How Much Training is Needed? Reducing Training Time using Deep Reinforcement Learning in an Intelligent Tutor *

Nazia Alam
North Carolina State University
nalam2@ncsu.edu

Behrooz Mostafavi
North Carolina State University
bzmostaf@ncsu.edu

Sutapa Dey Tithi
North Carolina State University
stithi@ncsu.edu

Min Chi
North Carolina State University
mchi@ncsu.edu

Tiffany Barnes
North Carolina State University
tmbarnes@ncsu.edu

ABSTRACT

Reducing learning time or training time in intelligent tutors is a challenging research problem. In this study, we aim to reduce training time while maintaining student performance in intelligent tutors. We propose a Deep Reinforcement Learning (DRL) based method to determine when students need more training and when they don't. We design an adaptive pedagogical policy using the DRL method to reduce training time in intelligent tutors. We incorporate the pedagogical policy within an intelligent logic tutor. Based on the DRL method outcome, the pedagogical policy determines when to provide more training problems to students, and when training problems can be skipped. We conduct a study to examine the effectiveness of the proposed policy against a control without any adaptive interventions. We also evaluate the proposed policy against a comparison condition that provides training problems or worked examples during training based on a DRL policy, and a non-adaptive condition that also may provide or skip training problems for students. Our analysis provides empirical evidence that the proposed policy reduced training time compared to the control condition. Results show that the proposed policy is more effective compared to the comparison conditions in reducing training time and at the same time, maintaining student performance. Overall, the study demonstrates the efficacy of the proposed pedagogical policy.

Keywords

Deep Reinforcement Learning, Intelligent Tutoring System, Educational Data Mining

N. Alam, B. Mostafavi, S. D. Tithi, M. Chi, and T. Barnes. How much training is needed? reducing training time using deep reinforcement learning in an intelligent tutor. In B. Paaßen and C. D. Epp, editors, *Proceedings of the 17th International Conference on Educational Data Mining*, pages 251–261, Atlanta, Georgia, USA, July 2024. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.12729806

1. INTRODUCTION

Intelligent tutoring systems (ITS) provide personalized instruction, feedback, and assistance to students. Most current research focuses on improving student performance and learning gain in intelligent tutors. However, total tutor time and training time or learning time are other important measures. Total tutor time refers to the total time taken to complete a tutor. Training time or learning time refers to the total time taken to complete the training section of a tutor. Learning efficiency refers to the ratio of learning gain and total tutor time. Reducing tutor time is a relatively under-explored area.

ITSs are widely used in classrooms nowadays as they help students learn without human instruction and intervention. With widespread use, reducing tutor time and improving learning efficiency in ITSs is a topic of significant importance. Students have a limited amount of time that they devote to studying and learning different subjects. Reducing tutor time and increasing learning efficiency can help students spend less time using an ITS while maintaining the same learning and performance, enabling them to use their saved time to learn something else. All the students who use an ITS may not have the same knowledge or grasp of all the concepts the tutor covers. If a student is good at solving problems involving certain concepts, reducing the amount of training problems can save students time without negatively affecting their performance. More than necessary training may not be helpful for a student and may not be a good use of learning time. At the same time, a lengthy tutor may cause frustration in students.

In this study, we aim to focus on the challenging task of reducing training time. We develop and evaluate a pedagogical policy to reduce training time in an intelligent tutor teaching the open-ended domain of logic proofs. Deep reinforcement learning (DRL) has been successfully used in ITSs for inducing pedagogical policies [2, 4, 14]. In this study, we investigate the use of a DRL-based pedagogical policy to determine when students need more training and when students don't. Based on the policy, the students will be adaptively given training problems, or a training problem may be skipped for them if the policy determines that

the student does not need more training. Thus, the policy attempts to provide students with an optimized and personalized training section. To the best of our knowledge, no prior work has utilized DRL to provide training problems adaptively in such a way and evaluated the impact on training time and efficiency on an ITS. We integrated the DRL policy within a logic tutor's training section and conducted a controlled experiment. Our goal was to evaluate the DRL policy's efficacy. Therefore, we designed a number of control and comparison conditions. We implemented 1) A control condition where all training problems were given to all students. 2) Adaptive condition where the proposed DRL policy was used to adaptively provide or skip training problems for students. 3) A comparison condition, where a non-adaptive policy (in this case, a random policy) was implemented to provide or skip training problems to students. 4) Finally, we wanted to understand the effectiveness of skipping a problem compared to providing a worked example. Therefore, we implemented another comparison condition, where the same DRL policy was used to adaptively provide a problem or worked exampled in the training section.

Overall, the main contributions of this study are as follows: we propose a new pedagogical policy to reduce training time in tutors and implement the policy by utilizing a DRL model. We investigate the impact of the proposed policy on an intelligent logic tutor. We inspect the effectiveness of the new adaptive policy against a control policy without the adaptive components and also against two other comparison policies.

The rest of the paper is organized as follows. In Section 2, we discussed some related works. We provide the tutor context and methodology in Section 3. Then, we present the experimental design in Section 4. The results and related discussions are presented in Sections 5 and 6. Then, the conclusion is given in Section 7 with limitations and future work in Section 8.

