce policies from that corpus. Again, many researchers have applied offline RL to induce pedagogical policies [5, 22, 40, 41]. All of the online and offline models described above were evaluated in classroom studies, yielding improved student learning and/or behaviors as compared to baseline policies.

Despite these successes, the necessity for accurate simulations (online) or large training corpora (offline) has limited the wide use of RL for policy induction. Additionally, most existing applications of RL for pedagogical policy induction has been flat in that all system decisions were treated equally or independently. On the other hand, the tutoring procedure of ITSs can be viewed as a two-loop structure [44]. The outer loop makes problem-level decisions as problem selection while the inner loop makes step-level decisions such as whether or not to give feedback or hint. Motivated by this two-loop structure, in this work we apply hierarchical reinforcement learning (HRL) to induce a hierarchical policy that makes decisions at both the problem (worked example vs. problem solving) and step (elicit vs. tell) levels.

It has been widely shown that HRL can be more effective and data-efficient than flat RL approaches [8, 19, 26, 33, 45]. HRL generally breaks down a large decision-making problem into a hierarchy of small sub-problems and induces a policy for each of them. Since the sub-problems are small, they usually require fewer data to find the optimal policies. Although promising, the use of hierarchy requires additional information, such as the transitions and rewards at different levels of granularity, to induce a policy, and this may be hard to get from pre-collected data. Therefore, most existing HRL applications have been online, but here, we apply an offline HRL approach. Previously, we have applied HRL to induce a hierarchical pedagogical policy. Empirical evaluation results showed that the HRL policy was significantly more effective than a DQN induced flat policy and a random flat policy [47]. However, that study did not explore the impact of explanations.

In terms of interpreting RL policies, previous research has often relied on interpretable RL models [12, 21]. For example, Maes et al. proposed an interpretable RL approach that searches high-performance policies in an interpretable policy space [21]. Our approach differs from previous ones in that it induces the policy using a non-interpretable Gaussian Processes (GP) model, allowing us more flexibility to learn better policies and then, relies on random forest to interpret the policy via identifying the key features.

2.2 The Impact of Explanations on Learning

A lot of SDT research has shown that giving explanations can lead to enhanced engagement [9, 14, 30, 35, 36]. For example, Jang et al. compared the impact of Explanation vs. No-Explanation [14] on learning correlations. They recruited college students in a teacher training program to learn correlation and only told the Explanation condition that learning correlation would help them became more reflective teachers. Results showed that the former were significantly more engaged than the latter, showing more on-task attention, effort, and persistence. Similar impacts were found in other studies [30, 36]. Moreover, Deci et al. examined several factors that may impact the effectiveness of explanations, such as the wording used [9]. Results showed that a low-controlling wording that minimizes pressure and conveys choice can be significantly more effective in enhancing participants' engagement than a high-controlling wording that expresses pressure. More importantly, a low-controlling wording can lead to a *positive correlation* between engagement and the desired learning outcomes. Note that most of prior studies used self-reported survey to measure engagement, while here, we used moment-by-moment student-system interaction logs, which avoid human bias. More importantly, prior research mainly focused on explaining the importance of the task, but our explanations state the reason behind pedagogical decisions and our goal is to see whether these explanations can enhance student-system interaction.

3 PEDAGOGICAL POLICY INDUCTION

Prior research applying RL to induce pedagogical policies often formalize student-system interaction as a Markov Decision Process (MDP). The central idea behind RL approaches is to transform the

problem of inducing effective policies into a computational problem of finding an optimal policy for choosing actions in MDP. An MDP describes a stochastic control process using a 4-tuple $< S, A, T, R >$. In pedagogical policy induction, states S are often represented by a vector composed of relevant learning environment features, such as the percentage of the correct entries a student entered so far and so on. Actions A are the tutor's possible actions, such as elicit or tell. The reward function R is usually calculated from the system's success measures, such as students' learning gain. Once the $< S, A, R >$ has been defined, the transition probability function T is estimated from the training corpus.

Given a defined MDP, we can transform our student-system interaction logs into trajectories as: $s_1 \xrightarrow{a_1,r_1} s_2 \xrightarrow{a_2,r_2} \cdots s_n \xrightarrow{a_n,r_n}$. Here $s_i \xrightarrow{a_i,r_i} s_{i+1}$ indicates that at the ith turn, the learning environment was in state s_i, the tutor executed action a_i and received reward r_i, and then the environment transferred into state s_{i+1}.

Most HRL research is based upon an extension of MDP called discrete Semi-Markov Decision Processes (SMDPs), which adds a set of complex activities [3] or options [43]. Complex activities can invoke other activities recursively, thus allowing the hierarchical policy to function. The *complex* activities are distinct from the primitive actions in that a complex activity may contain multiple *primitive* actions. A complex activity consists of three elements: an initiation set, a termination condition, and a policy π that maps *states to each available option*. A solution to the SMDP mentioned above is an optimal policy (π^*), a mapping from *state to complex activities or primitive actions*, that maximizes the expected discounted cumulative rewards for each state.

Since the complex activities in SMDPs can take a variable number of low-level activity (or actions) to execute across multiple time steps, it is necessary to extend the state-transition function to take into account the activity length. If an activity a in state s takes t' time steps to be executed, then the state transition probability function given s and a is defined by the joint distribution of the result state s' and the number of time steps t' the activity a takes: $P(s', t'|s, a)$. Accordingly, the expected reward function needs to be extended to accumulate over the waiting time t' in s given activity a: $R(s, a, t', s')$.

