Evaluating Critical Reinforcement Learning Framework in the Field

Song Ju, Guojing Zhou, Mark Abdelshiheed, Tiffany Barnes, and Min Chi(✉)

Department of Computer Science, North Carolina State University,
Raleigh, NC 27695, USA
{sju2,gzhou3,mnabdels,tmbarnes,mchi}@ncsu.edu

Abstract. Reinforcement Learning (RL) is learning what action to take next by mapping *situations* to *actions* so as to maximize cumulative rewards. In recent years RL has achieved great success in inducing effective pedagogical policies for various interactive e-learning environments. However, it is often prohibitive to identify the *critical* pedagogical decisions that actually contribute to desirable learning outcomes. In this work, by utilizing the RL framework we defined *critical decisions* to be those states in which the agent has to take the optimal actions, and subsequently, the *Critical policy* as carrying out optimal actions in the critical states while acting randomly in others. We proposed a general *Critical-RL framework* for identifying critical decisions and inducing a Critical policy. The effectiveness of our Critical-RL framework is empirically evaluated from two perspectives: whether optimal actions *must* be carried out in critical states (*the necessary hypothesis*) and whether only carrying out optimal actions in critical states is as effective as a fully-executed RL policy (*the sufficient hypothesis*). Our results confirmed both hypotheses.

Keywords: Critical decisions · Reinforcement learning · ITS

1 Introduction

Intelligent Tutoring Systems (ITSs) have been shown to be effective for improving student learning. Most ITSs are adaptive instructional systems in that tutor decides what to do next. For example, the tutor can elicit the solution to the next step from the students with prompting and support or without. At each step, the ITS records its success or failure and may give feedback (e.g. correct/incorrect signals) and hints (suggestions for what to do next) automatically or on-demand. Alternatively, the tutor can choose to tell students the solution to the next step directly. Each of these tutor decisions will affect the students' subsequent actions and performance, and some may be more *impactful* than others. *Pedagogical policies* are used for the agent (tutor) to decide what action to take next in the face of alternatives.

Reinforcement Learning (RL) offers one of the most promising approaches to data-driven decision-making. RL algorithms are designed to induce effective

I. Roll et al. (Eds.): AIED 2021, LNAI 12748, pp. 215–227, 2021.
https://doi.org/10.1007/978-3-030-78292-4_18

policies that determine the best action for an agent to take in any given situation to maximize a cumulative reward. In recent years, RL, especially Deep RL, has achieved superhuman performance in several complex games [1,31,32]. However, different from the classic game-play situations where the ultimate goal is to make the agent effective, in human-centric tasks such as ITSs, the ultimate goal is for the agent to make the *student-system interactions* more productive and fruitful. Several researchers have studied the application of existing RL algorithms to improve the effectiveness of interactive e-learning environments such as ITSs [7,10,22,25–28,30,33,40,43]. While promising, relatively little work has been done to analyze, interpret, explain, or generalize RL-induced policies. While traditional hypothesis-driven, cause-and-effect approaches offer clear conceptual and causal insights that can be evaluated and interpreted, RL-induced policies especially Deep RL-induced ones, are often referred to as black-box models. This raises a major open question: *How can we identify the critical system pedagogical decisions that are linked to student learning outcomes?*

In this work, by utilizing the RL framework, we defined *critical decisions* to be those states in which the agent has to take the optimal actions and subsequently defined *Critical policy* as carrying out optimal actions in the critical states while acting randomly in others. We proposed a general *Critical-RL framework* for identifying critical decisions and inducing a Critical policy. In our prior work, we evaluated the effectiveness of our Critical-RL framework using simulations and our results showed that by carrying out critical decisions only, our Critical policy can be as effective as a fully executed RL policy. In this work, we *empirically* evaluate the Critical-RL framework in a classroom setting. To confirm whether the identified critical decisions are indeed critical, we argue that our identified critical decisions and induced Critical policy should satisfy two conditions.