2. RELATED WORKS

2.1 Reducing Tutor Time

Reducing tutor time and improving efficiency is a relatively under-explored area. In one study, Cen et al. [5] investigated the use of Learning Factors Analysis (LFA) to improve efficiency and reduce tutor time in a geometry tutor. Using a statistical model, LFA can identify over-practiced and under-practiced knowledge components (KC). The authors optimized the curriculum of the geometry tutor to improve learning efficiency. Compared to a control group, it was found that the optimized curriculum group saved significant time and showed no significant difference in performance in the posttest or retention test. To use LFA, the problems in the tutor need to be split based on the KC. For our tutor, the problems consist of multiple KC's. In another study, Shen et al. [15] proposed a data-driven framework called Constrained Action-based Partially Observable Markov Decision Process (CAPOMDP) to induce effective pedagogical policies. They developed a policy called $CAPOMDP_{Time}$ using time as a reward for reducing students' time on task. The policy determines whether to provide students with worked examples (WE), or they should do problem-solving (PS) with a logic tutor. Results showed no significant difference among the high incoming competence groups. However, for the low incoming competence groups, students in the $CAPOMDP_{Time}$ condition spent significantly less time than those using the baseline policies. In [11], authors worked with ALEKS ("Assessment and Learning in Knowledge Spaces"), which is an adaptive learning and assessment system based on knowledge space theory. They tried to improve the efficiency of ALEKS assessment. Their goal was to develop an algorithm to predict when the assessment should be stopped. They developed a recurrent neural network classifier to predict the final result of each assessment. They used the classifier to develop a stopping algorithm. Results showed that potentially the length of the assessment can be reduced by a large amount using the stopping algorithm, thereby reducing the time taken for assessment while maintaining a high level of accuracy.

2.2 Use of Reinforcement Learning in Intelligent Tutoring Systems

Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) have been widely used in intelligent tutoring systems to induce pedagogical policies. In [3], authors used a DRL-based pedagogical policy to determine when to provide proactive help in an intelligent logic tutor. The proposed policy provided next-step proactive hints based on the prediction of the DRL model. Abdelshiheed et al. [1, 2] used DRL policy to induce and deploy metacognitive interventions in a logic tutor. The authors found that the DRL policy closed the metacognitive skills gap between students and prepared the students for future learning. In another work [6], authors proposed a pedagogical modeling framework using DRL to induce policies to provide ICAP-inspired scaffolding in adaptive learning environments. Results showed that adaptive scaffolding policies induced with DRL outperformed the baseline policies. Ausin et al. [4] used a DRL policy to determine when to provide worked examples and when students should do problem-solving with an intelligent tutor. They found that when combined with inferred rewards, the DRL policy outperformed the baseline policy. In another work [14], Ausin et al. conducted studies where DRL was used to decide whether to provide worked examples or problems, along with simple explanations to share the decision with students. Results demonstrated that the DRL policy with simple explanations significantly improved students' learning performance compared to an expert policy. Zhou et al. [19] used a hierarchical RL policy to make decisions at both problem and step levels in an ITS. They found that the proposed policy was significantly more effective than the baseline policies. Ju et al. [9] proposed a DRL framework to identify critical decisions and induce critical policies in an ITS. They evaluated the framework in terms of necessity and sufficiency. Results confirmed that the framework met both criteria.

Overall, to the best of our knowledge, no prior work utilized DRL policy to provide or skip training problems in intelligent tutors to reduce training time. Researchers have investigated how to combine PS and WE to improve learning efficiency, but rarely studied skipping problems, which requires either a student knowledge model or other mastery learning mechanism. However, in domains without such mechanisms, the proposed approach can apply DRL to prior student data to improve learning efficiency.

3. METHOD

In this study, we propose a pedagogical policy to reduce student training time in an intelligent tutor. We develop the policy using Deep Reinforcement Learning and incorporate it in the tutor to decide, for each problem, whether students should solve it or skip it. In this section, we first discuss our logic proof tutor. Then, we describe the proposed policy. Finally, we present our research questions for the study.

3.1 Tutor

Our intelligent logic tutor [16, 17, 10], shown in Figure 1, is for learning and practicing formal propositional logic proofs in a Discrete Mathematics course. The main workspace is on the left which contains the logic statements as nodes. For each logic proof problem, a set of logic statements are given as premises, shown at the top of the workspace, and a conclusion to be derived is shown at the bottom. In the bottom left of Figure 1, there is a 'get hint' button which students can click to request hints. The logic rules are given in the middle pane. The right pane contains instructions, guidance, and directions for the students. The tutor consists of two types of problems: worked examples (WE) and problem-solving (PS). The WE problems are solved by the tutor; students click the next step button to see the step-by-step solution. The PS problems are to be solved by students by deriving new statements until the conclusion is reached. The students can use forward, backward, or indirect strategies to solve the PS problems. The forward strategy is where the students start with the given premises and derive new statements towards deriving the conclusion. In the backward strategy, students start with the given conclusion and derive new statements towards the premises. In the indirect strategy, students prove by contradiction, where the tutor places the negation of the conclusion in the proof, and students work toward deriving a contradiction.

The tutor consists of four sections: introduction, pretest, training, and posttest. The introduction contains two worked examples to familiarize the students with the tutor. Then, in the pretest section, the students are given two problems to solve. Next is the training section, with five levels increasing in difficulty and each level adding new rules. Each training level has three training problems and a level-end posttest problem. Finally, there is a posttest section with six problems. Students may request hints during the training but not in the pre-test, level-end posttest, or final posttest problems. In total, students solve up to thirty problems in the tutor. Each problem has a problem score based on time, number of steps, and accuracy; problems with lower time, lower steps, and higher accuracy have higher scores. Pretest and posttest scores are the average of all the problem scores in each respective section.