Similar to RL, HRL learns the policy through estimating the Q-value function $Q(s, a)$, denoted as the expected cumulative rewards the agent will receive if it takes action a in state s and follows the policy to the end. The optimal Q-value function Q^* denotes the expected cumulative rewards the agent can receive if it follows the optimal policy and Q^* satisfies the Bellman equation [43]. In SMDPs, the Bellman equation can be rewritten as:

$$Q(s, a)^* = R(s, a) + \sum_{s',t'} \gamma^{t'} P(s', t'|s, a) \max_{a' \in A} Q(s', a'), \qquad (1)$$

where $0 \leq \gamma \leq 1$ is a discount factor. Once Q^* is calculated, the optimal policy can be easily determined by simply taking the action a with the highest Q value in state s. For HRL, learning occurs at multiple levels. The global learning generates a policy for the top level decisions and local learning generates a policy for each complex activity. This process retains the fundamental assumption of RL: that goals are defined by their association with rewards, and thus that the objective is to discover actions that maximize the long-term cumulative rewards. Local learning focuses not on learning the best policy for the overall task but the best policy for the corresponding complex activity.

In our offline HRL framework, both problem- and step-level policies were learned by recursively using the Gaussian Processes (GP) to estimate the Q-value function [29] following equation 1 until the Q-value function and the policy converge. In each iteration, a Q value was generated for each state-action pair in the training trajectories following equation 1 based on the reward function and the latest GP model. Then the GP model was updated based on the new Q-value assigned to each state-action pair.

Our ITS is a web-based ITS that teaches college probability such as Addition Theorem and Bayes' Theorem. During training, for each problem, the tutor first makes a problem-level decision and then makes step-level decisions based on the problem-level decision. More specifically, the tutor first decides whether the next problem should be worked example (WE), problem solving (PS), or collaborative problem solving (CPS). In WE, students *observe* how the tutor solves a problem; in PS, students *solve* the problem by themselves; while in CPS, students *co-construct* the solution with the tutor. Based on the problem-level decision, the tutor then makes step-level decisions on whether to *elicit* the next solution step from the student or to *tell* or show it to the student directly. We refer to such decisions as elicit/tell. If WE is selected, an all-tell step-level policy will be carried out; if PS is selected, an all-elicit policy will be executed; finally, if CPS is selected, the tutor will decide whether to elicit or to tell a step based on the corresponding step-level policy.

Our training corpus contains 1,147 students' interaction logs collected from training students on the tutor using random (yet reasonable) pedagogical decisions at the problem and step levels. Each student spent around 2 hours on the system and completed around 400 steps. From the logs, we extracted 142 state features to represent the learning environment. More specifically, the features can be grouped into five categories:

- **Autonomy (10 features):** the amount of work done by the student, such as the number of elicits since the last tell *nElicitSinceTell*;
- **Temporal (29):** time related information about the student's behavior, such as the average time per step *avgStepTime*;
- **Problem Solving (35):** information about the current problem solving context, such as problem difficulty *problemDifficulty*;
- **Performance (57):** information about the student's performance so far, such as the percentage of correct entries *pctCorrect*;
- **Hints (11):** information about the student's hint usage, such as the total number of hints requested *nHint*.

Since the primary goal of the ITS is to improve students' learning gains, we used Normalized Learning Gain (NLG) as the reward because it measures students' gain *irrespective of their incoming competence*. $NLG = \frac{posttest - pretest}{\sqrt{1 - pretest}}$ where $pretest$ and $posttest$ refer to students' test scores before and after the ITS training respectively and 1 is the maximum score. To induce the hierarchical policy, we defined a problem-level semi-MDP for determining whether the next problem should be WE, PS, or CPS and for each of the training problems, we defined a step-level semi-MDP for inducing a step-level policy to determine elicit vs. tell if a complex activity CPS was selected for that training problem.

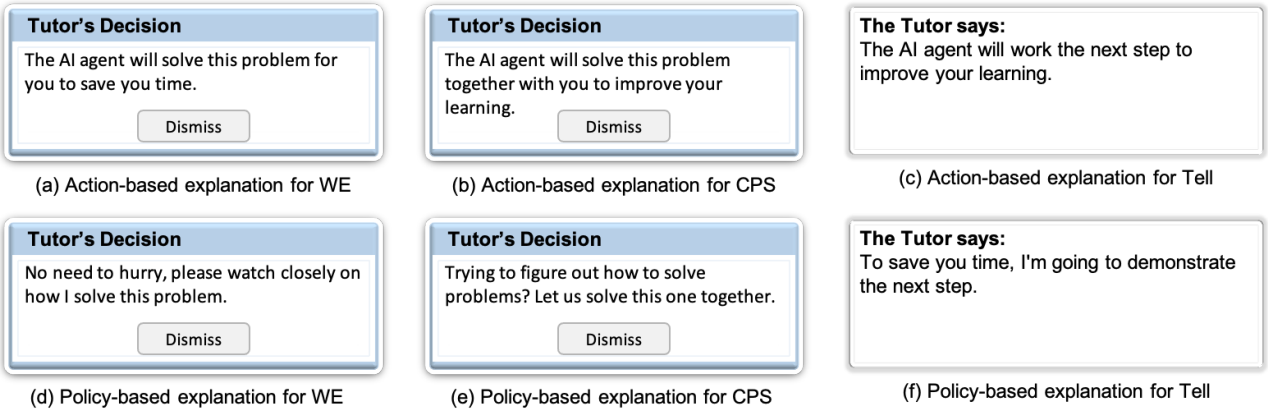

Figure 1: Screenshots of Explanations

4 EXPLAINING TUTOR'S DECISIONS

We explored two types of explanations: *action-based* vs. *policy-based*, as shown in Figure 1. Action-based explanations specify the *benefit of taking a tutorial action* while policy-based explanations give *a reason behind the HRL policy's current decision*. Since students often skip long sentences, we made our explanations short. The explanations were written following Deci, et al.'s design principles, which suggest that effective explanations should present the benefits of a decision, and be written in a way that minimizes pressure and conveys choice [9]. To avoid bothering students by giving too many explanations, we explain only the problem-level WE and CPS decisions and the step-level tell decisions. That's because traditional ITSs are designed to support student problem solving and thus PS and elicit are often considered as the default actions.

