Student-Tutor Mixed-Initiative Decision-making Supported by Deep Reinforcement Learning

Song Ju, Xi Yang, Tiffany Barnes, and Min Chi

Department of Computer Science
North Carolina State University, Raleigh, NC 27695, USA
{sju2,yxi2,tmbarnes,mchi}@ncsu.edu

Abstract. One fundamental goal of education is to enable students to *act independently* in the world by continuously adapting and learning. Certain learners are less sensitive to learning environments and can always perform well, while others are more sensitive to variations in learning environments and may fail to learn. We refer to the former as *high performers* and the latter as *low performers*. Previous research showed that low performers benefit more from tutor-driven Intelligent Tutoring Systems (ITSs), in which the tutor makes pedagogical decisions, while the high ones often prefer to take control of their own learning by making decisions by themselves. We propose a *student-tutor mixed-initiative (ST-MI)* decision-making framework which balances allowing students some control over their own learning while ensuring effective pedagogical interventions. In an empirical study, ST-MI significantly improved student learning gains than an Expert-designed, tutor-driven pedagogical policy on an ITS. Furthermore, our ST-MI framework was found to offer low performers *the same benefits as* the Expert policy, while that for high performers was *significantly greater* than the Expert policy.

Keywords: Critical Decisions · Reinforcement Learning · Student Choice.

1 Introduction

One fundamental purpose of education is to enable students to act independently in the world — to make good decisions and to adapt and continue to learn. On one hand, students who are more actively involved in deciding what and how to learn will benefit from the sense of control, such as becoming more engaged, motivated, and persistent [4, 9, 6]. On the other hand, not all students are adept at making decisions. Prior research has shown that *low performing learners* may not always have the necessary metacognitive skills to make effective pedagogical decisions [1, 19]. As a result, most Intelligent Tutoring Systems (ITSs) are tutor-driven in that *the tutor* decides what to do in the next step. For example, the tutor can *elicit* the subsequent step from the student, either with prompting and support or without. When a student enters a step, the ITS records its success or failure and may give feedback (e.g., correct/incorrect markings) and/or hints. Alternatively, the tutor can choose to *tell* them the next step directly, or provide a partially-worked step [11]. Each of these decisions affects the student's successive actions and performance. *Pedagogical policies* are used for the agent (i.e., tutor) to decide what action to take next among several alternatives.

In this work, we present a generalizable ***student-tutor mixed-initiative (ST-MI) decision-making*** framework which balances allowing students some control over their own learning while ensuring effective pedagogical interventions. More specifically, our framework is supported by a general Critical Deep Reinforcement Learning (Critical-DRL) approach, which uses Long-Short Term Rewards (LSTRs) and Critical Deep Q-Network (Critical-DQN). In the ST-MI framework, the tutor would take over decision-making ***only when students fail to make the optimal choice at critical moments***.

Figure 1 illustrates that our ST-MI framework consists of two loops with two agents: a student agent (SA) and a pedagogical agent (PA). The SA interacts with the environment in the inner loop (dashed area in Figure 1), whereas the PA interacts with the inner loop in the outer loop. Here the SA is the front-end decision-maker, and our PA is the back-end. *If the SA makes a sub-optimal choice on a critical decision, the PA will intervene by taking an alternative choice and explain why it is better; Otherwise, the SA's decision is carried out.* To identify critical decisions, we proposed and developed a *Critical-DRL* approach using Long-Short Term Rewards and Critical Deep Q-Network described in 3.1.