First, they should satisfy the *Necessary Hypothesis* stating that it is *necessary* to carry out optimal actions in critical states otherwise the performance would suffer. To validate it, we compared two policies: Critical-optimal ($\text{Critical}_{\text{opt}}$) vs. Critical-suboptimal ($\text{Critical}_{\text{sub}}$). Both policies would carry out random actions in non-critical states and the only difference is that in critical states, $\text{Critical}_{\text{opt}}$ takes optimal actions while $\text{Critical}_{\text{sub}}$ takes suboptimal actions. As expected, our results showed that the former was indeed significantly more effective than the latter. Second, our induced Critical policy should satisfy the *Sufficient Hypothesis* stating that carrying out optimal actions in the critical states is *sufficient*. In other words, only carrying out optimal actions in critical states is as effective as a fully-executed RL policy. To validate it, we compared the $\text{Critical}_{\text{opt}}$ policy with a Full RL policy which takes optimal actions in every state. Our results showed that no significant difference was found between them.

In this work, we focus on pedagogical decisions at two levels of granularity: *problem* and *step*. More specifically, our tutor will first make a problem-level decision and then make step-level decisions based on the problem-level decision. For the former, our tutor first decides whether the next *problem* should be a worked example (WE), problem solving (PS), or a faded worked example (FWE). In WEs, students observe how the tutor solves a problem; in PSs students solve

the problem themselves; in FWEs, the students and the tutor *co-construct* the solution. 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 show it to the student directly. We refer to such decisions as *elicit/tell* decisions. If WE is selected, an all-tell step policy will be carried out; if PS is selected, an all-elicit policy will be executed; finally, if FWE is selected, the tutor will decide whether to elicit or tell a step based on the corresponding induced step-level policy. While much of the prior work has relied on hand-coded or RL-induced pedagogical policies on these decisions, there is no well-established theory or widely accepted consensus on how WE vs. PS. vs. FWE can be best used and how they may impact students' learning. As far as we know, no prior research has investigated *when it is critical to give WE vs. PS vs. FWE*. In this work, by empirically confirming that our identified critical decisions and Critical policy satisfy the two hypotheses, we argue that the proposed Critical-RL framework sheds some light on identifying the moments that offering WE, PS, or FWE can make a difference.

2 Related Work

2.1 Applying RL to ITSs

Prior work has shown that RL can induce effective pedagogical policies for Intelligent Tutoring Systems [2,3,6,11,14,21,38]. For example, Shen et al. [29] applied an offline RL approach, value iteration, to induce a pedagogical policy with the goal of improving students' learning performance. Empirical evaluation results suggested that the RL policy can improve certain learners' performance as compared to a random policy. Mandel et al. [14] applied a partially observable Markov decision process (POMDP) to induce a pedagogical policy that aims to maximize students' learning gain. The effectiveness of the POMDP policy was evaluated by comparing it with an expert policy, and a random policy, on both simulated students and real students. Results showed that the POMDP policy significantly outperformed the other two. Wang et al. [38] applied a variety of Deep RL (DRL) approach to induce pedagogical policies aims at improving students' normalized learning gain in an educational game. Simulation evaluation results suggested that the DRL policies were more effective than a linear model-based RL policy. Finally, Zhou et al. [41] applied Hierarchical Reinforcement Learning (HRL) to induce a pedagogical policy to improve students' normalized learning gain. The HRL policy makes decisions first at the problem level and then at the step level. In a classroom study, the HRL policy was compared with two step level policies: DQN and random. Results showed that the HRL policy was significantly more effective than the other two.

In sum, prior work suggests that employing RL-induced pedagogical policies can improve the effectiveness of ITSs. However, despite this effectiveness, RL policies often make a lot of fine-grained decisions in training. For example, the HRL policy induced by Zhou et al. [41] can make over 400 decisions in 12 training

problems. Therefore, it can be difficult to identify and study the origin of this fine-grained decision-making style of RL policies.

2.2 Identifying Critical Decisions

Recent advances in computational neuroscience have enabled researchers to simulate and study the decision-making mechanisms of humans and animals through computational approaches [13,15,19,24,34]. A number of works showed that RL-like learning and decision-making processes exist in humans/animals and we humans use immediate reward and Q-value to make decisions [13,15]. In RL, the Q-value is defined as the expected cumulative reward for taking an action a at state s and following the policy until the end of the episode. Therefore, the difference of Q-values between two actions reflects the magnitude of difference in the final outcomes. Motivated by research in human and animal behaviors, a lot of RL work has applied Q-value difference to measure the importance of a state and decide when to give advice in a simulated environment called the "Student-Teacher" framework [8,9,36,44]. In this framework, a "student" agent learns from the interaction with the environment, while a "teacher" agent provides action suggestions to accelerate the learning process. Their research question is not what to advise but when to advise, especially with a limited budget of advice. Results showed that the Q-value difference approach is significantly better than baseline strategies such as random advising and early advising. Overall, prior studies explored the problem of when to give advice in simulated environments. They showed that Q-value difference is an accurate heuristic function to estimate the importance of a state. However, they have not considered the immediate rewards and have not validated their findings on human students.