3.2 Proposed Policy and Model

3.2.1 Proposed Policy

Our overarching research question is whether or not a DRL policy for choosing between PS and skipping a problem can reduce student time while keeping learning the same or better. Our investigation of student data showed that students were not spending significant time interacting with the WEs provided. Therefore, we hypothesized that we could learn a policy from our prior data that could choose to skip problems instead of showing them as WEs for students, and further hypothesized that this policy could save students time when compared with the same policy that provides WEs instead of skipping problems. Therefore, we used our prior data, as described in more detail below, to learn an offline, off-policy DRL policy to decide on each problem, when students should solve it themselves (PS) or skip it.

Students are randomly assigned to one of 4 conditions: DRL-skip, DRL-WE, all-PS, and Random-skip. In both DRL conditions, the same proposed adaptive DRL policy, optimized for student learning and efficiency, provides interventions during the tutor's training section. For each problem, the DRL-skip policy is used to choose between problem-solving (PS) and skipping the problem. The DRL-WE policy chooses between PS and a WE. Students in the all-PS condition solve all the problems using PS. The Random-skip policy randomly assigns students to solve each problem as PS or skip it, subject to the tutor rule that every level must have at least one PS problem.

3.2.2 Proposed Model

Reinforcement Learning (RL), especially Deep Reinforcement Learning (DRL), has been used successfully by different intelligent tutors. The main objective of an agent in RL is to take specific actions to maximize cumulative rewards. In DRL, deep neural networks are used to make decisions. Among the different RL algorithms, one of the most successful is Q-learning, a model-free RL algorithm [18]. In the Q-learning algorithm, a q-value is calculated which is the expected cumulative reward for a particular action taken in a state. Deep q-learning methods make use of neural networks as function approximators. DQN [12] is an off-policy Q-learning model where the same neural network is used for action selection and action evaluation, which can cause an overestimation of values. Double-DQN (DDQN) [7] solves this problem by using two different neural networks for action selection and action evaluation. Here, two identical neural networks are used called the online network (θ) and the target network ($\bar{\theta}$). The online network is trained in each iteration which is used to select the next action ($a \in A$) with the highest q-value for the next state ($s' \in S$). The target network is updated periodically which is used for the evaluation of q-value of actions. The q-values are updated using the following equation:

$$Q(s, a; \theta) = r + \gamma Q(s', argmax_{a \in A} Q(s', a'; \theta); \bar{\theta})$$

Here, $Q(s, a; \theta)$ is q-value which is calculated using θ in state s after taking action a. The reward r is defined below, and γ is the discount factor that determines the importance of future rewards. Our study uses an off-policy and offline version of the DDQN model for our experiments.

To train the DRL model, we used tutor data from random PS/WE conditions from the previous four semesters (Fall 2019, Spring 2019, Spring 2022, Spring 2023). We look at the time taken for WE problems and based on that, set a threshold of 100 seconds. If a student takes less than the threshold in a WE problem, we consider it as a skipped problem. We formulate the problem of determining when to give a problem and when to intervene as a Markov Decision