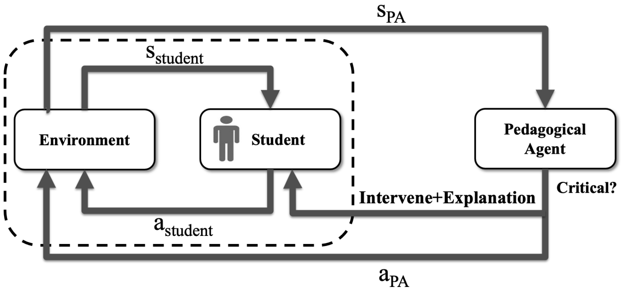

Fig. 1. Our ST-MI Decision-making Framework

The effectiveness of the ST-MI framework is empirically compared against an Expert-designed policy, referred to as the *Expert policy*, where the tutor makes all pedagogical decisions. In this study, we focused on the decisions on whether to present the next problem as a Worked Example (WE), a Problem Solving (PS), or a faded worked example (FWE). In WE, students were given a detailed example showing how the tutor solves a problem; in PS, by contrast, students were tasked with solving the same problem on their own on the ITS; in FWEs, the students and the tutor *co-construct* in that their solutions are intertwined. Our results showed that the ST-MI students achieved significantly higher learning gains than the Expert peers. Further, we separated students based on their incoming competencies, i.e., pretest scores, and examined the impact of the ST-MI framework on the Aptitude-Treatment Interaction (ATI). For low incoming competence students, in particular, prior research has shown that they are less likely to benefit from making pedagogical decisions on their own [30], our ST-MI framework was found to offer the low performers the same benefits as the Expert policy. While previous research has shown that high incoming competence students are just as effective at learning as those who make their own decisions or follow the Expert policy [30], our findings showed that the ST-MI framework can significantly improve their learning over the Expert policy.

2 Related Work

Applying RL to ITSs: In ITSs, the student-agent interactions can be described as sequential decision-making problems under uncertainty, which can be formulated as problems of RL, a learning paradigm that depends on long-term rewards without knowing the "correct" decisions at the immediate time-steps [24]. An increasing number of prior research has explored the use of RL and Deep RL (DRL) to ITSs (e.g. [7, 10, 20]) and specifically, it has showed that they can be used to induce effective pedagogical policies for ITSs [10, 27]. For example, Shen et al. [22] utilized value iteration algorithm to induce a pedagogical policy with the goal of enhancing students' learning performance. Empirical evaluation results suggested that the RL policy can improve certain learners' performance as compared to a random policy. Wang et al. [27] applied a variety of Deep RL (DRL) approaches to induce pedagogical policies that aim to improve students' normalized learning gain in an educational game. The simulation evaluation revealed that the DRL policies were more effective than a linear model-based RL policy. Recently, Zhou et al. [29] applied offline Hierarchical Reinforcement Learning (HRL) to induce a pedagogical policy to improve students' normalized learning gain. In a classroom study, the HRL policy was significantly more effective than the other two flat-RL baseline policies. In summary, prior studies suggest that RL-induced pedagogical policies can enhance the effectiveness of tutor-driven ITS where tutors are the ones making pedagogical choices. As far as we know, none of the prior work has attempted to employ RL for an ST-MI-like framework that would allow both students and tutors to make pedagogical decisions, and none of them has examined the effectiveness of the ST-MI framework on student learning.

Identifying Critical Decisions: The advances of computational neuroscience allow researchers to treat the brain as a supercomputing machine to understand the learning and decision-making process in animals and humans [15, 18, 23]. A lot of studies have shown that RL-like signals and decision-making processes exist in humans/animals and we humans use immediate reward and Q-value to make decisions [12]. In RL, the Q-value is defined as the expected cumulative reward for taking an action at a state and following the policy until the end of the episode. Therefore, the difference in Q-values between two actions for a given state reflects the magnitude of the difference in the final outcomes. Motivated by research in human and animal behaviors, lots of RL work has applied Q-value difference as a heuristic measurement for the importance of a state and decide when to give advice in a simulated environment called the "Student-Teacher" framework [25, 32, 5]. Their research question is when to provide an advice and their results showed that the Q-value difference was an effective heuristic function to estimate the importance of a state.

Student Decisions: Much of prior research has shown that while students can benefit from making their own decisions during learning [4, 21], they are not always good at making effective pedagogical decisions. For example, Mitrovic et al. showed that even college students often make poor problem selections [13]. Aleven & Koedinger found that students often do not use hints effectively in that they tended to wait too long before asking for hints [1]. Wood et al. found that students with low prior knowledge exhibit ineffective help-seeking behaviors than those with high prior knowledge [28].