2.3 WE, PS and FWE

A variety of studies have explored the effectiveness of WE, PS, FWE, and their various combinations [16,17,20,23,37,39,42]. For example, Mclaren et al. compared WE-PS pairs with PS-only in a study [17] and WE-only, PS-only and WE-PS pairs in another study [16]. Overall, results suggested that studying WE can be as effective as doing PS, but students spend less time on WE. For FWE-involved studies, Renkl et al. [23] compared WE-FWE-PS with WE-PS pairs. Results showed that the WE-FWE-PS condition significantly outperformed the WE-PS condition, and there is no significant time-on-task difference between them. Similarly, Najar et al. [20] compared adaptive WE/FWE/PS with WE-PS pairs and found the former is significantly more effective than the latter. In summary, prior studies have demonstrated that adaptively alternating amongst WE, PS, and FWE is more effective than hand-coded expert rules in terms of improving student learning. However, it is still not clear which alternating is critical to the student learning outcome.

3 Method

3.1 Critical Deep Q-Network

To determine whether a state is critical, our Critical-RL framework considers both short-term reward (immediate reward) and long-term reward (Q-value difference). For the former, we consider the amount of the immediate rewards over all possible actions to determine the criticalness of a state. One of the primary challenges is that *on most ITSs we only have delayed rewards, and immediate rewards are often not available*. The most appropriate rewards to use in ITSs are student learning performance, which is typically delayed until the entire trajectory is complete. This is due to the complex nature of learning, which makes it 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 [4] to infer the immediate rewards from delayed rewards. Prior work has evaluated the effectiveness of inferred rewards, and results showed that inferred immediate rewards can be as effective as real immediate rewards in our application. Therefore, we think the *inferred* immediate rewards from InferNet are reliable to be considered as short-term rewards in our Critical-RL framework. More specifically, we apply the elbow method on the distribution of the inferred immediate rewards to determine two thresholds: 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. If any action on a state can lead to an inferred immediate reward either higher than the positive threshold or lower than the negative one, it should be critical.

To get the long-term rewards, our Critical-RL framework used Deep Q-Network (DQN). In recent years, DQN has shown a strong ability to handle complicated tasks, such as robot control and video game playing [18]. DQN approximates the Q-value function using deep neural networks following the Bellman equation. In the original DQN, the Q-values are calculated based on the assumption that the agent takes the optimal action in every state. However, in our Critical-RL framework, the **Critical policy** takes optimal actions only in the critical states, and takes random action in the non-critical states. To accommodate this difference, we modify the original Bellman equation:

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

In Eq. 1, when the state s' is critical, its value function is the max Q-value of the optimal action while when it is non-critical, its value function is the mean Q-value over all the available actions. To induce the Critical-DQN policy, during each iteration in training, our algorithm first calculates the Q-value difference $\Delta(Q)$ for all states in the training dataset, where $\Delta(Q) = \max_a Q(s,a) - \min_a Q(s,a)$. Then the median of the 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. After

the critical states have been determined, the algorithm follows Eq. 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-RL 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.

3.2 Hierarchical RL Policy Induction

Our tutor can make both problem-level decisions (WE/PS/FWE) and step-level decisions (elicit/tell). With the two levels of granularity, 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 Markov Decision Processes (MDPs) called Discrete Semi-Markov Decision Processes (SMDPs). Different from MDPs, SMDPs have an additional set of complex activities [5] or options [35], each of which 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. 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. More specifically, both problem and step level policies were learned by recursively using DQN or Critical-DQN to update the Q-value function until convergence.

