Leveraging Deep Reinforcement Learning for Metacognitive Interventions across Intelligent Tutoring Systems

Mark Abdelshiheed, John Wesley Hostetter,
Tiffany Barnes, and Min Chi
{mnabdels, jwhostet, tmbarnes, mchi}@ncsu.edu

North Carolina State University, Raleigh, NC 27695, USA

Abstract. This work compares two approaches to provide metacognitive interventions and their impact on preparing students for future learning across Intelligent Tutoring Systems (ITSs). In two consecutive semesters, we conducted two classroom experiments: Exp. 1 used a classic artificial intelligence approach to *classify* students into different metacognitive groups and provide *static* interventions based on their classified groups. In Exp. 2, we leveraged Deep Reinforcement Learning (*DRL*) to provide *adaptive* interventions that consider the dynamic changes in the student's metacognitive levels. In both experiments, students received these interventions that taught *how* and *when* to use a backward-chaining (BC) strategy on a logic tutor that supports a default forward-chaining strategy. Six weeks later, we trained students on a probability tutor that only supports BC without interventions. Our results show that adaptive DRL-based interventions closed the metacognitive skills gap between students. In contrast, static classifier-based interventions only benefited a subset of students who knew *how* to use BC in advance. Additionally, our DRL agent prepared the experimental students for future learning by significantly surpassing their control peers on both ITSs.

Keywords: Reinforcement Learning · Artificial Intelligence · Intelligent Tutoring Systems · Metacognitive Interventions · Metacognitive Skills

1 Introduction

A challenging desired aspect of learning is being continuously prepared for future learning [9]. Our incremental knowledge is the evidence that preparation for future learning exists yet is hard to predict and measure [10]. Considerable research has found that one factor that facilitates preparing students for future learning is their metacognitive skills [7, 26]. We focus on two types of metacognitive skills related to problem-solving strategies: *strategy-awareness* [5, 20, 22] and *time-awareness* [6, 8], that are respectively, *how* and *when* to use each strategy.

Substantial work has demonstrated that metacognitive interventions of domain knowledge or problem-solving strategies accelerate preparation for future learning [7, 19, 22] and promote strategy- and time-awareness [12, 23, 26]. Such

interventions included hints, feedback, prompted nudges, worked examples, and direct strategy presentation. However, these interventions were *static or hard-coded* into the learning environment despite the fact that students often acquire and master metacognitive skills as they learn [17]. Reinforcement Learning (RL) [24] is one of the most effective approaches for providing adaptive support and scaffolding across Intelligent Tutoring Systems (ITSs) [14, 16, 27]. The deep learning extension of RL, known as Deep RL (DRL), has been commonly utilized in pedagogical policy induction across ITSs [15, 21] due to its higher support of model sophistication. As far as we know, no prior work has leveraged DRL to provide *adaptive* metacognitive interventions and investigated their impact on preparation for future learning across ITSs.

In this work, we conducted two consecutive experiments to compare two approaches for providing metacognitive interventions and their impact on preparing students for future learning on ITSs. In Exp. 1, we utilized a Random Forest Classifier (RFC) to classify students into different metacognitive groups and then provide static interventions based on their classification, while in Exp. 2, we leveraged a DRL-based approach for adaptive interventions that consider the dynamic changes in the student's metacognitive levels. Based on strategy- and time-awareness, our prior work classified students into those who are *both* strategy- and time-aware (*StrTime*), those who are *only* strategy-aware (*StrOnly*), and the rest who follow the default strategy (*Default*) [5, 6]. We found that only *StrTime* students were prepared for future learning, as they learned significantly better than their peers across different deductive domains. Motivated by such findings, we designed metacognitive interventions to teach students *how* and *when* to use a backward-chaining (BC) strategy on a logic tutor that supports a default forward-chaining strategy. After six weeks, students were trained on a probability tutor that only supports BC without receiving interventions. Our results showed that *Default* and *StrOnly* students benefited equally from our DRL policy and surprisingly outperformed *StrTime* students. However, the RFC-based approach only helped *StrOnly* students to catch up with *StrTime*.