WE, PS, and FWE Many studies have examined the effectiveness of WE, PS, and FWE, as well as their different combinations [26, 17, 16]. Renkl et al. [17] compared WE-FWE-PS with WE-PS pairs and the results showed that WE-FWE-PS condition significantly outperformed WE-PS condition on posttest scores. Similarly, Najar et al. [16] compared adaptive WE/FWE/PS with WE-PS pairs and found that the former is significantly more effective than the latter on improving student learning. Overall, it is demonstrated that adaptively alternating amongst WE, PS, and FWE is more effective than hand-coded expert rules in terms of improving student learning. However, when students making decisions among WE, PS, and FWE, there's no significant difference with tutor making decisions on students' learning performance [30]. As far as we know, no prior research has explored how to combine students' decision-making with RL-induced policy's decision-making to facilitate learning.

3 Method

3.1 Long-Short Term Rewards

To determine whether a state is critical, Critical-DRL considers both short-term reward (ShortTR) and long-term reward (LongTR) [8]. For the ShortTR, it considers the *immediate rewards* over all possible actions to determine the criticality of a state. A primary challenge, however, is that in most ITSs we only have delayed rewards, and immediate rewards are often not available. Specifically, in ITSs, student's learning performance is the most appropriate reward, but it is typically not available until the entire learning trajectory has been completed. Due to the complex nature of learning, it is difficult to assess students' knowledge level moment by moment, and more importantly, many instructional interventions that boost short-term performance may not be effective over the long term. To tackle this issue, we apply a Deep Neural Network-based approach called InferNet, which infers the immediate rewards from delayed rewards. Prior work showed that the InferNet-learned immediate rewards can be as effective as real immediate rewards [2]. Here we employ the InferNet to infer the ShortTR for each state-action pair. Furthermore, to determine whether a state is critical or not, we calculate two thresholds by applying the elbow method to the inferred immediate rewards distribution: one is a *positive reward threshold* above which the agent should pursue, and the other is a *negative reward threshold* below which the agent should avoid. A state is critical if any action on it can lead to an inferred immediate reward either higher than the positive threshold or lower than the negative threshold.

For the LongTR, *Q-value difference* is used to measure the criticality of a state. Q-values are the expected cumulative reward for an agent to take an action a at state s and follow the policy to the end. In theory, if all the actions for a given state have the same Q-value, which one should be taken doesn't matter because they all lead to the same reward. Conversely, if the Q-values of various actions differ widely, taking the wrong action could result in a significant loss of rewards. We define the LongTR of a state s, then, as the difference between its minimum and maximum Q-values: $LongTR(s) = \max_a Q(s,a) - \min_{a'} Q(s,a')$. In general, the higher the LongTR, the more important the state should be.

3.2 Critical Deep Q-Network

In order to determine LongTR, we developed a Critical-DRL approach using Deep Q-Networks (DQN) because of its great success in handling complicated tasks, such as robot control and video game playing [14]. DQN approximates the Q-value function using deep neural networks following the Bellman equation. In the original Bellman equation, the Q-values are calculated assuming that the agent takes the optimal action in every state. In our ST-MI framework, however, optimal actions are taken in critical states, and any action can be taken in non-critical states. Thus, we used the modified Bellman equation as:

$$Q(s,a) = \begin{cases} r + \gamma * max(Q(s',a')) & \text{s' is critical} \\ r + \gamma * average(Q(s',a')) & \text{s' is non-critical.} \end{cases} \tag{1}$$

For a state s and an action a, $Q(s,a)$ follows the original Bellman equation (top) if the next state s' is critical; otherwise we use the average Q-value over all the available actions for s' to update $Q(s,a)$ (bottom). To induce the Critical-DQN policy, we first apply the ShortTR threshold to identify a fixed set of critical states. Then, during each iteration in training, our Critical-DQN algorithm first calculates the Q-value difference $\Delta(Q)$ for all states in the training dataset. Then the median of the Q-value differences is defined as a threshold. If the $\Delta(Q)$ of a state is greater than the threshold, it is critical; otherwise, it is non-critical. The critical states are the union of the two sets identified by the ShortTR and the LongTR, respectively. After the critical states have been determined, the algorithm follows Equation 1 to update the Q-values. Then in the next iteration, the updated Q-values are applied to determine a new median threshold to update the critical states recursively. This process will repeat until convergence. Once the Critical-DQN policy is induced, for any given state, we calculate its Q-value difference and compare it with the corresponding median threshold. If the Q-value difference is larger than the threshold, the state is critical.