4 Policy Induction

Training Corpus: Our training dataset contains a total of 1,148 students' interaction logs collected over six semesters' classroom studies (16 Fall to 19 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 components for RL induction are defined as follows:

State: From the student-system interaction logs, 142 features were extracted to represent the student learning state, which can be categorized into five groups: *Autonomy*(10) features describe the amount of work done by the student; *Temporal*(29) features are the time-related information during tutoring; *Problem Solving*(35) features indicate the context of the problem itself; *Performance*(57) features denote student's performance, and *Student Action*(11) features record the student behavior information. **Action:** Our tutor can make both problem

and step-level decisions. There are two actions (elicit/tell) at the step level and three actions (WE/PS/FWE) at the problem level. **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.

Three Policies: We induced a standard DQN policy as the Full policy to carry out optimal actions in all states. Note that our prior work showed that the Full policy significantly outperformed the expert-designed policy on improving students' learning performance [12]. In this work, we induced a Critical-DQN policy to identify critical states. The $\text{Critical}_{\text{opt}}$ policy would carry out optimal actions in critical states but the $\text{Critical}_{\text{sub}}$ policy would take sub-optimal actions with minimum Q-value. In non-critical states, both of them acted randomly.

5 Empirical Experiment

Participants: This study was given to students as a homework assignment in an undergraduate Computer Science class in the Spring 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. 164 students were randomly assigned into three conditions: $N = 58$ for $\text{Critical}_{\text{opt}}$, $N = 55$ for $\text{Critical}_{\text{sub}}$ and $N = 51$ for Full. Due to preparation for final exams and the length of study, 129 students completed the study. In addition, 14 students were excluded from our subsequent statistical analysis in which 8 students performed perfectly in the pre-test and 6 students worked in groups. The final group sizes were $N = 37$ for $\text{Critical}_{\text{opt}}$, $N = 39$ for $\text{Critical}_{\text{sub}}$ and $N = 39$ for Full. A Chi-square test on the relationship between students' condition and their completion rate found no significant difference among the conditions: $\chi^2(2) = 0.167$, $p = 0.92$.

Pyrenees Tutor: Our tutor is a web-based ITS teaching probability. It covers ten major principles of probability, such as the Additional Theorem, De Morgan's Theorem, and Bayes Rule. The Pyrenees tutor provides step-by-step adaptive instructions, immediate feedback, and on-demand hints to prompt students' learning. More specifically, help in Pyrenees tutor is provided via a sequence of increasingly specific hints, in which the last hint tells the student exactly what to do next.

Procedure and Grading: In the classroom study, students were required to complete 4 phases: 1) pre-training, 2) pre-test, 3) training on Pyrenees tutor, and 4) post-test. During the **pre-training** phase, all students studied the domain principles through a probability textbook, reviewed some examples, and solved certain training problems. Students then took a **pre-test** which contained 14 probability problems. The textbook was not available at this phase and 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**, students

in all three conditions received the same 12 problems in the same order on Pyrenees tutor. The minimal number of steps needed to solve each problem ranged from 20 to 50, which included defining variables, applying principles, and solving equations. Each domain principle was applied at least twice in the 12 problems, and all of the students could access the textbook during this phase. Finally, all of the students completed a **post-test** with 20 problems: 14 of the problems were isomorphic to the pre-test, and the remaining six were non-isomorphic complicated problems. The pre- and post-test were graded in a double-blind manner by experienced graders. All scores are normalized in the range of 0 to 1.

6 Results

We will report our results based on the two hypotheses. For the Necessary Hypothesis, we compare $\text{Critical}_{\text{opt}}$ vs. $\text{Critical}_{\text{sub}}$ conditions and for the Sufficient Hypothesis, we compare $\text{Critical}_{\text{opt}}$ vs. Full conditions.

6.1 Necessary Hypothesis ($\text{Critical}_{\text{opt}}$ vs. $\text{Critical}_{\text{sub}}$)

Table 1 shows the comparisons between $\text{Critical}_{\text{opt}}$ (in gray) vs. $\text{Critical}_{\text{sub}}$. The left four columns show the mean and standard deviation (SD) of their learning performance, percentage of critical states and tutor decisions with the corresponding pairwise t-test results. No significant difference was found between the two conditions on pre-test: $t(112) = 0.56$, $p = .57$, $d = 0.13$. The result suggests that the two conditions are balanced in terms of incoming competence.