4.1 Action-based Explanations

For action-based explanations, we relied on prior cognitive science research to explain the benefit of a tutorial action. For example, a lot of research showed that studying WEs can save students' time [23, 24]. Thus our action-based explanation for WE says " The AI agent will solve the next problem for you to save you time." Similarly, for CPS, it says "The AI agent will solve this problem together with you to improve your learning." as prior research had shown that CPS could be more effective than PS in improving student learning performance [34, 39]. Note that the explanations are written using a low-controlling wording in that it says *what the AI agent will* do, rather than *what the student must* do.

4.2 Policy-based Explanations

The policy-based explanations were written based on the key features of the HRL induced policy, which were identified via mining the HRL policy's decision making examples using random forest (RF). More specifically, we first train an RF classification model based on the HRL policy's decision making examples and then take the key features of the RF as the HRL policy's key features. The examples are in the form of state(142 features)-action pairs and thus the RF takes 142 features as its input and predicts the action.

We chose RF here because it could generate clear, intuitive rule-based classifiers to shed some light on the important features. Specifically, RF is an enssembled model consisting of decision trees. The decision trees provide us with an intuitive way to identify the important features because they always put the important features at higher levels (from the root). To identify the key features of an RF, we first count the occurrence of each feature in the top two levels (from the root) across all decision trees. Then, we pick the most frequent one in each feature category to be the key feature. At the step level, problem-level features were excluded to improve relevance.

When training the RF, we also took the importance of each decision into account, which is determined by Q-values generated by HRL. Recall that the Q-values $Q(s, a)$ indicate the expected cumulative rewards the agent would receive if it takes action a in state s and follows the policy to the end. Thus, for a given state s, the larger the Q-value difference is between two actions, the more important it is for the agent to take the good one. To account for the importance of each decision, we weight each decision making example using the difference between the highest ($max_a Q(s, a)$) and lowest ($min_a Q(s, a)$) Q-value for the state s.

The key features for the problem- and step-level policies were identified separately through training an RF consisting of 70 trees, each with a max depth of 10. All the models were trained using the "sklearn" package [25] and all the results reported below were generated with 5-fold cross-validation. For the problem-level decisions, the RF trained on 12,180 examples yielded an accuracy of 64.9% vs. 44.6% for majority voting. For the step-level decisions, with 450,165 examples, the RF yielded an accuracy of 87.6% vs. 57.1% for majority voting.

At the problem level, the RF identified four key features:

- ***timeOnTutoring:*** the total time that the student has spent on the tutoring task so far;
- ***problemDifficulty:*** the difficulty of the current problem;
- ***avgTimeOnStep:*** the average time the student spent on a step.
- ***nCorrectElicitStepSinceLastWrongKC:*** the number of correct elicit steps the student has completed since last wrong for the knowledge component required by the current step;

At the step level, the RF identified three key features:

- ***avgTimeOnStepSessionTell:*** the average time the student spent on a tell step in the current session;
- ***nCorrectElicitStepSinceLastWrongPrinSession:*** the number of correct elicit step the student has completed since last wrong in the current session for the steps that require selecting or applying a probability principle to solve;
- ***ntellsSinceElicitKC:*** the number of tell steps the student has received since last elicit for the knowledge component required by the current step.

For each of the identified key features, we wrote several explanation sentences to cover all possible decision-reason combinations. Note that since there are multiple key features (for both problem- and step-level decisions), each tutor decision has multiple candidate explanations and the tutor will randomly select one to present. Figure 1.d shows an example explanation for the decision WE, which is generated based on the feature "avgTimeOnStep", Figure 1.e shows an example explanation for CPS based on "nCorrectElicitStepSinceLastWrongKC", and Figure 1.f shows an example explanation for the step level decision *tell* based on "avgTimeOnStepSessionTell".

Note that the explanations present our interpretations of the features instead of their strict definition. That's because some of the features, such as "nCorrectElicitStepSinceLastWrongPrinSession" are too complicated for students without strong educational background to interpret. Thus, to promote student acceptance, the system presents our interpretations. Moreover, to avoid negative wording, we present negative reasons implicitly. For example, in Figure 1.e, the feature indicates that the student is not performing well, but we implicitly hint this reason by asking "Trying to figure out how to solve problems?".

5 EXPERIMENTS SETUP

Participants The participants in our studies were undergraduate students enrolled in the Discrete Mathematics course offered by the computer science department at North Carolina State University in the Fall 2018 or Spring 2019 semester. The ITS was given as one of the regular homework assignment and students had one week to complete it and were graded based upon their demonstrated effort rather than performance. Completion of the ITS was required to earn assignment credit. In the following, we will describe the three empirical studies and also our post-hoc analysis in detail. Note that all studies used the same baseline (control) condition: step-level random with no explanation. We chose step-level decisions as the baseline because problem-level WEs and PS can be seen as two extreme cases of CPS, and thus a non-hierarchical way to make decisions would be to focus on the step level only. Since both elicit and tell are always considered to be reasonable interventions in our learning context, the baseline random policy is "random yet reasonable".

- **The HRL Study**

Research Question: Would the HRL induced pedagogical policy improve students' learning performance?
Conditions: HRL vs. Baseline;
Measurements: learning performance measured by NLG;
Hypothesis: $HRL > Baseline$ on NLG.