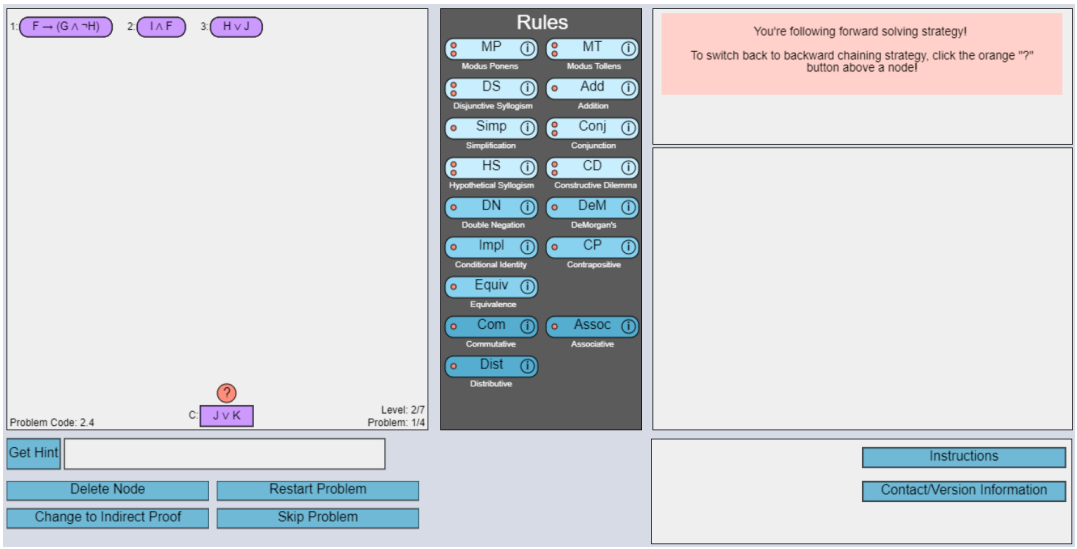

Figure 1: Tutor user interface

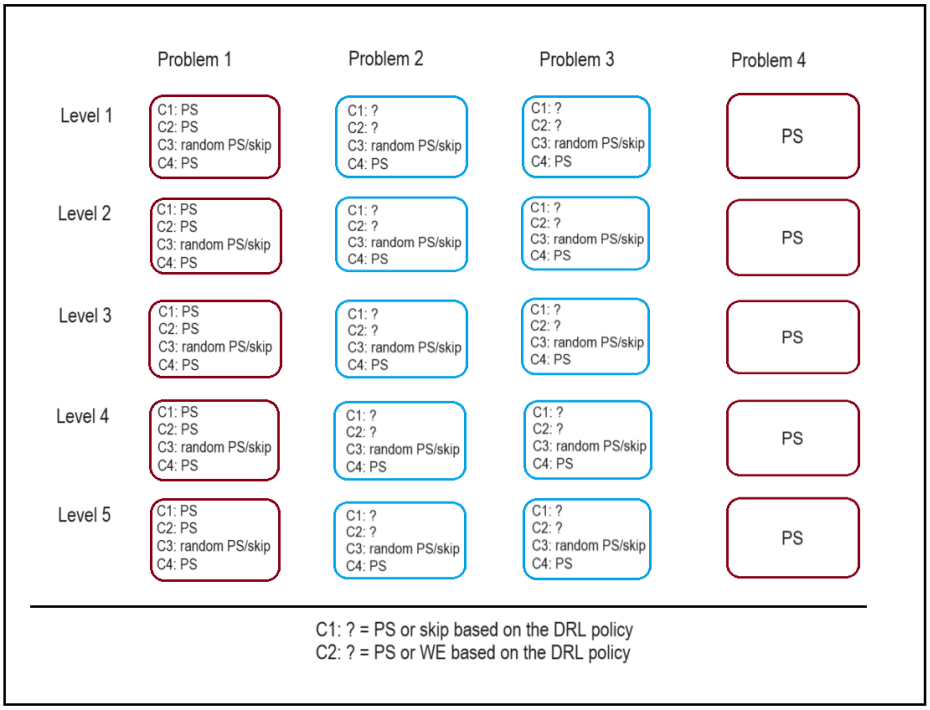

Figure 2: Problem organization in the training levels for the four training conditions

Table 1: Number of participants in different conditions

Condition	Number of students assigned	Number of students who completed the tutor
DRL-skip (C1-Skip)	63	52
All-PS (C4-all-PS)	64	58
DRL-WE (C2-WE)	60	49
Random-skip (C3-Random)	63	53
Total	250	212

Process with a specific state, action, and reward.
State: The state consists of 75 student log features that describe students' interaction with the tutor. Some example features include those related to time (totalTime, totalPSTime, totalWEtime, idleTime), actions (actionCount, forwardActionCount, backwarddActionCount, directProofActionCount, indirectProofActionCount), rule scores for each rule, accuracy (rightApp, wrongApp, RightAppRatio, WrongAppRatio), and steps (stepCount, avgStepTime).
Action: At a training problem of the tutor, there are two possible actions: 1) provide a PS problem and 2) intervene to skip or show WE.
Reward: We want to reduce training time and, at the same time, maintain student performance. Therefore, we formalize our reward as defined below. The reward is a combination of the post-test score and the problem time using the following equation:

$$Reward = posttestScore * (1 - problemTime)$$

The reward will be high when the posttestScore is high, and problemTime will be low, and vice versa. The problemTime is 0 when a problem is skipped. We conducted hyperparameter tuning using a grid search strategy to select the parameters. The final model contained three hidden layers with 32, 64, and 32 neurons, respectively. The learning rate was 0.001, and the value of γ was 0.99.

3.3 Specific Research Questions

- ***RQ1:*** How is the performance of students who got a high number of interventions in the adaptive DRL-skip condition?
- ***RQ2(a):*** Do the adaptive condition students have similar posttest performance compared to the control all-PS condition?
- ***RQ2(b):*** Does the adaptive DRL-skip condition take less time compared to the control condition?
- ***RQ3:*** How effective is the DRL-skip policy compared to a DRL-WE policy providing worked examples (WE) instead of skipping problems?
- ***RQ4:*** How effective is the proposed adaptive policy compared to a non-adaptive Random-skip policy that also skips training problems?
- ***RQ5:*** How does the adaptive policy impact students' problem-solving behavior?

To answer these research questions, we will examine students' post-test performance and time taken in different conditions. We will look at their learning gain and learning efficiency as defined in Section 4.3. We will do significance analysis to compare the conditions. We will also investigate the impacts of the policies on students' problem-solving behaviors, such as hint usage.

4. EXPERIMENT

4.1 Experimental Design

We designed four training conditions to measure the proposed policy's effectiveness. The conditions are as described below:
Control all-PS condition: Students assigned to the control condition will receive only PS problems in the training levels.
Adaptive DRL-skip condition: Students assigned to this condition will receive PS problems in the training levels. However, some PS problems can be skipped as determined by the DRL policy.
Adaptive DRL-WE condition: Students assigned to this condition will receive PS problems or WE problems as determined by the DRL policy in the training levels.
Random-skip condition: Students assigned to this condition will receive PS problems at the training level. However, some PS problems will be skipped using a non-adaptive policy, which is in this case a random policy.
Figure 2 shows the problem organization in the training levels for the four training conditions. Students in all conditions can request hints on the training level PS problems. The hints are shown as messages at the bottom of the tutor interface.

4.2 Participants

This research study was approved by the university internal review board. The tutor was given as an assignment in an undergraduate Discrete Mathematics course at a US University in the Fall 2023 semester. The students were placed in different conditions using random sampling. As we can see in Table 1, a total of 212 students completed the tutor. We performed Fisher's exact test between the adaptive DRL-skip condition and the other conditions to check for significant differences in tutor completion, but no significant differences were found ($p - value > 0.05$).