2 Background and Related Work

2.1 Metacognitive Interventions for Strategy Instruction

Metacognition indicates one's awareness of their cognition and the ability to control and regulate it [13]. Strategy- and time-awareness are two metacognitive skills that address *how* and *when* to use a problem-solving strategy, respectively [6, 8]. Much prior work has emphasized the role of strategy awareness in preparation for future learning [5, 22] and the impact of time awareness on academic performance and planning skills [8, 11].

Considerable research has shown that metacognitive interventions promote strategy- and time-awareness [12, 23, 26]. We focus on two metacognitive interventions: directly presenting the strategy [12, 23] and prompting nudges to use it [7, 19, 26]. Spörer et al. [23] found that students who were explicitly instructed on comprehensive reading strategies surpassed their peers, who were taught by

the instructors' text interactions, on a transfer task and follow-up test. They also understood how, when, and why to use each reading strategy.

Zepeda et al. [26] demonstrated that metacognitive interventions impact learning outcomes, strategy mastery, and preparation for future learning. The experimental condition who received tutoring nudges and worked examples performed significantly better on a physics test than their control peers. They also made better metacognitive judgments and demonstrated mastery of knowing how and when to use physics strategies. As an example of preparation for future learning, the experimental students performed better on a novel self-guided 'control of variables' learning task than their control peers.

Despite much prior work on metacognitive interventions, the interventions were either not adaptive, not applied to ITSs, or had no preparation for future learning assessment. In our work, we conducted two experiments to provide metacognitive interventions and investigated their impact on preparing students for future learning across ITSs. Specifically, we first attempted static metacognitive interventions using a classifier-based approach, then compared it against a DRL-based approach for adaptive metacognitive interventions.

2.2 Reinforcement Learning in Intelligent Tutoring Systems

Reinforcement Learning (RL) is a popular machine learning branch ideal in environments where actions result in numeric rewards without knowing a ground truth [24]. Due to its aim of maximizing the cumulative reward, RL has been widely used in educational domains due to the flexible implementation of reward functions [14, 15, 21]. Deep RL (DRL) is a field that combines RL algorithms with neural networks; for instance, Deep Q-Learning is the neural network extension of Q-Learning, where a neural network is used to approximate the Q-function [18]. Substantial work has used RL and DRL in inducing pedagogical policies across ITSs [15, 21, 27]. Zhou et al. [27] utilized hierarchical RL to improve the learning gain on an ITS. They showed that their policy significantly outperformed an expert and a random condition.

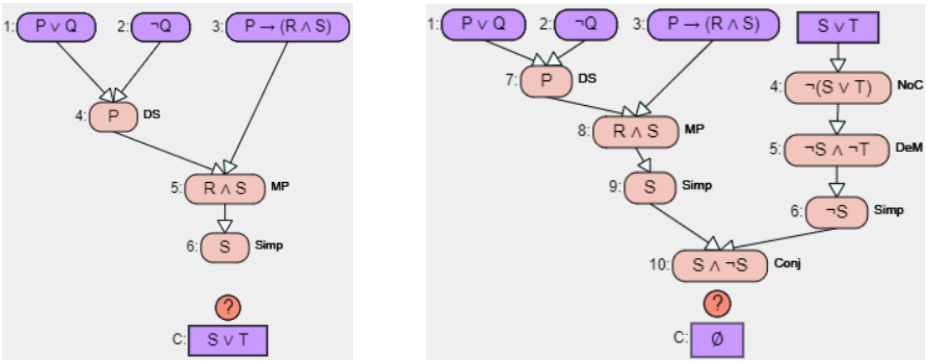
Ju et al. [15] presented a DRL framework that identifies the critical decisions to induce a critical policy on an ITS. They evaluated their critical-DRL framework based on two success criteria: *necessity* and *sufficiency*. The former required offering help in **all** critical states, and the latter required offering help **only** in critical states. Their results showed that the framework fulfilled both criteria. Sanz-Ausin et al. [21] conducted two consecutive classroom studies where DRL was applied to decide whether the student or tutor should solve the following problem. They found that the DRL policy with simple explanations significantly improved students' learning performance more than an expert policy.