- **The Exp Study**

Research Question: would explaining the tutor's decisions enhance the student-system interaction in terms of autonomy and engagement?
Conditions: Exp vs. Baseline. The Exp condition employed action-based explanations and a random hierarchical policy which first randomly decides whether the next problem should be WE, PS, or CPS, and then if CPS is selected, it randomly decides whether to elicit or tell each step. Here, we chose the hierarchical random policy for the Exp condition because it allows us to explain both the problem- and step-level decisions. Note that, a series of our prior studies suggest that the two types of random policies (step-level vs. hierarchical) do not result in significant differences in student learning [48, 50], allowing us to examine the impact of explanations across both policies.
Measurements: Based on previous SDT research, we expect that giving explanations would improve student-system interaction in terms of autonomy and engagement but may or may not improve student learning performance. Student autonomy is measured by: *hints per elicit*, which is calculated as the number of hints requested on *elicit* steps during the training divided by the total number of elicit steps. This measure indicates to what extent students relied on their own knowledge to complete problems. For engagement, we used *percentage of correct entries*, which is defined as the percentage of correct entries the student made on the first attempt in elicit steps. Engagement is defined in different ways in learning science literature. A commonly used definition is the time and effort students devote to the task, which is also suggested by Next Generation Science Standards (NGSS). Prior research suggests that disengaged students often "game the system" to finish quickly rather than to actually learn the material [2]. As a result, disengaged students should have both lower time on task and performance during training. To verify that these two measures are correlated, we performed an analysis and found a significant correlation ($r = 0.26$; $p = 0.0008$) between a time on task indicator (the average time students spent on each wrong step) and their training performance (the percentage of correct elicit entries). Since the latter reflects both time and effort, we chose it to measure student engagement.
Hypothesis: Exp would ask for less hint per elicit and have a higher percentage of correct entries than Baseline.

- **The HRL+Exp Study**

Research Question: would the HRL policy and policy-based explanations together improve both students' learning performance and student-system interaction?
Conditions: HRL+Exp vs. Baseline;
Measurements: NLG, hints per elicit, and percentage of correct entries;
Hypothesis: $HRL + Exp > Baseline$ on NLG and student-system interaction.

- **Post-hoc Analysis**

Research Questions: how would the HRL policy and the explanations impact students' learning independently and jointly?
Conditions: HRL, Exp, HRL+Exp, Baseline;
Measurements: NLG, hints per elicit, and percentage of correct entries;
Hypothesis: $HRL, HRL + Exp > Exp, Baseline$ on NLG while $HRL + Exp, Exp > HRL, Baseline$ on student-system interaction.

Procedure All students went through the identical four phases: 1) textbook, 2) pre-test, 3) training on ITS, and 4) post-test. During **textbook**, all students read a general description of each principle, reviewed some examples, and solved some practice problems. The students then took a **pre-test** which contained a total of 14 problems. No feedback was given to their answers and they were not allowed to go back to earlier questions (this was also true for the post-test). During **training on the ITS**, all students received *the same 12 problems in the same order*. Each domain principle was applied at least twice in the 12 problems and each of the problems required 20-50 steps to solve. Finally, all students took the 20-problem **post-test**: 14 of them were isomorphic to the pre-test, and the remainder were non-isomorphic multiple-principle problems. Conditions differed only in the pedagogical policy used (HRL vs. Baseline) and the explanations of the tutor's decision (No-Explanation vs. Explanation) in the training phase.

Grading Criteria The pre- and post-test problems required students to derive an answer by writing and solving one or more equations. We used three scoring rubrics: binary, partial credit, and one-point-per-principle. Under the binary rubric, a solution was worth 1 point if it was completely correct or 0 if not. Under the partial credit rubric, each problem score was defined by the proportion of correct principle applications evident in the solution. A student who correctly applied 4 of 5 possible principles would get a score of 0.8. The one-point-per-principle rubric in turn gave a point for each correct principle application. All of the tests were graded in a double-blind manner by a single experienced grader. The results presented below were based upon the partial-credit rubric but the same results hold for the other two. For comparison purposes, all test scores were normalized to the range of $[0, 100]$.

6 RESULTS

6.1 The HRL Study

120 students were randomly assigned to the HRL (N = 60) and the Baseline (N = 60) conditions [1]. Due to preparations for exams and the length of the study, 92 students completed the training. 11 students who performed perfectly in the pre-test, completed the study in groups, or quickly rushed through the pre- or post-test with minimal and wrong answers were excluded from our subsequent analysis. The remaining 81 students were distributed as follows: 44 for the HRL condition and 37 for the Baseline condition. A χ^2 test showed that the participants' completion rate did not significantly differ by condition: $\chi^2(1, N = 120) = 0.419, p = 0.517$. A t-test analysis on the pre-test score showed no significant difference between the HRL ($M = 66.4$, $SD = 18.8$) and the Baseline condition ($M = 68.5$, $SD = 16.6$): $t(79) = -0.55$, $p = .587$, $d = 0.12$. This suggested that the two conditions were balanced in incoming competence.

Learning Performance and Training Time To measure students' learning improvement, we conducted a repeated measures analysis using test type (pre-test vs. isomorphic post-test) as a factor and test score as the dependent measure. Results showed a main effect for test type: $F(1, 79) = 141.40$, $p < 0.0001$, $\eta = 0.635$ in that students scored significantly higher in the isomorphic post-test than in the pre-test. More specifically, both conditions scored significantly higher in the post-test than in the pre-test: $F(1, 43) = 54.10$, $p < 0.0001$, $\eta = 0.557$ for the HRL condition and $F(1, 36) = 31.40$, $p < 0.0001$, $\eta = 0.466$ for the Baseline condition. This showed that the basic practice and problems, domain exposure, and interactivity of our ITS could effectively help students acquire knowledge, even when the decisions were made randomly yet reasonably.

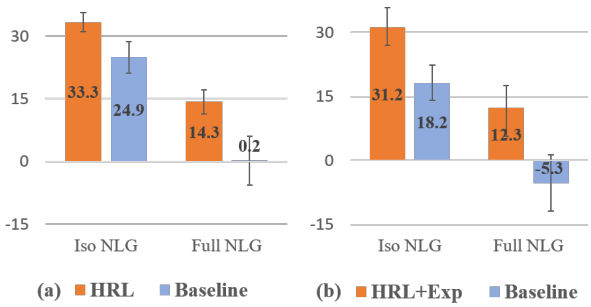

Figure 2: Iso NLG and Full NLG Results

Figure 2.a shows a comparison of students' learning gains between the two conditions. The Iso NLG was calculated based on the pre- and isomorphic post-test while the Full NLG was calculated based on the pre- and full post-test. The full post-test had six additional multiple-principle problems. A t-test on the Iso NLG showed that there was a trend that the HRL condition ($M = 33.3$, $SD = 15.4$) scored higher than the Baseline condition ($M = 24.9$, $SD = 23.0$): $t(79) = 1.96$, $p = .053$, $d = 0.44$. In addition, a t-test on the Full NLG showed that the HRL condition ($M = 14.3$, $SD = 19.2$) scored significantly higher than the Baseline condition ($M = 0.2$, $SD = 35.9$): $t(79) = 2.24$, $p = .028$, $d = 0.50$. Overall, the results suggested that the HRL policy was more effective than the baseline policy. Finally, a t-test on the total training time showed no significant difference between the two conditions.