3.3 Hierarchical RL Policy Induction

Our ITS first makes the problem-level decisions (WE/PS/FWE) and if a FWE is selected, step-level decisions (elicit/tell) will be made. With the two levels of decisions, we extended the existing flat-RL algorithm to Hierarchical RL (HRL), which aims to induce an optimal policy to make decisions at different levels. Most HRL algorithms are based upon an extension of MDPs called Discrete Semi-Markov Decision Processes (SMDPs). Different from MDPs, SMDPs have an additional set of complex activities or options, each of which can invoke other activities recursively, thus allowing the hierarchical policy to function [3]. The complex activities are distinct from the primitive actions in that a complex activity may contain multiple primitive actions. In our applications, WE, PS, and FWE are complex activities, while elicit and tell are primitive actions. For HRL, learning occurs at multiple levels. A global learning generates a policy for the complex level decisions and local learning generates a policy for the primitive level decisions in each complex activity. More importantly, the goal of local learning is not inducing the optimal policy for the overall task but the optimal policy for the corresponding complex activity. Therefore, our HRL approach learns a global problem-level policy to make decisions on WE/PS/FWE and learns a local step-level policy for each problem to choose between elicit/tell.

4 Policy Induction

Training Corpus: Our training dataset contains a total of 1,307 students' interaction logs collected over seven semesters' classroom studies (2016 Fall to 2020 Spring). During the studies, all students used the same tutor, followed the same general procedure, studied the same training materials, and worked through the same training problems. The training corpus provides us with the state representation, action, and reward information for policy induction. **State:** We extracted 142 features that might impact student learning from the student-system interaction logs. More specifically, these state features can be categorized into the following five groups: *Autonomy:* the amount of work done by the student; *Temporal Situation:* the time-related information about the work process; *Problem-Solving:* information about the current problem-solving context; *Performance:* information about the student's performance during problem-solving; *Student Action:* the statistical measurement of student's behavior. **Action:** Our tutor makes decisions at two levels of granularity: problem and step. In the problem-level, there are three actions WE/PS/FWE. In the step-level, there are two actions elicit/tell. **Reward:** There's no immediate reward during tutoring, and the delayed reward is the students' Normalized Learning Gain (NLG), which measures their learning gain irrespective of their incoming competence. NLG is defined as $\frac{posttest-pretest}{\sqrt{1-pretest}}$, where 1 is maximum score for both pre- and post-test. **Two Policies:** Our **ST-MI** follows the *Critical-DRL model* in that in non-critical states, the student's decision is always carried out, while in critical states, if the student's choice aligns with the ST-MI policy's optimal action, the tutor executes it; otherwise, *the tutor executes the policy's choice and explains it to the student why it is better*. Each type of pedagogical actions has multiple explanation messages, and the tutor will select a message at random to display to students. Because of space constraints, we only include one example message per type of intervention in Table 1. The explanation is intended to smooth the student-system interactions. In our prior work, we found that adding these explanations to RL-induced policies does not improve their effectiveness while adding them to a policy that is not effective does not harm it [31]. The **Expert policy** is designed by an instructor with more than 20 years of experience on the subject. Based on our ITS and prior instructional experience, the Expert policy consists of alternating between elicit and tell at step-level, which was shown to be more effective than other baselines [30].

Table 1. Examples of Explanation Messages in Problem-Level

Student	*ST-MI*	*Explanation Messages*
WE	FWE	*"We are good on time. Let's work together on this problem."*
WE/FWE	PS	*"We are good on time. Try to solve this one yourself."*
PS	FWE	*"To learn more efficiently, let's solve this together."*
PS/FWE	WE	*"You performed pretty well so far. Let me solve this problem."*

5 Experiment Setup

Participants: This study was given to students as a homework assignment in an undergraduate Computer Science class in the Fall of 2020. Students were told to complete the study in one week, and they will be graded based on demonstrated effort rather than learning performance. 153 students were *randomly assigned* into the two conditions: $N = 65$ for Expert and $N = 88$ for ST-MI. *It is important to note that the difference in size between the two conditions is due to the fact that we prioritized having a sufficient number of participants in the ST-MI condition to perform a meaningful analysis of the ATI effect.* Due to preparation for final exams and the length of study, 117 students completed the study. In addition, 12 students were excluded from our subsequent statistical analysis due to the perfect performance in the pre-test. The final group sizes were $N = 47$ for Expert and $N = 58$ for ST-MI. A Chi-square test on the relationship between students' condition and their completion rate found no significant difference between the two conditions: $\chi^2(1) = 2.4335$, $p = 0.12$.