Table 1. Results of necessary hypothesis: $\text{Critical}_{\text{opt}}$ vs. $\text{Critical}_{\text{sub}}$

Learning performance				
	Critical_{opt}	Critical_{sub}	Pairwise T-test result	Full
Pre	0.75 (0.18)	0.72 (0.20)	$t(112) = 0.56$, $p = .570$, $d = 0.13$	0.70(0.19)
Iso Post	0.89 (0.16)	0.86 (0.16)	$t(112) = 0.81$, $p = .420$, $d = 0.18$	0.84(0.20)
Full Post	0.82 (0.19)	0.78 (0.19)	$t(112) = 0.99$, $p = .320$, $d = 0.23$	0.75(0.20)
Iso NLG	0.70 (0.36)	0.40 (0.85)	**$t(112) = 2.27$, $p = .025$*, $d = 0.52$**	0.56(0.40)
Full NLG	0.41 (0.39)	0.01 (1.25)	**$t(112) = 2.18$, $p = .031$*, $d = 0.49$**	0.18(0.55)
Time	94.5 (35.1)	78.1 (26.7)	**$t(112) = 2.30$, $p = .023$*, $d = 0.52$**	91.5(31.7)
Percentage of critical states				
Prob-Level	46.9 (23.4)	31.5 (17.6)	**$t(112) = 3.69$, $p < .001$*, $d = 0.84$**	38.4(12.4)
Step-Level	60.2 (20.1)	45.3 (26.0)	**$t(112) = 2.42$, $p = .017$*, $d = 0.55$**	62.1(34.0)
Tutor decisions				
PS	3.56 (1.85)	2.38 (1.41)	**$t(112) = 3.60$, $p < .001$*, $d = 0.81$**	3.32(0.81)
WE	2.54 (1.87)	5.13 (1.51)	**$t(112) = -7.27$, $p < .001$*, $d = 1.65$**	5.24(1.19)
FWE	3.90 (2.00)	2.49 (1.34)	**$t(112) = 4.01$, $p < .001$*, $d = 0.91$**	1.43(1.10)
Elicit	83.3 (49.2)	44.0 (30.6)	**$t(112) = 4.37$, $p < .001$*, $d = 0.99$**	33.2(35.1)
Tell	82.9 (50.3)	55.4 (35.2)	**$t(112) = 3.06$, $p = .003$*, $d = 0.69$**	29.8(28.3)

Improvement Through Training: To measure the improvement students gained through the ITS training, we compared their pre-test and isomorphic post-test scores. A repeated measures analysis showed that both conditions scored significantly higher in the post-test than in the pre-test: $F(1, 38) = 13.68$, $p = .0004$, $\eta = 0.392$ for $\text{Critical}_{\text{opt}}$ and $F(1, 38) = 11.5$, $p = .0011$, $\eta = 0.362$ for $\text{Critical}_{\text{sub}}$. It suggests that our ITS indeed helps students learning regardless of the pedagogical policies deployed.

Learning Performance: To investigate students' learning performance between the two conditions, we compared their isomorphic NLG (calculated based on Pre- and Iso Post-test) and full NLG (based on Pre- and Full Post-test). The full post-test contains six additional multiple-principle problems. Pairwise t-tests showed that $\text{Critical}_{\text{opt}}$ scored significantly higher than $\text{Critical}_{\text{sub}}$ on both the isomorphic NLG: $t(112) = 2.27$, $p = .025$, $d = 0.52$ and the full NLG: $t(112) = 2.18$, $p = .031$, $d = 0.49$. The results showed that the $\text{Critical}_{\text{opt}}$ policy is more effective than the $\text{Critical}_{\text{sub}}$ policy. It supports our hypothesis that different actions in the critical states can make a significant difference, so optimal actions *must* be made in critical states.

Time on Task and Percentage of Critical States: A pairwise t-test analysis revealed that $\text{Critical}_{\text{opt}}$ spend significantly more time (measured in minutes) than $\text{Critical}_{\text{sub}}$ in the training phase: $t(112) = 2.30$, $p = .023$, $d = 0.52$. The middle section in Table 1 presents the percentage of critical states (both problem and step level) each condition experienced. Pairwise t-test showed that $\text{Critical}_{\text{opt}}$ experienced significantly more critical states than $\text{Critical}_{\text{sub}}$ on both problem level: $t(112) = 3.69$, $p < .001$, $d = 0.84$ and step level: $t(112) = 2.42$, $p = .017$, $d = 0.55$. This suggests that the $\text{Critical}_{\text{opt}}$ policy is more likely to lead students to the critical intersections that make a difference.