Student-system Interaction As expected, t-tests on the hints per elicit and percentage of correct entries showed no significant differences between the HRL and the Baseline conditions.

6.2 The Exp Study

114 students were randomly assigned to the Exp (N = 57) and Baseline (N = 57) conditions and 94 students completed the study. 9 students were excluded from our subsequent analysis for perfect pre-test, group work or rush behavior. The remaining 85 students were distributed as follows: 42 for Exp and 43 for Baseline. A χ^2 test showed that the participants' completion rate did not significantly differ by condition: $\chi^2(1, N = 114) = 0.061, p = 0.81$. A t-test on the pre-test score showed that there was a trend that the Baseline condition ($M = 75.9$, $SD = 18.6$) scored higher than the Exp condition ($M = 69.1$, $SD = 17.9$): $t(83) = 1.72$, $p = .089$, $d = 0.37$. This suggests that our random assignment did not balance students' incoming competence perfectly. Therefore, for learning performance, we mainly focused on Iso NLG and Full NLG because they consider the pre-test differences.

[1] Since the three studies were conducted in the same class but in two different semesters (Fall 2018 or Spring 2019), the size of the conditions in them differed slightly.

Learning Performance and Training Time As expected, we found that while both conditions learned significantly from our ITS and there was no significant difference between the Exp and Baseline conditions on learning performance (Iso NLG and Full NLG) and training time.

Student-system Interaction For this study, we expected that the Exp condition would request less hints per elicit and have a higher percentage of correct entries than the Baseline condition. However, no significant difference was found between them on hints per elicit ($M = .165$, $SD = .676$ for Exp and $M = .2$, $SD = .555$ for Baseline): $t(83) = 0.62$, $p = .534$ and percentage of correct entries ($M = 80.5$, $SD = 11.4$ for Exp and $M = 80.6$, $SD = 11.9$ for Baseline): $t(83) = 0.05$, $p = .964$. (Since hints per elicit has a large variance, its p values were calculated based on the log transformation of the original value here and in the following.) There are two possible reasons for the results: 1) the pedagogical decisions are randomly made rather than adaptively and 2) the action-based explanations are not helpful. Therefore, in the HRL+Exp study, we employed HRL induced policy and data-driven explanations.

6.3 The HRL+Exp Study

113 students were randomly assigned to the HRL+Exp (N = 56) and Baseline (N = 57) conditions and 91 of them completed the study. 10 students were excluded from our subsequent analysis for prefect pre-test, group work or rash behavior. The remaining 80 students were distributed as follows: 37 for HRL+Exp and 43 for Baseline. A χ^2 test showed that the participants' completion rate did not significantly differ by condition: $\chi^2(1, N = 113) = 0.675, p = 0.448$. A t-test on the pre-test score showed that there was no significant difference between the HRL+Exp ($M = 71.6$, $SD = 21.2$) condition and the Baseline condition ($M = 75.9$, $SD = 18.6$): $t(78) = -0.97$, $p = .336$, $d = 0.22$.

Learning Performance and Training Time Repeated measures analysis showed that overall students scored significant higher in the isomorphic post-test and in the pre-test: $F(1, 78) = 47.54$, $p < 0.0001$, $\eta = 0.370$. This was also true for each individual condition. Figure 2.b shows a comparison between the two conditions on Iso NLG and Full NLG. A t-test on the Iso NLG showed that the HRL+Exp ($M = 31.2$, $SD = 28.3$) condition scored significantly higher than the Baseline condition ($M = 18.2$, $SD = 27.1$): $t(78) = 2.10$, $p = .039$, $d = 0.47$. Similarly, on Full NLG, HRL+Exp ($M = 12.3$, $SD = 32.2$) significantly outperformed Baseline ($M = -5.3$, $SD = 43.2$): $t(78) = 2.04$, $p = .045$, $d = 0.46$. The results suggest that the combination of the HRL policy and data-driven explanations could effectively improve students' learning performance. A t-test on the total training time showed that there was no significant difference between the two conditions.

Student-system interaction We expected that the HRL+Exp condition would have higher percentage of correct entries and request less hints per elicit than the Baseline condition. A t-test on the percentage of correct entries showed that HRL+Exp ($M = 86.5$, $SD = 6.2$) indeed scored significantly higher than Baseline ($M = 80.6$, $SD = 11.9$): $t(78) = 2.72$, $p = .008$, $d = 0.61$. Additionally, a t-test on hints per elicit showed that there was a trend that HRL+Exp ($M = .044$, $SD = .071$) requested less hints than Baseline ($M = .2$, $SD = .555$): $t(78) = -1.87$, $p = .066$, $d = 0.42$.

6.4 Post-hoc Analysis

In order to comprehensively evaluate the impact of HRL policies and explanations on learning, we conducted a two-factor post-hoc analysis on policy (HRL vs. baseline) and explanation (Explanation vs. No-Explanation) using the data collected in the three empirical classroom studies. Four conditions are included in the analysis, HRL+Exp, HRL, Exp, and Baseline. To balance the group size among the four conditions, we randomly sampled 40 students from the combined Baseline group of 80 students. Among them, 19 students were selected from the Fall 2018 semester and 21 students were selected from the Spring 2019 semester. A χ^2 test showed that the selection rate did not significantly differ by semester: $\chi^2(1, N = 80) = 0$, $p = 1$. T-tests showed that there was no significant difference between the 40 sampled and the 40 unsampled students on test scores, training time, and all other measures. This was also true between the sampled and the unsampled students for the Fall 2018 and Spring 2019 semesters.