4.3 Performance Metrics

We look at the time taken to solve the problems. The time is calculated by the total time taken to complete the different sections of the tutor. As the students may leave the tutor open when inactive, we cap each action to 1 minute.

Table 2: Time comparison in the DRL-skip condition high and low intervention groups

	Mean Time (Std Dev) in minutes			
	DRL-skip	DRL-skip (upper median)	DRL-skip (lower median)	P-value
Training	38.54 (23.36)	25.671 (17.17)	52.43 (21.08)	**<0.001**
Final posttest	44.84 (32.84)	39.18 (29.12)	50.95 (35.44)	0.389
Total tutor time	194.33 (88.01)	153.53 (66.14)	238.40 (87.45)	**<0.001**

Table 3: Posttest performance in DRL-skip condition high and low intervention group

	DRL-skip (upper median)	DRL-skip (lower median)	P-value
Posttest Score	0.746	0.622	**0.002**
Pretest Score	.75	0.576	**<0.001**
NLG	-0.074	0.056	0.128
LE	0.329	0.163	**<0.001**

We take the sum of the times of all actions to calculate the total time. For example, training section time is the sum of the times of all actions taken by students to complete the training problems.
Each problem is given a problem score, calculated based on the number of steps (fewer is better), problem solving time (lower is better), and rule accuracy (higher is better). The pretest and posttest scores are calculated by taking the average problem scores of the pretest and posttest section problems, respectively. We examine the Normalized Learning Gain (NLG) to understand student learning. We calculate students' NLG [8] using the following function:

$$NLG = \frac{posttest - pretest}{\sqrt{1 - pretest}}$$

We calculate students' Learning efficiency (LE) by first scaling the NLG scores between 0 and 1 and using the following equation:

$$LE = \frac{NLG}{Total\ tutor\ time\ (in\ hour)}$$

We use hint justification as a measure of hint usage. A hint is said to be justified if the hinted statement is derived using the existing statements. The hint justification rate (HJR) is the ratio of the number of hints that have been justified and the total number of hints given.

5. RESULTS

In this section, we examine the students' posttest scores, normalized learning gain, and time taken in different sections of the tutor to answer our research questions. We found the data to be non-normally distributed, therefore we use the Mann-Whitney test to conduct significance analysis.

5.1 RQ1: How is the performance of students who got a high number of interventions in the adaptive DRL-skip condition?

We start by finding some statistics on the DRL-skip condition students. We find the average number of problems received by the students to be 24. The maximum number of skip interventions (where each intervention is skipping a problem) was 9, the minimum number of interventions was 0, and the median number of interventions was 6. We are especially interested to learn more about the students who got a high number of interventions and whether they had any negative impact on the students. Therefore, we divide the adaptive DRL-skip condition students into high and low-intervention groups based on the median number of interventions to investigate the impact of receiving interventions.

We look at the time taken by high and low intervention students in the different sections of the tutor. Table 2 shows the mean time and standard deviation of time taken by the high and low intervention group students in the DRL-skip condition in minutes for different sections of the tutor. It also shows the result of the Mann-Whitney test by showing the corresponding p-values. By conducting a significance analysis, we find that students in the upper median took significantly less time in the training section of the tutor and also in the total tutor.

Next, we look at the performance of the DRL-skip students in high and low intervention groups using their posttest problem scores and NLG. From Table 3, we find that the upper median students had a significantly better average posttest score compared to the lower median group's average posttest score. Note that upper median students also had a significantly better pretest score compared to the lower median group. We did not find any significant difference in NLG between the two groups. Then, when we look at learning efficiency (LE), we find that upper median students had significantly better LE, implying that they could achieve similar mastery by spending significantly less time in training and posttest problems than the lower median group spent. We did not find anything to suspect that getting a high number of interventions had any negative impact on students.

Finally, we separate the DRL-skip students who got the

Table 4: Comparing student data from the first problem in each level for students with the highest number of interventions and students with the lower number of interventions

	Data from first problem in the level							
	Student with highest intervention				Student with lowest intervention			
Training level	stepCount	totalTime	rightApp	wrongApp	stepCount	totalTime	rightApp	wrongApp
2	5	72	2	0	12	650	6	37
3	10	189	6	0	18	459	15	6
4	5	148	5	1	17	254	13	12
5	7	93	5	1	6	95	3	0
6	7	128	4	1	7	228	4	0

Table 5: Posttest performance in DRL-skip condition and the control (all-PS) condition

	Condition		
Metric	DRL-skip (C1)	Control (C4)	P-value
Posttest Score	0.686	0.685	0.855
Pretest Score	0.667	0.659	0.876
NLG	-0.011	-0.037	0.659
LE	0.261	0.251	0.797

Table 6: Time comparison in DRL-skip condition and the control (all-PS) condition

	Mean Time (Std Dev) in minutes		
	DRL-skip(C1)	Control(C4)	P-value
Training	38.54 (23.36)	59.99 (32.18)	**<0.001**
Final posttest	44.84 (32.84)	46.37 (38.36)	0.681
Total tutor time	194.33 (88.01)	213.34 (99.99)	0.424

highest number of interventions and students who got the lowest number of interventions. We examine the data from the first problem of each training level to determine the reason behind the difference in the number of interventions. Table 4 shows some key student features for the first problem in each training level for the DRL-skip students with the highest number of interventions and the students with the lowest number of interventions. Here, stepCount refers to the total number of steps taken to complete the problem, totalTime is the total time taken to complete the problem, rightApp is the number of times students correctly applied the rules, and wrongApp is the number of times students incorrectly tried to apply the rules. We find that the DRL-skip students with the highest number of skipped problems generally had good values for the features in the first problem in the levels, which were used as features for the DRL-skip policy. Therefore, it makes sense that the policy decided that for this student, more problems can be skipped. Meanwhile, the DRL-skip students who had the lowest number of skipped problems had lower performance metric values in the first problem in most of the levels. Therefore, the policy decided not to skip the next problem, since metrics show a need for more training and practice.