Pyrenees tutor: Our tutor is a web-based ITS to teach students probability and covers 10 major principles, such as the Complement Theorem, Bayes' Rule, etc. It provides step-by-step instruction and immediate feedback. As with other systems, Pyrenees provides students with help via a sequence of increasingly specific hints, which prompts them with what they should do next. The last hint in the sequence, i.e., the bottom-out hint, tells the student exactly what to do.

Experiment Procedure & Grading: Both conditions went through the same four phases: 1) textbook, 2) pre-test, 3) training on the ITS, and 4) post-test. The only difference among them was how the pedagogical decisions were made. During **textbook**, all students read a general description of each principle, reviewed some examples, and solved some training problems. The students then took a **pre-test** which contained a total of 14 single- and multiple-principle problems. Students were not given feedback on their answers, nor were they allowed to go back to earlier questions (this was also true for the post-test). During **training**, both conditions received the same 12 problems in the same order. Each domain principle was applied at least twice. Finally, all students took the 20-problem **post-test**: 14 of the problems were isomorphic to the pre-test, and the remainders were non-isomorphic multiple-principle problems. All of the tests were graded in a double-blind manner by a single experienced grader. For comparison purposes, all test scores were normalized to the range of $[0, 1]$.

6 Results

6.1 ST-MI vs. Expert

Pre-test Score: No significant difference was found between the Expert condition ($M = 0.77, SD = 0.13$) and the ST-MI condition ($M = 0.73, SD = 0.22$) on the pre-test scores: $t(103) = 1.18$, $p = 0.23$, $d = 0.23$. It suggests that the two conditions are balanced in terms of incoming competence.

Improvement through Training: A repeated measures analysis using test type (pre-test vs. isomorphic post-test) as a factor and test score as the dependent measure showed a main effect for test type for both conditions in that students scored significantly higher in the isomorphic post-test than in the pre-test: $F(1, 46) = 10.6$, $p = .0016$, $\eta = 0.319$ for Expert and $F(1, 57) = 13.64$, $p = .0003$, $\eta = 0.315$ for ST-MI respectively. In details, the isomorphic post-test scores in the ST-MI condition is ($M = 0.86, SD = 0.19$) while the Expert condition is ($M = 0.85, SD = 0.11$). It shows that both conditions learned significantly from training on our tutor.

Learning Performance & Training Time: In comparing students' learning performance between the two conditions, we compared their isomorphic posttest and full posttest scores, as well as their isomorphic and full NLGs. The goal of the isomorphic posttest is to assess the learning gain and whether or not the tutor is helpful, while the purpose of the full posttest is to determine whether the intervention makes a difference in student learning. There was no significant difference between the two conditions on either isomorphic posttest or posttest. For