One-way ANOVA analysis on the pre-test score showed that there was no significant difference among the four conditions ($M = 71.6$, $SD = 21.2$ for HRL+Exp; $M = 66.4$, $SD = 18.8$ for HRL; $M = 69.1$, $SD = 17.9$ for Exp; $M = 71.6$, $SD = 18.5$ for Baseline): $F(3, 159) = 0.72$, $p = 0.540$, $\eta = 0.013$. This suggested that the four conditions were balanced in incoming competence.

Learning Performance A repeated measures analysis showed that overall students scored significantly higher in the isomorphic post-test than in the pre-test: $F(1, 159) = 168.12$, $p < 0.0001$, $\eta = 0.508$. This was also true for each individual condition. Figure 3.a shows the Full NLG for the four conditions. A two-way ANOVA analysis on policy and explanation showed a significant main effect of policy: $F(1, 159) = 5.59$, $p = 0.019$, $\eta = 0.034$ in that the two HRL conditions scored significantly higher than the Baseline and the Exp conditions. But there was no significant interaction effect or main effect of explanation. Subsequent contrast analysis showed that the HRL condition ($M = 14.3$, $SD = 19.2$) scored significantly higher than the Baseline condition ($M = -1.4$, $SD = 39.3$): $t(159) = 2.35$, $p = 0.020$, $d = 0.52$. This suggests again that the HRL policy was more effective than the Baseline policy. Similar results were found for Iso NLG.

Training Time (on the system) results are shown in Figure 3.b. A two-way ANOVA analysis showed a significant main effect of explanation: $F(1, 159) = 12.40$, $p = 0.0006$, $\eta = 0.068$ in that the two Explanation conditions spent less time than the two No-Explanation conditions and also a significant main effect of policy: $F(1, 159) = 11.38$, $p = 0.0009$, $\eta = 0.062$ in that the two HRL conditions spent more time than the Baseline and the Exp conditions. No significant interaction effect was found. Subsequent contrast analysis revealed that for the effect of explanation, the HRL+Exp condition ($M = 1.89$, $SD = .48$) spent significantly less time than the HRL condition ($M = 2.19$, $SD = .64$): $t(159) = -2.72$, $p = 0.007$, $d = 0.53$; and the Exp condition ($M = 1.67$, $SD = .43$) spent significantly less time than the Baseline condition ($M = 1.92$, $SD = .41$): $t(159) = -2.26$, $p = 0.025$, $d = 0.60$. For the effect of policy, the HRL condition ($M = 2.19$, $SD = .64$) spent significantly more time than the Baseline condition ($M = 1.92$, $SD = .41$): $t(159) = 2.51$, $p = 0.013$, $d = 0.51$; and the HRL+Exp condition ($M = 1.89$, $SD = .48$) tended to spend more time than the Exp condition ($M = 1.67$, $SD = .43$): $t(159) = 1.96$,

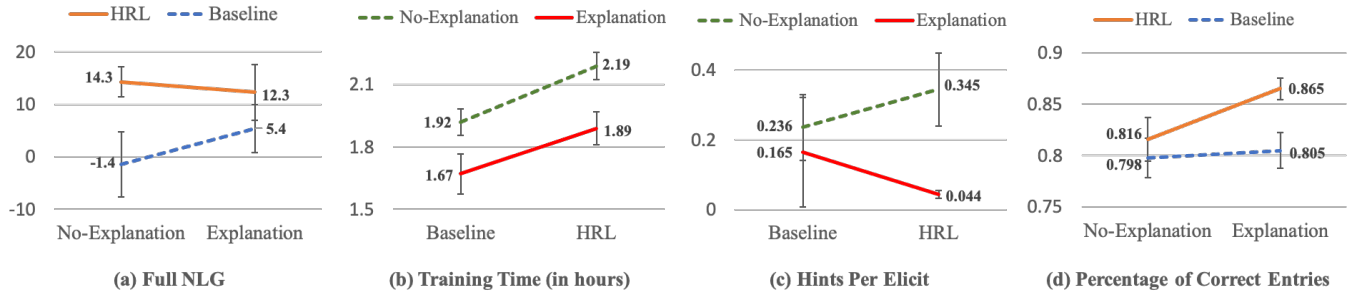

Figure 3: Comparisons on policy (HRL vs. Baseline) and Explanation (No-Explanation vs. Explanation) for a) Full NLG; b) Total time on the ITS training task (in hours); c) Hints Per Elicit; and d) Percentage of Correct Entries.

$p = 0.051$, $d = 0.49$. Overall, the results suggest that explaining the tutor's decision could make students work more efficiently.

Student-system Interaction We expected that the two Explanation conditions would have higher percentage of correct entries and request less hints per elicit than the two No-Explanation conditions. Figure 3.c shows the number of hints requested per elicit. A two-way ANOVA analysis showed a main effect of explanation: $F(1, 159) = 4.49$, $p = 0.036$, $\eta = 0.027$ in that the two Explanation conditions requested significantly less hints per elicit than the two No-Explanation conditions. But there was no significant interaction effect or main effect of policy. Subsequent contrast analysis revealed that the HRL+Exp condition ($M = .044$, $SD = .071$) requested significantly less hints than the HRL condition ($M = .345$, $SD = 1.04$): $t(159) = -2.11$, $p = 0.037$, $d = 0.47$. To investigate whether difficulty may have been a factor in increased hint requests, we partitioned our data by frequent and infrequent hint users and found that there is no significant difference between the two groups in terms of the difficulty of the elicit steps they received. ($M = 29.45$, $SD = 0.80$ for frequent hint users and $M = 29.66$; $SD = 0.73$ for infrequent hint users): $t(161) = -1.74$, $p = .084$, $d = 0.27$.