Table 7: Posttest performance in adaptive DRL-skip condition and the DRL-WE condition

	Condition		
Metric	DRL-skip (C1)	DRL-WE(C2)	P-value
Posttest Score	0.686	0.671	0.444
Pretest Score	0.667	0.657	0.841
NLG	-0.011	-0.100	0.981
LE	0.261	0.224	0.488

Table 8: Time comparison in adaptive condition and the DRL-WE condition

	Mean Time (Std Dev) in minutes		
	DRL-skip (C1)	DRL-WE (C2)	P-value
Training	38.54 (23.36)	38.04 (16.51)	0.721
Final posttest	44.84 (32.84)	59.47 (38.05)	**0.028**
Total tutor	194.33 (88.01)	217.49 (59.61)	0.091

5.2 RQ2(a): Do the DRL-skip adaptive condition students have similar posttest performance compared to the control condition?

Here, we examine the performance of the adaptive DRL-skip condition students compared to the control all-PS condition students. In the control all-PS condition, students received all the PS problems.

Students' performance in terms of posttest score, pretest score, NLG, and LE are presented in Table 5. We conducted a significance analysis and found no significant difference in the posttest scores of the two groups. We also did not find any significant difference between the two groups in terms of pretest score, NLG, and LE.

5.3 RQ2(b): Does the adaptive DRL-skip condition take less time compared to the control condition?

Here, we examine the time taken by adaptive DRL-skip condition students in different sections of the tutor and compare it to the control All-PS condition.

The mean and standard deviation of time taken by the adaptive DRL-skip and control all-PS condition students in the training, posttest, and total tutor is given in Table 6. The training time in the adaptive DRL-skip condition is significantly less compared to the control all-PS condition. We do not find any significant difference in post-test or total tutor time between the two conditions.

5.4 RQ3: How effective is the DRL-skip policy compared to a DRL-WE policy providing worked examples (WE) instead of skipping problems?

Here, we compare the adaptive DRL-skip condition to the DRL-WE policy, where students either received PS problems or WE problems based on the same learned DRL policy, but provided WEs instead of skipping problems.

First, we investigate the student performance and learning in both conditions. Table 7 shows the posttest score, pretest score, NLG, and LE in two conditions. We conduct significance analysis using the Mann-Whitney test for comparison, and corresponding p-values are also provided in Table 7. We do not find any significant difference in post-test score, pretest score, NLG, or LE between the conditions ($p - value > 0.05$).

Next, we look at the time taken by students in both the DRL conditions. The mean and standard deviation of time can be found in Table 8 for the two conditions. We find that there is no significant difference between the two conditions in terms of training time or total tutor time. We note that the training time of DRL-WE is 38.04 minutes, which is similar to the DRL-skip condition; however, some of that time is spent on worked examples instead of problem-solving. We further examine the time students spent on problem-solving during training in the DRL-WE condition, and find that students spent on average 30.31 minutes on problem-solving, and 7.72 minutes on worked examples. Interestingly, we find that there is a significant difference in posttest section time, where the adaptive DRL-skip condition students did significantly better than the DRL-WE condition students.

5.5 RQ4: How effective is the proposed adaptive policy compared to a non-adaptive policy that also skips training problems?

Here, we compare the adaptive condition to a non-adaptive policy that randomly provided students with either a PS problem or skipped the problem in the training levels.

First, we analyze the performance and learning of students in the two conditions. Table 9 provided the posttest score, pretest score, NLG, and LE of the students in the two conditions. We do the Mann-Whitney test for significance analysis, and the corresponding p-values are given in Table 9. We do not find any significant difference between the two conditions for posttest score, pretest score, and NLG. However, we find that there is a significant difference in LE between the two conditions, with the adaptive condition having significantly better LE.

Table 9: Posttest performance in adaptive condition and the random-skip condition

	Condition		
Metric	DRL-skip (C1)	Random-skip (C3)	P-value
Posttest Score	0.686	0.644	0.181
Pretest Score	0.667	0.659	0.913
NLG	-0.011	-0.129	0.347
LE	0.261	0.180	**0.002**

Table 10: Time comparison in adaptive condition and the random-skip condition

	Mean Time (Std Dev) in minutes		
	DRL-skip (C1)	Random-skip (C3)	P-value
Training	38.54 (23.36)	40.19 (20.89)	0.59
Final posttest	44.84 (32.84)	50.29 (29.44)	0.13
Total tutor	194.33 (88.01)	219.00 (113.94)	0.5

Next, we compare the time taken by the students in the two conditions. Table 10 shows the mean and standard deviation of time for the two conditions. We do not find any significant difference between the two conditions in training time, posttest time, or total tutor time.