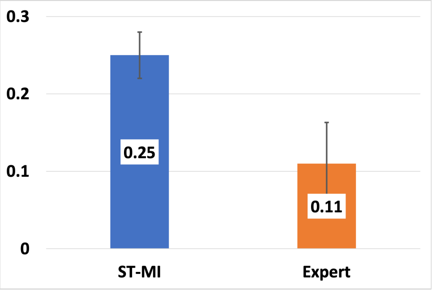

Fig. 2. Isomorphic NLG

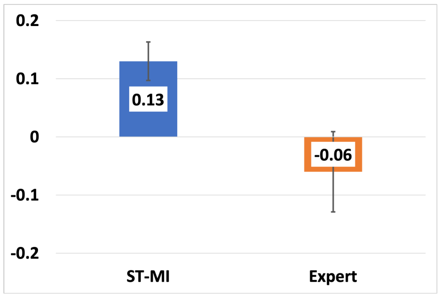

Fig. 3. Full NLG

example, the ST-MI students had higher post-test scores ($M = 0.81, SD = 0.21$) than the Expert students ($M = 0.78, SD = 0.15$) but such difference is not significant: $t(103) = 0.83, p = 0.411, d = 0.16$. Our most important interest, however, is in student performance improvement from pre- to posttest, so we focus on isomorphic NLG and NLG. ***NLGs of both types demonstrate how beneficial our ITS actually are, as well as their role as reward functions in our Critical DRL framework***. The ST-MI condition scored significantly higher than the Expert condition on both the isomorphic NLG: $t(103) = 2.35$, $p = 0.021$, $d = 0.46$ and the full NLG: $t(103) = 2.72$, $p = .008$, $d = 0.53$. In Fig 2, the isomorphic NLG for the ST-MI condition is ($M = 0.25, SD = 0.23$) and the Expert condition is ($M = 0.11, SD = 0.36$). Similarly, in Fig 3, the full NLG for the ST-MI condition is ($M = 0.13, SD = 0.25$) while the Expert condition is ($M = -0.06, SD = 0.47$). Finally, on training time the ST-MI condition spend less time (measured in minutes, $M = 109.6, SD = 38.1$) than the Expert condition ($M = 123.2, SD = 47.1$) during the training on the tutor but the difference is not significantly: $t(103) = -1.63$, $p = 0.106$, $d = 0.32$. In short, our results indeed show that the ST-MI policy significantly improves students' learning gains with less time cost than the Expert policy.

6.2 The Impact of ST-MI on ATI Effect

In order to measure ATI, we further divided students into High vs. Low groups by a median split on their pretest scores, also known as incoming competence. Thus, we had four groups based upon their pretest scores and policies: High-ST-MI (n=28), Low-ST-MI (n=30), High-Expert (n=21), Low-Expert (n=26). No significant difference was found among the two conditions on the distribution of High vs. Low students: $\chi^2(1) = 0.0291$, $p = 0.86$. Table 2 presents the comparison between the policies {ST-MI, Expert} and incoming competence {High, Low} in terms of learning performance. As expected, in both conditions the high group significantly outperformed their low peers in the pretest: $t(45) = 6.07$, $p < 0.001$, $d = 1.10$ for Expert and $t(56) = 9.03$, $p < 0.001$, $d = 1.10$ for ST-MI. Moreover, while no significant difference was found between the High-Expert and High-ST-MI ones: $t(47) = 0.33$, $p = 0.74$, $d = 1.10$, the Low-Expert significantly out-performed the Low-ST-MI ones: $t(54) = 2.57$, $p = 0.012$, $d = 1.10$.

Table 2. Learning Performance for Four Groups

Group	*Pre*	*Iso Post*	*Post*	*Iso NLG*	*NLG*	*Time*
Low-Expert	0.67 (0.08)	0.80 (0.11)	0.71 (0.15)	0.23 (0.20)	0.06 (0.27)	129.9 (52)
Low-ST-MI	0.58 (0.21)	0.78 (0.23)	0.70 (0.24)	0.31 (0.22)	0.18 (0.24)	108.6 (44)
High-Expert	0.90 (0.05)	0.91 (0.08)	0.86 (0.11)	-0.02 (0.46)	-0.21 (0.61)	114.8 (40)
High-ST-MI	0.88 (0.06)	0.96 (0.06)	0.92 (0.07)	0.19 (0.23)	0.07 (0.26)	111.8 (31)

Table 2 shows that the test score results are consistent with our hypothesis. Despite their significantly lower pre-test scores, the Low-ST-MI students catch up with their Low-Expert peers on the following four performance measures in that no significant difference was found between them on Iso-Post, Post, Iso

NLG, and full NLG. According to [30], the low incoming competence students are less likely to benefit from making pedagogical decisions on their own, but our results showed that our ST-MI framework with ST-MI policy could make them catch up to their peers in the Expert condition. As for the two High groups, both scored high for the isomorphic and full posttests, and the High-ST-MI group outperformed the High-Expert group on both the Iso NLG: $t(47) = 2.59, p = 0.011, d = 0.41$, and NLG: $t(47) = 2.78, p = 0.007, d = 0.40$. While previous research has shown that high incoming students are just as effective at learning as those who make their own decisions or follow the Expert policy [30], our findings showed that despite having a high score in the Iso-post and Posttest scores, the High-Expert group does not seem to benefit from the tutor, as their average NLG is negative. In contrast, ST-MI policy can significantly enhance the High performers' learning gains when compared with Expert policy.