Figure 3.d shows the percentage of correct entries. A two-way ANOVA analysis showed a main effect of policy: $F(1, 159) = 4.13$, $p = 0.044$, $\eta = 0.025$ in that the two HRL conditions scored significantly higher than the Baseline and the Exp conditions. But there was no significant interaction effect or main effect of explanation. Subsequent contrast analysis revealed that the HRL+Exp condition ($M = 86.5$, $SD = 6.2$) scored significantly higher than the Exp condition ($M = 80.5$, $SD = 11.4$): $t(159) = 2.32$, $p = 0.021$, $d = 0.64$ and tended to score higher than the HRL condition ($M = 81.6$, $SD = 14.1$): $t(159) = -1.93$, $p = 0.055$, $d = 0.44$. In order to examine whether enhanced student-system interaction led to improved learning performance, we performed Pearson's correlation tests between percentage of correct entries and Full NLG. Results showed that only the HRL+Exp condition had a significant correlation: $r = 0.517$, $p = 0.001$. This is consistent with prior research that effective explanations can lead to a positive correlation between engagement and the desired learning outcome [9]. Overall the results suggest that the HRL policy together with data-driven explanations could enhance students' engagement and autonomy.

7 CONCLUSION AND DISCUSSION

This work demonstrates how reinforcement learning (RL) can be productively applied to improve student-system interaction. Specifically, our work proposes a novel but replicable combination of cutting-edge RL (for adaptation) and human authored explanations (for interpretability) to improve student-system interaction. Empirical results show that personalized RL decisions can be paired with human-authored explanations to achieve improved student-system interaction outcomes, rather than using RL decisions or explanations alone. In recent years, RL, especially Deep RL, has achieved superhuman performance in several complex games. However, different from the classic game-play situations where the ultimate goal is to make smart system decisions, our ultimate goal for RL is to make the student-system interaction productive and fruitful. Thus, we argue that it is crucial to communicate the RL decisions to students. By using intelligent tutoring systems (ITSs) as the testbed, we show that neither RL decisions nor explanations alone promote ITS goals. Our results suggest that future e-learning applications can combine RL with human-authored explanations to significantly influence how students and systems interact.

One limitation of this work is explanations were not validated in a separate study. More specifically, it is unknown that how students may interpret the messages and whether it is important for them to benefit from the explanations. The mixed results in where explanations are helpful (only for HRL) make it especially important to further understand the nuance in when/why explanations are helpful. In the future, we will add a survey in the study to explore user perception and experience. Additionally, we will compare our data-driven explanations with learning-theory-based explanations to further explore their impacts. Another limitation is that the four conditions were investigated in three studies instead of one study that provides a more sensitive evaluation. Finally, a third limitation is we used percentage of correct entries to measure students' engagement, which could be impacted by the pedagogical policy.

Acknowledgements This research was supported by the NSF Grants: #1726550, #1651909, and #1660878.

REFERENCES

[1] John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1995. Cognitive tutors: Lessons learned. *The journal of the learning sciences* 4, 2 (1995), 167–207.

[2] Ryan Shaun Baker, Albert T Corbett, Kenneth R Koedinger, and Angela Z Wagner. 2004. Off-task behavior in the cognitive tutor classroom: when students" game the system". In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 383–390.

[3] Andrew G Barto and Sridhar Mahadevan. 2003. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems* 13, 1-2 (2003), 41–77.

[4] Joseph Beck, Beverly Park Woolf, and Carole R Beal. 2000. ADVISOR: A machine learning architecture for intelligent tutor construction. *AAAI/IAAI* 2000, 552-557 (2000), 1–2.

[5] Min Chi, Kurt VanLehn, Diane Litman, and Pamela Jordan. 2011. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction* 21, 1-2 (2011), 137–180.

[6] Min Chi, Kurt VanLehn, Diane Litman, and Pamela Jordan. 2011. An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *International Journal of Artificial Intelligence in Education* 21, 1-2 (2011), 83–113.

[7] Diana I Cordova and Mark R Lepper. 1996. Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of educational psychology* 88, 4 (1996), 715.

[8] Heriberto Cuayáhuitl et al. 2010. Generating adaptive route instructions using hierarchical reinforcement learning. In *International Conference on Spatial Cognition*. Springer, 319–334.

[9] Edward L Deci, Haleh Eghrari, Brian C Patrick, and Dean R Leone. 1994. Facilitating internalization: The self-determination theory perspective. *Journal of personality* 62, 1 (1994), 119–142.

[10] Shayan Doroudi, Kenneth Holstein, Vincent Aleven, and Emma Brunskill. 2015. Towards Understanding How to Leverage Sense-Making, Induction and Refinement, and Fluency to Improve Robust Learning. *International Educational Data Mining Society* (2015).

[11] Martha Evens and Joel Michael. 2006. *One-on-one tutoring by humans and computers*. Psychology Press.

[12] Daniel Hein, Steffen Udluft, and Thomas A Runkler. 2018. Interpretable policies for reinforcement learning by genetic programming. *Engineering Applications of Artificial Intelligence* 76 (2018), 158–169.

[13] Ana Iglesias, Paloma Martínez, Ricardo Aler, and Fernando Fernández. 2009. Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems. *Knowledge-Based Systems* 22, 4 (2009), 266–270.

[14] Hyungshim Jang. 2008. Supporting students' motivation, engagement, and learning during an uninteresting activity. *Journal of Educational Psychology* 100, 4 (2008), 798.

[15] Mable B Kinzie and Howard J Sullivan. 1989. Continuing motivation, learner control, and CAI. *Educational Technology Research and Development* 37, 2 (1989), 5–14.

[16] Kenneth R Koedinger, John R Anderson, William H Hadley, and Mary A Mark. 1997. Intelligent tutoring goes to school in the big city. (1997).

[17] Kenneth R Koedinger, Emma Brunskill, Ryan SJd Baker, Elizabeth A McLaughlin, and John Stamper. 2013. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine* 34, 3 (2013), 27–41.