5.6 RQ5: How does the adaptive policy impact students' problem-solving behavior?

Here, we look at the problem-solving behavior of students in the adaptive condition and compare it to the other conditions. First, we look at the hint usage of students in different conditions. Table 11 provides the number of hints per problem, the number of hints justified, and the hint justification rate for the different conditions. When comparing the number of hints per problem, we find that there is a marginal difference between the adaptive condition and the control condition ($p-value = 0.057$). We do not find any significant difference in the number of hints in the adaptive condition compared to the other conditions. Next, we compared the hint justification rate of the adaptive condition to the other conditions, and there were no significant differences.

6. DISCUSSION

When we examined the high and low intervention groups in the adaptive DRL-skip condition, we found that the upper median group took less time in the training section and the overall tutor. They had better posttest scores and pretest scores compared to the lower median group. It can be concluded that receiving a high amount of intervention (skipping a training problem) did not seem to show any negative effect on their posttest performance. Also, as the high intervention group had higher average pretest scores, it is possible that the DRL policy could identify the students with good prior proficiency and provided them with more

Table 11: Hint usage in different conditions

	Mean hints per problem (Std Dev)			
	DRL-skip (C1)	Control (C4)	DRL-WE (C2)	Random-skip (C3)
Number of Hints	0.35 (0.42)	0.54 (0.62)	0.31 (0.38)	0.36 (0.42)
Number of hints justified	0.31 (0.39)	0.49 (0.59)	0.29 (0.35)	0.32 (0.39)
Hint justification rate	0.90	0.91	0.92	0.89

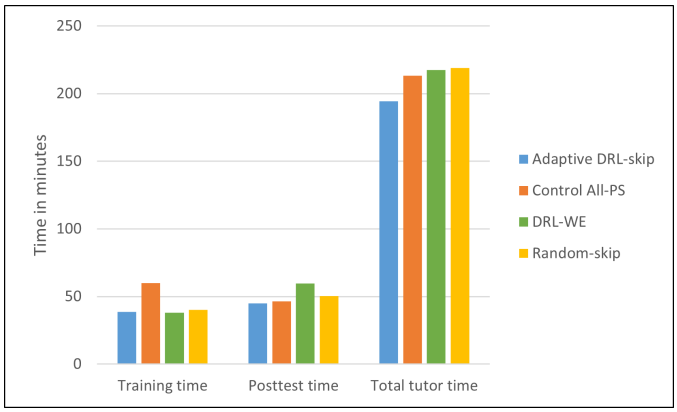

Figure 3: Time comparison among all the conditions

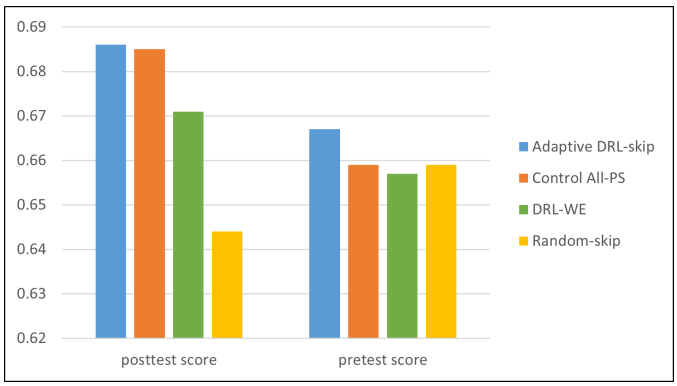

Figure 4: Posttest and pretest performance in all the conditions

interventions.

As we compared the posttest performance of the adaptive condition to the control (all PS) condition, we did not find any significant difference in any of the performance metrics. When comparing time, we found that the adaptive condition had reduced training time compared to the control condition, suggesting that the adaptive policy worked well and reduced training time while maintaining performance, as seen from the comparison with the control condition.

When we compared the posttest performance of the adaptive condition with the DRL-WE condition, we did not find any differences in posttest scores. We also did not find any difference in training time, indicating that students spend a very short time with WE-type training problems. The interesting finding was that the posttest time of the adaptive condition was significantly better compared to the DRL-WE condition. When students are given WEs, they may skip them quickly without learning. Therefore, although they needed a short time in training, it is possible that they did not learn to solve problems effectively, which is why they needed more time in the posttest.

Based on the comparison of the adaptive DRL-skip condition and the non-adaptive random-skip condition, we find no difference in posttest scores or time; however, the adaptive condition had significantly better learning efficiency. This demonstrates the effectiveness of the proposed DRL policy.

Figure 3 shows the time taken in different sections of the tutor for the different conditions. We can see that the adaptive condition took less or similar time in the training section, posttest section, and the total tutor compared to the control condition and the other conditions. From Figure 4 we find that the adaptive condition had a similar posttest or better

posttest score compared to the other conditions. Overall, it can be said that for the adaptive condition students, the training time was reduced while maintaining the posttest performance. The results show that our proposed adaptive DRL policy can be an effective intervention to reduce training time in intelligent tutors. The proposed adaptive DRL-based policy can be easily adopted in other problem-solving domains as well to reduce training time.

7. CONCLUSIONS

In this paper, we proposed a pedagogical policy to determine when to provide training problems and when to skip them in intelligent tutors in order to reduce training time while maintaining student performance. The pedagogical policy utilized a DRL-based method to make decisions. We incorporated the proposed policy in an intelligent logic tutor and conducted a study to evaluate the impact of the policy against three different conditions: a control policy without any adaptive components, a comparison policy that provided worked examples using a DRL policy to reduce training time, and a non-adaptive random policy that also skipped training problems. The study showed that the adaptive policy reduced training time while maintaining student performance compared to the control policy. The findings show that the proposed policy performed better in the posttest in terms of time against the policy that provided worked examples in training using a DRL policy, and had better learning efficiency than the non-adaptive policy that skipped problems in training. Overall, these findings provide insights into the effectiveness of the proposed policy in reducing training time in intelligent tutors.