In summary, our findings confirm that ST-MI can benefit both High and Low performers. More specifically, low performers who are more sensitive to learning environments can parallel their Expert peers with our framework, while high performers, who are less sensitive to learning environments and always perform well, can further boost their learning gains with our ST-MI framework.

6.3 Log Analysis

Table 3. Problem-Level Critical Decisions in ST-MI

Decisions	*High*	*Low*	*T-test Result*
Critical Decision	8.2 (1.8)	6.1 (2.9)	**$t(56)$=3.36,p=0.001*,d=0.88**
Correct Critical Choice	3.4 (2.0)	2.5 (2.3)	$t(56)$=1.46,p=0.150,d=0.38
Intervention	4.9 (2.6)	3.5 (1.9)	**$t(56)$=2.21,p=0.031*,d=0.58**

Next, we analyze the pedagogical decision behaviors between the High and Low groups in the ST-MI condition. Table 3 shows the average number of different types of critical decisions students received in the problem-level. In Table 3, there are three types of critical decisions: 'Critical Decision' means the decision state is identified as critical by our ST-MI policy; 'Correct Critical Choice' means the students select the optimal actions (same as our policy's choice) in the critical decision; 'Intervention' means the students select the sub-optimal actions (different from our policy's choice) in the critical decision. By definition, correct critical choices and intervention are exclusive and they are subsets of critical decisions. First, the High students experienced significantly more critical decisions than the Low students. Then, by facing more critical moments, not only were the High students able to make more correct critical choices (not significant), but also they received more interventions (significant) to achieve their goals. Additionally, there's no significant difference between High vs. Low on all three types of critical decisions in the step-level. In summary, the results showed that the High students experienced more interventions than the Low group students, and as a result, the intervention could help the High students experience more critical optimal actions, which can lead to better learning performance.

7 Conclusion

In the classroom study, we evaluated the effectiveness of the ST-MI framework by comparing the ST-MI policy with a baseline Expert policy. In the ST-MI condition, students could control their own learning process by making decisions on what type of questions they want, and in the meantime, the RL-induced policy would intervene when they make sub-optimal choices in critical decisions and give dedicated explanations. The results show that the students in the ST-MI condition significantly outperform the students in the Expert condition in terms of learning performance. Additionally, a log analysis suggests that the students with high incoming competence received more interventions than the students with low incoming competence. The reason is that the RL-induced policy aims to maximize NLG, and the high students usually have lower NLG due to little room to improve. As a result, the RL-induced policy would intervene more on the high students to improve their NLG. Finally, we observe a trend that giving students control over their learning could make the learning more efficient. Overall, the empirical study demonstrates that our proposed ST-MI framework could improve students' learning without the trivial tutor-driven step decisions.

Acknowledgements: This research was supported by the NSF Grants: 1660878, 1651909, 1726550 and 2013502.

References

1. Aleven, V., Koedinger, K.R.: Limitations of student control: Do students know when they need help? In: Intelligent Tutoring Systems. pp. 292–303 (2000)
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: AIED (2021)
3. Barto, A.G., Mahadevan, S.: Recent advances in hierarchical reinforcement learning. Discrete event dynamic systems **13**(1-2), 41–77 (2003)
4. Cordova, D.I., Lepper, M.R.: Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. Journal of educational psychology **88**(4), 715 (1996)
5. Fachantidis, A., Taylor, M.E., Vlahavas, I.P.: Learning to teach reinforcement learning agents. Machine Learning and Knowledge Extraction (2017)
6. Flowerday, T., Schraw, G., Stevens, J.: The role of choice and interest in reader engagement. The Journal of Experimental Education **72**(2), 93–114 (2004)
7. Ju, S., Zhou, G., Abdelshiheed, M., Barnes, T., Chi:, M.: Evaluating critical reinforcement learning framework in the field. AIED pp. 215–227 (2021)
8. Ju, S., Zhou, G., Barnes, T., Chi, M.: Pick the moment: Identifying critical pedagogical decisions using long-short term rewards. In: EDM (2020)
9. Kinzie, M.B., Sullivan, H.J.: Continuing motivation, learner control, and cai. Educational Technology Research and Development **37**(2), 5–14 (1989)
10. Mandel, T., Liu, Y.E., Levine, S., Brunskill, E., Popovic, Z.: Offline policy evaluation across representations with applications to educational games. In: AAMAS. pp. 1077–1084 (2014)
11. Maniktala, M., Cody, C., Barnes, T., Chi, M.: Avoiding help avoidance: Using interface design changes to promote unsolicited hint usage in an intelligent tutor. International Journal of Artificial Intelligence in Education **30**(4), 637–667 (2020)