Tutor Decisions: We investigated the number of different types of actions students received during training, as shown in the lower section of the Table 1. *Note that for step level decisions, we only considered the elicits and tells in the FWEs.* For the problem level, $\text{Critical}_{\text{opt}}$ received significantly more PS: $t(112) = 3.60$, $p < .001$, $d = 0.81$, more FWE: $t(112) = 4.01$, $p < .001$, $d = 0.91$ and fewer WE: $t(112) = -7.27$, $p < .001$, $d = 1.65$ than $\text{Critical}_{\text{sub}}$. For the step level, the former also received significantly more elicit: $t(112) = 4.37$, $p < .001$, $d = 0.99$ and more tell: $t(112) = 3.06$, $p = .003$, $d = 0.69$ than $\text{Critical}_{\text{sub}}$. The results indicate that the $\text{Critical}_{\text{sub}}$ policy prefers WEs while the $\text{Critical}_{\text{opt}}$ policy prefers PSs and FWEs.

6.2 Sufficient Hypothesis ($\text{Critical}_{\text{opt}}$ vs. Full)

In the Sufficient Hypothesis, we expect no significant difference in learning performance between the $\text{Critical}_{\text{opt}}$ and Full conditions. To align the analysis, we still focus on the three aspects as above (learning performance, critical states, tutor decisions). To save space, the statistics of the Full condition were shown in the rightmost column in Table 1. A pairwise t-test showed that there is no significant difference between $\text{Critical}_{\text{opt}}$ (2nd column in gray) vs. Full (last column)

on the pre-test score: $t(112) = 1.18$, $p = .24$, $d = 0.27$. This suggests again that our random assignment indeed balanced students' incoming competence.

Improvement Through Training: A repeated measures analysis using test-type (pre-test and isomorphic post-test) as factors and test score as dependent measure showed that similar to $\text{Critical}_{\text{opt}}$, Full scored significantly higher in isomorphic post-test than in pre-test: $F(1, 36) = 11.0$, $p = .0015$, $\eta = 0.363$.

Learning Performance: The pairwise t-tests showed that there is no significant difference between the $\text{Critical}_{\text{opt}}$ and Full conditions on the two learning metrics, isomorphic NLG: $t(112) = 1.00$, $p = .32$, $d = 0.23$, full NLG: $t(112) = 1.24$, $p = .217$, $d = 0.29$. It implied that only carrying out optimal actions in critical states can be as effective as a fully-executed policy.

Furthermore, to determine whether these null results are significant, that is, the $\text{Critical}_{\text{opt}}$ is indeed perform as effective as Full, we calculated the effect size on all the comparisons and we found that they are all not statistically significant in that $\beta < 0.8$. On the other hand, across all the comparisons, $\text{Critical}_{\text{opt}}$ was slightly better than the Full. This result suggests that if we have enough population samples, the former can outperform the latter.

Time on Task and Percentage of Critical States: A pairwise t-test analysis revealed that the $\text{Critical}_{\text{opt}}$ condition spend a similar amount of time as the Full condition in the training phase: $t(112) = 0.42$, $p = .678$, $d = 0.10$. Pairwise t-tests showed that the $\text{Critical}_{\text{opt}}$ condition has significantly more critical states than the Full condition in the problem level: $t(112) = 2.02$, $p = .046$, $d = 0.46$ but no difference in the step level: $t(112) = -0.29$, $p = .769$, $d = 0.07$. The result suggests that the optimal actions in the non-critical states could reduce the chance of entering critical states.

Tutor Decisions: For the problem level, the $\text{Critical}_{\text{opt}}$ condition received significantly more FWE: $t(112) = 6.91$, $p < .001$, $d = 1.59$, fewer WE: $t(112) = -7.50$, $p < .001$, $d = 1.72$ decisions than the Full condition, but no difference on PS: $t(112) = 0.72$, $p = .472$, $d = 0.17$. For the step level, the $\text{Critical}_{\text{opt}}$ condition received significantly more elicit: $t(112) = 5.50$, $p < .001$, $d = 1.26$ and more tell: $t(112) = 5.83$, $p = .003$, $d = 1.34$ than the Full condition. The results suggest that the random actions in non-critical states could lead the RL policy to give more FWE and fewer WE in critical states.