8. LIMITATIONS AND FUTURE WORK

One limitation of the study is that the results were not corrected for multiple tests. Also, the impact of skips on student motivation was not measured. However, since the tutor always decides how or whether problems are shown to students, skipping problems should not adversely affect student motivation or confidence. An analysis of long-term knowledge retention was beyond the scope of the current study. However, prior research has shown that 50-64% of students completing the tutor average over 80% on a delayed posttest [13]. Another limitation is that DRL-skip policy was not compared to a simpler hand-coded skip policy. However, a hand-coded policy must skip problems for students with high accuracy and/or lower problem-solving times, primarily benefiting students with high prior knowledge. But learning is not a linear process. Using a DRL policy optimized for learning by the end of the tutor allows for variation in the policies to meet individual student needs and potentially save time for more students. The study was done using an intelligent logic tutor. Utilizing only one ITS does not provide further information about the generalizability of the proposed pedagogical policy. Therefore, in the future, we would like to incorporate the pedagogical policy in ITSs in other domains. Additionally, in the future, we plan to extend our study and perform further analyses among the students of different proficiency in the proposed condition and the other conditions in order to get a more comprehensive understanding of the proposed policy's effectiveness across a spectrum of proficiency levels.

9. ACKNOWLEDGMENTS

This research was supported by the NSF Grants: Integrated Data-driven Technologies for Individualized Instruction in STEM Learning Environments(1726550) and Generalizing Data-Driven Technologies to Improve Individualized STEM Instruction by Intelligent Tutors (2013502).

10. REFERENCES

[1] M. Abdelshiheed, J. W. Hostetter, T. Barnes, and M. Chi. Bridging declarative, procedural, and conditional metacognitive knowledge gap using deep reinforcement learning. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 45:333–340, 2023.

[2] M. Abdelshiheed, J. W. Hostetter, T. Barnes, and M. Chi. Leveraging deep reinforcement learning for metacognitive interventions across intelligent tutoring systems. In *International Conference on Artificial Intelligence in Education*, pages 291–303, 2023.

[3] N. Alam, B. Mostafavi, M. Chi, and T. Barnes. Exploring the effect of autoencoder based feature learning for a deep reinforcement learning policy for providing proactive help. In *International Conference on Artificial Intelligence in Education*, pages 278–283. Springer, 2023.

[4] M. S. Ausin, H. Azizsoltani, T. Barnes, and M. Chi. Leveraging deep reinforcement learning for pedagogical policy induction in an intelligent tutoring system. *International Educational Data Mining Society*, pages 168–177, 2019.

[5] H. Cen, K. R. Koedinger, and B. Junker. Is over practice necessary?-improving learning efficiency with the cognitive tutor through educational data mining. *Frontiers in artificial intelligence and applications*, 158:511–518, 2007.

[6] F. M. Fahid, J. P. Rowe, R. D. Spain, B. S. Goldberg, R. Pokorny, and J. Lester. Adaptively scaffolding cognitive engagement with batch constrained deep q-networks. In *International conference on artificial intelligence in education*, pages 113–124. Springer, 2021.

[7] H. v. Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2094–2100, 2016.

[8] J. W. Hostetter, C. Conati, X. Yang, M. Abdelshiheed, T. Barnes, and M. Chi. Xai to increase the effectiveness of an intelligent pedagogical agent. In *Proceedings of the 23rd ACM International Conference on Intelligent Virtual Agents*, pages 1–9, 2023.

[9] S. Ju, G. Zhou, M. Abdelshiheed, T. Barnes, and M. Chi. Evaluating critical reinforcement learning framework in the field. In *International conference on artificial intelligence in education*, pages 215–227. Springer, 2021.

[10] M. Maniktala, C. Cody, A. Isvik, N. Lytle, M. Chi, and T. Barnes. Extending the hint factory for the assistance dilemma: a novel, data-driven helpneed predictor for proactive problem-solving help. *arXiv preprint arXiv:2010.04124*, 2020.

[11] J. Matayoshi, E. Cosyn, and H. Uzun. Using recurrent neural networks to build a stopping algorithm for an adaptive assessment. In *Artificial Intelligence in*

Education: 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part II 20, pages 179–184. Springer, 2019.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[13] B. Mostafav and T. Barnes. Exploring the impact of data-driven tutoring methods on students' demonstrative knowledge in logic problem solving. *International Educational Data Mining Society*, 2016.

[14] M. Sanz Ausin, M. Maniktala, T. Barnes, and M. Chi. Exploring the impact of simple explanations and agency on batch deep reinforcement learning induced pedagogical policies. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part I 21*, pages 472–485. Springer, 2020.

[15] S. Shen, M. S. Ausin, B. Mostafavi, and M. Chi. Improving learning & reducing time: A constrained action-based reinforcement learning approach. In *Proceedings of the 26th conference on user modeling, adaptation and personalization*, pages 43–51, 2018.

[16] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. The hint factory: Automatic generation of contextualized help for existing computer aided instruction. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems Young Researchers Track*, pages 71–78, 2008.

[17] J. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education*, 22(1-2):3–17, 2013.

[18] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[19] G. Zhou, H. Azizsoltani, M. S. Ausin, T. Barnes, and M. Chi. 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 20*, pages 544–556. Springer, 2019.