12. McClure, S.M., Laibson, D.I., Loewenstein, G., Cohen, J.D.: Separate neural systems value immediate and delayed monetary rewards. Science pp. 503–507 (2004)
13. Mitrovic, A., Martin, B.: Scaffolding and fading problem selection in sql-tutor. In: AIED. pp. 479–481 (2003)
14. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature (518), 529–533 (2015)
15. Morris, G., Nevet, A., Arkadir, D., Vaadia, E., Bergman, H.: Midbrain dopamine neurons encode decisions for future action. NatureNeuro **9**(8), 1057–1063 (2006)
16. Najar, A.S., Mitrovic, A., McLaren, B.M.: Adaptive support versus alternating worked examples and tutored problems: Which leads to better learning? In: UMAP, pp. 171–182. Springer (2014)
17. Renkl, A., Atkinson, R.K., Maier, U.H., Staley, R.: From example study to problem solving: Smooth transitions help learning. J Exp Educ **70**(4), 293–315 (2002)
18. Roesch, M.R., Calu, D.J., Schoenbaum, G.: Dopamine neurons encode the better option in rats deciding between different delayed or sized rewards. Nature Neuroscience **10**(12), 1615–1624 (2007)
19. Roll, I., Wiese, E.S., Long, Y., Aleven, V., Koedinger, K.R.: Tutoring self-and coregulation with intelligent tutoring systems to help students acquire better learning skills. Design recommendations for intelligent tutoring systems **2**, 169–182 (2014)
20. Rowe, J.P., Lester, J.C.: Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework. In: AIED. pp. 419–428. Springer (2015)
21. Schneider, S., Nebel, S., Beege, M., Rey, G.D.: The autonomy-enhancing effects of choice on cognitive load, motivation and learning with digital media. Learning and Instruction **58**, 161–172 (2018)
22. Shen, S., Chi, M.: Reinforcement learning: the sooner the better, or the later the better? In: UMAP. pp. 37–44. ACM (2016)
23. Sul, J.H., Jo, S., Lee, D., Jung, M.W.: Role of rodent secondary motor cortex in value-based action selection. Nature Neuroscience **14**(9), 1202–1208 (2011)
24. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (2018)
25. Torrey, L., Taylor, M.E.: Teaching on a budget: Agents advising agents in reinforcement learning. AAMAS pp. 1053–1060 (2013)
26. Van Gog, T., Kester, L., Paas, F.: Effects of worked examples, example-problem, and problem-example pairs on novices' learning. Contemporary Educational Psychology **36**(3), 212–218 (2011)
27. Wang, P., Rowe, J., Min, W., Mott, B., Lester, J.: Interactive narrative personalization with deep reinforcement learning. In: IJCAI (2017)
28. Wood, H., Wood, D.: Help seeking, learning and contingent tutoring. Computers & Education **33**(2), 153–169 (1999)
29. Zhou, G., Azizsoltani, H., Ausin, M.S., Barnes, T., Chi, M.: Hierarchical reinforcement learning for pedagogical policy induction. In: AIED. pp. 544–556 (2019)
30. Zhou, G., Chi, M.: The impact of decision agency & granularity on aptitude treatment interaction in tutoring. CogSci pp. 3652–3657 (2017)
31. Zhou, G., Yang, X., Azizsoltani, H., Barnes, T., Chi, M.: Improving student-tutor interaction through data-driven explanation of hierarchical reinforcement induced pedagogical policies. In: UMAP. ACM (2020)
32. Zimmer, M., Viappiani, P., Weng, P.: Teacher-student framework: A reinforcement learning approach. AAMAS Workshop (2013)