7 Conclusion

In this study, we evaluated the effectiveness of the Critical-RL framework in identifying critical decisions through an empirical classroom study. Specifically, we compared the $\text{Critical}_{\text{opt}}$ policy with two baseline policies: a $\text{Critical}_{\text{sub}}$ policy and a Full policy. The comparisons are based upon two hypotheses: 1) optimal actions must be carried out in critical states (the Necessary Hypothesis), 2) only carrying out optimal actions in critical states can be as effective as the fully-executed policy (the Sufficient Hypothesis). The result shows that in terms

of students' learning performance, 1) the $\text{Critical}_{\text{opt}}$ condition significantly outperforms the $\text{Critical}_{\text{sub}}$ condition; 2) more importantly, the former performs as effective as the Full condition. It suggests that our Critical-RL framework indeed identifies the critical decisions and satisfies the two hypotheses that 1) taking optimal actions in the identified critical states is significantly more effective than taking suboptimal actions; 2) only taking optimal actions during the critical moments can be as effective as taking optimal actions in every moment.

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

References

1. Andrychowicz, M., Baker, B., et al.: Learning dexterous in-hand manipulation. arXiv preprint arXiv:1808.00177 (2018)
2. Ausin, M.S., Azizsoltani, H., Barnes, T., Chi, M.: Leveraging deep reinforcement learning for pedagogical policy induction in an intelligent tutoring system. In: EDM (2019)
3. Sanz Ausin, M., Maniktala, M., Barnes, T., Chi, M.: Exploring the impact of simple explanations and agency on batch deep reinforcement learning induced pedagogical Policies. In: Bittencourt, I.I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds.) AIED 2020. LNCS (LNAI), vol. 12163, pp. 472–485. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52237-7_38
4. 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)
5. Barto, A.G., Mahadevan, S.: Recent advances in hierarchical reinforcement learning. Discret. Event Dyn. Syst. **13**(1–2), 41–77 (2003). https://doi.org/10.1023/A:1022140919877
6. Beck, J., Woolf, B.P., Beal, C.R.: Advisor: a machine learning architecture for intelligent tutor construction. In: AAAI/IAAI, pp. 552–557 (2000)
7. Chi, M., VanLehn, K., Litman, D., Jordan, P.: Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. User Model. User-Adap. Inter. **21**(1–2), 137–180 (2011). https://doi.org/10.1007/s11257-010-9093-1
8. Clouse, J.A.: On integrating apprentice learning and reinforcement learning. Ph.D. thesis (1996)
9. Fachantidis, A., Taylor, M.E., Vlahavas, I.P.: Learning to teach reinforcement learning agents. Mach. Learn. Knowl. Extract. **1**, 21–42 (2017)
10. Iglesias, A., Martínez, P., Aler, R., Fernández, F.: Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems. Knowl.-Based Syst. **22**(4), 266–270 (2009)
11. Ju, S., Zhou, G., Azizsoltani, H., Barnes, T., Chi, M.: Identifying critical pedagogical decisions through adversarial deep reinforcement learning. In: EDM (2019)
12. Ju, S., Zhou, G., Barnes, T., Chi, M.: Pick the moment: identifying critical pedagogical decisions using long-short term rewards. In: EDM (2020)
13. Li, J., Daw, N.D.: Signals in human striatum are appropriate for policy update rather than value prediction, **31** (2011)