[18] Alfie Kohn. 1993. Choices for children. *Phi Delta Kappan* 75, 1 (1993), 8–20.

[19] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*. 3675–3683.

[20] Mark R Lepper, Maria Woolverton, Donna L Mumme, and J Gurtner. 1993. Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. *Computers as cognitive tools* 1993 (1993), 75–105.

[21] Francis Maes, Raphael Fonteneau, Louis Wehenkel, and Damien Ernst. 2012. Policy search in a space of simple closed-form formulas: Towards interpretability of reinforcement learning. In *International Conference on Discovery Science*. Springer, 37–51.

[22] Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. 2014. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1077–1084.

[23] Bruce M McLaren and Seiji Isotani. 2011. When is it best to learn with all worked examples?. In *International Conference on Artificial Intelligence in Education*. Springer, 222–229.

[24] Bruce M McLaren, Sung-Joo Lim, and Kenneth R Koedinger. 2008. When and how often should worked examples be given to students? New results and a summary of the current state of research. In *Proceedings of the 30th annual conference of the cognitive science society*. 2176–2181.

[25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[26] Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel Van De Panne. 2017. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 41.

[27] Pipatsarun Phobun and Jiracha Vicheanpanya. 2010. Adaptive intelligent tutoring systems for e-learning systems. *Procedia - Social and Behavioral Sciences* 2, 2 (2010), 4064 – 4069. https://doi.org/DOI:10.1016/j.sbspro.2010.03.641 Innovation and Creativity in Education.

[28] Anna N Rafferty, Emma Brunskill, Thomas L Griffiths, and Patrick Shafto. 2016. Faster teaching via pomdp planning. *Cognitive science* 40, 6 (2016), 1290–1332.

[29] Carl Edward Rasmussen. 2004. Gaussian processes in machine learning. In *Advanced lectures on machine learning*. Springer, 63–71.

[30] Johnmarshall Reeve, Hyungshim Jang, Pat Hardre, and Mafumi Omura. 2002. Providing a rationale in an autonomy-supportive way as a strategy to motivate others during an uninteresting activity. *Motivation and emotion* 26, 3 (2002), 183–207.

[31] Jonathan Rowe, Bradford Mott, and James Lester. 2014. Optimizing player experience in interactive narrative planning: a modular reinforcement learning approach. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

[32] Jonathan P Rowe and James C Lester. 2015. Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework. In *International Conference on Artificial Intelligence in Education*. Springer, 419–428.

[33] Malcolm Ryan and Mark Reid. 2000. Learning to fly: An application of hierarchical reinforcement learning. In *In Proceedings of the 17th International Conference on Machine Learning*. Citeseer.

[34] Ron JCM Salden, Vincent Aleven, Rolf Schwonke, and Alexander Renkl. 2010. The expertise reversal effect and worked examples in tutored problem solving. *Instructional Science* 38, 3 (2010), 289–307.

[35] Carol Sansone, Charlene Weir, Lora Harpster, and Carolyn Morgan. 1992. Once a boring task always a boring task? Interest as a self-regulatory mechanism. *Journal of personality and social psychology* 63, 3 (1992), 379.

[36] Carol Sansone, Deborah J Wiebe, and Carolyn Morgan. 1999. Self-regulating interest: The moderating role of hardiness and conscientiousness. *Journal of personality* 67, 4 (1999), 701–733.

[37] Gregory Schraw, Terri Flowerday, and Marcy F Reisetter. 1998. The role of choice in reader engagement. *Journal of Educational Psychology* 90, 4 (1998), 705.

[38] Devin Schwab and Soumya Ray. 2017. Offline reinforcement learning with task hierarchies. *Machine Learning* 106, 9-10 (2017), 1569–1598.

[39] Rolf Schwonke, Alexander Renkl, Carmen Krieg, Jörg Wittwer, Vincent Aleven, and Ron Salden. 2009. The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior* 25, 2 (2009), 258–266.

[40] Shitian Shen and Min Chi. 2016. Aim Low: Correlation-Based Feature Selection for Model-Based Reinforcement Learning. *International Educational Data Mining Society* (2016).

[41] Shitian Shen and Min Chi. 2016. Reinforcement Learning: the Sooner the Better, or the Later the Better?. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. ACM, 37–44.

[42] Hsin-Yih Shyu and Scott W Brown. 1992. Learner control versus program control in interactive videodisc instruction: What are the effects in procedural learning. *International Journal of Instructional Media* 19, 2 (1992), 85–95.

[43] Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112, 1-2 (1999), 181–211.

[44] Kurt Vanlehn. 2006. The behavior of tutoring systems. *International journal of artificial intelligence in education* 16, 3 (2006), 227–265.

[45] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. 2018. Video captioning via hierarchical reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4213–4222.

[46] Shiou-Wen Yeh and James D Lehman. 2001. Effects of learner control and learning strategies on English as a foreign language (EFL) learning from interactive hypermedia lessons. *Journal of Educational Multimedia and Hypermedia* 10, 2 (2001), 141–159.

[47] Guojing Zhou, Hamoon Azizsoltani, Markel Sanz Ausin, Tiffany Barnes, and Min Chi. 2019. Hierarchical reinforcement learning for pedagogical policy induction. In *Artificial Intelligence in Education - 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part I*. Springer, 544–556.

[48] Guojing Zhou, Thomas W Price, Collin Lynch, Tiffany Barnes, and Min Chi. 2015. The Impact of Granularity on Worked Examples and Problem Solving. In *Proceedings of the 37th annual conference of the cognitive science society*. 2817–2822.

[49] Guojing Zhou, Jianxun Wang, Collin Lynch, and Min Chi. 2017. Towards Closing the Loop: Bridging Machine-induced Pedagogical Policies to Learning Theories. In *EDM*.

[50] Guojing Zhou, Xi Yang, and Min Chi. 2019. Big, Little, or Both? Exploring the Impact of Granularity on Learning for Students with Different Incoming Competence. In *Proceedings of the 41th annual conference of the cognitive science society*. 3206–3212.