14. 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)
15. McClure, S.M., Laibson, D.I., Loewenstein, G., Cohen, J.D.: Separate neural systems value immediate and delayed monetary rewards. Science **306**, 503–507 (2004)
16. McLaren, B.M., Isotani, S.: When is it best to learn with all worked examples? In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS (LNAI), vol. 6738, pp. 222–229. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21869-9_30
17. McLaren, B.M., Lim, S.J., Koedinger, K.R.: When and how often should worked examples be given to students? New results and a summary of the current state of research. In: CogSci, pp. 2176–2181 (2008)
18. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**, 529–533 (2015)
19. Morris, G., Nevet, A., Arkadir, D., Vaadia, E., Bergman, H.: Midbrain dopamine neurons encode decisions for future action. Nat. Neurosci. **9**(8), 1057–1063 (2006)
20. Najar, A.S., Mitrovic, A., McLaren, B.M.: Adaptive support versus alternating worked examples and tutored problems: which leads to better learning? In: Dimitrova, V., Kuflik, T., Chin, D., Ricci, F., Dolog, P., Houben, G.-J. (eds.) UMAP 2014. LNCS, vol. 8538, pp. 171–182. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08786-3_15
21. Narasimhan, K., Kulkarni, T., Barzilay, R.: Language understanding for text-based games using deep reinforcement learning. arXiv preprint arXiv:1506.08941 (2015)
22. Rafferty, A.N., Brunskill, E., et al.: Faster teaching via POMDP planning. Cogn. Sci. **40**(6), 1290–1332 (2016)
23. 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)
24. Roesch, M.R., Calu, D.J., Schoenbaum, G.: Dopamine neurons encode the better option in rats deciding between different delayed or sized rewards. Nat. Neurosci. **10**(12), 1615–1624 (2007)
25. Rowe, J., Mott, B., Lester, J.: Optimizing player experience in interactive narrative planning: a modular reinforcement learning approach. In: Tenth Artificial Intelligence and Interactive Digital Entertainment Conference (2014)
26. Rowe, J.P., Lester, J.C.: Improving student problem solving in narrative-centered learning environments: a modular reinforcement learning framework. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS (LNAI), vol. 9112, pp. 419–428. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19773-9_42
27. Shen, S., Ausin, M.S., Mostafavi, B., Chi, M.: Improving learning & reducing time: a constrained action-based reinforcement learning approach. In: UMAP (2018)
28. Shen, S., Chi, M.: Aim low: correlation-based feature selection for model-based reinforcement learning. In: EDM (2016)
29. Shen, S., Chi, M.: Reinforcement learning: the sooner the better, or the later the better? In: Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, pp. 37–44. ACM (2016)
30. Shen, S., Mostafavi, B., Lynch, C., Barnes, T., Chi, M.: Empirically evaluating the effectiveness of POMDP vs. MDP towards the pedagogical strategies induction. In: Penstein Rosé, C., et al. (eds.) AIED 2018. LNCS (LNAI), vol. 10948, pp. 327–331. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93846-2_61
31. Silver, D., et al.: Mastering the game of go with deep neural networks and tree search. Nature **529**(7587), 484–489 (2016)

32. Silver, D., Hubert, T., Schrittwieser, J., et al.: A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science **362**(6419), 1140–1144 (2018)
33. Stamper, J.C., Eagle, M., Barnes, T., Croy, M.: Experimental evaluation of automatic hint generation for a logic tutor. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS (LNAI), vol. 6738, pp. 345–352. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21869-9_45
34. Sul, J.H., Jo, S., Lee, D., Jung, M.W.: Role of rodent secondary motor cortex in value-based action selection. Nat. Neurosci. **14**(9), 1202–1208 (2011)
35. Sutton, R.S., Precup, D., Singh, S.: Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. Artif. Intell. **112**(1–2), 181–211 (1999)
36. Torrey, L., Taylor, M.E.: Teaching on a budget: agents advising agents in reinforcement learning. In: International conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2013, pp. 1053–1060 (2013)
37. Van Gog, T., Kester, L., Paas, F.: Effects of worked examples, example-problem, and problem-example pairs on novices' learning. Contemp. Educ. Psychol. **36**(3), 212–218 (2011)
38. Wang, P., Rowe, J., Min, W., Mott, B., Lester, J.: Interactive narrative personalization with deep reinforcement learning. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (2017)
39. Zhou, G.: Big, little, or both? Exploring the impact of granularity on learning for students with different incoming competence. In: CogSci (2019)
40. Zhou, G., et al.: Towards closing the loop: bridging machine-induced pedagogical policies to learning theories. In: EDM (2017)
41. Zhou, G., Azizsoltani, H., Ausin, M.S., Barnes, T., Chi, M.: Hierarchical reinforcement learning for pedagogical policy induction. In: Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., Luckin, R. (eds.) AIED 2019. LNCS (LNAI), vol. 11625, pp. 544–556. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23204-7_45
42. Zhou, G., Price, T.W., Lynch, C., Barnes, T., Chi, M.: The impact of granularity on worked examples and problem solving. In: CogSci, pp. 2817–2822 (2015)
43. 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)
44. Zimmer, M., Viappiani, P., Weng, P.: Teacher-student framework: a reinforcement learning approach. In: AAMAS Workshop Autonomous Robots and Multirobot Systems (2013)