Despite the wide use of RL and DRL on ITSs, the attempts to combine either with metacognitive learning have been minimal [16]. Krueger et al. [16] used RL to teach the metacognitive skill of knowing how much to plan ahead (Deciding How to Decide). Their metacognitive reinforcement learning framework builds on the semi-gradient SARSA algorithm [24] that was developed to approximate Markov decision processes. They defined a meta Q-function, known as Q_{meta},

that takes the meta state of the environment and the planning horizon action. They evaluated their framework on two planning tasks, where the authors defined constrained reward functions, and the rewards could be predicted many steps ahead to facilitate forming a plan.

In our work, we induced and deployed a DRL policy of metacognitive interventions on a logic tutor and investigated its impact on preparing students for future learning on a subsequent probability tutor. Additionally, we did not override any DRL mathematical definition, such as the Q-function. Instead, our DRL algorithm's metacognitive aspect resides in our interventions' nature.

3 Logic and Probability Tutors



(a) Forward Chaining (b) Backward Chaining

Fig. 1. Logic Tutor Problem-Solving Strategies

Logic Tutor: It teaches propositional logic proofs by applying valid inference rules such as Modus Ponens through the standard sequence of pre-test, training and post-test. The three phases share the same interface, but training is the *only* one where students can seek and get help. The pre-test has two problems, while the post-test is harder and has six problems; the first two are isomorphic to the pre-test problems. Training consists of five ordered levels with an *incremental degree of difficulty*, and each level consists of four problems. Every problem has a score in the $[0, 100]$ range based on the accuracy, time and solution length.

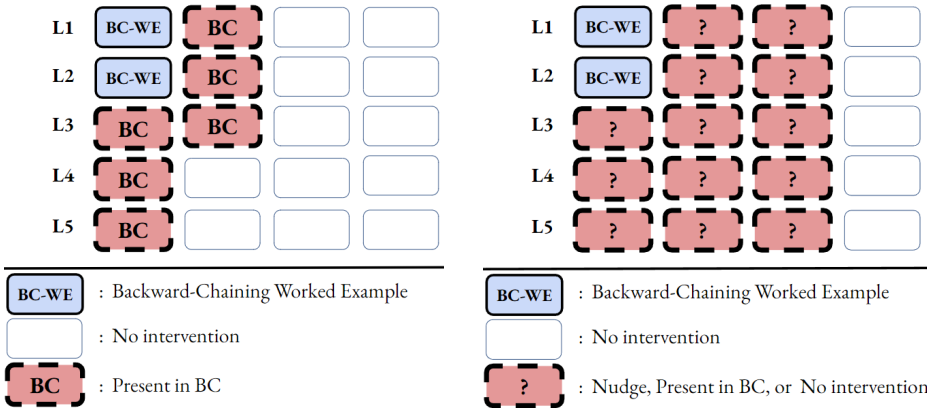
The *pre-* and *post-test* scores are calculated by averaging their pre- and post-test problem scores. A student can solve any problem throughout the tutor by either a *forward-chaining* or a *backward-chaining (BC)* strategy. Figure 1a shows that for *forward chaining*, one must derive the conclusion at the bottom from givens at the top, while Figure 1b shows that for *BC*, students need to derive a contradiction from givens and the *negation* of the conclusion. Problems are presented by *default* in forward chaining, but students can switch to BC by clicking a button in the tutor interface.

Probability Tutor: It teaches how to solve probability problems using ten principles, such as the Complement Theorem. The tutor consists of a textbook, pre-test, training, and post-test. Like the logic tutor, training is the only section for students to receive and ask for hints, and the post-test is harder than the pre-test. The textbook introduces the domain principles, while training consists of 12 problems, each of which can *only* be solved by *BC* as it requires deriving an answer by *writing and solving equations* until the target is ultimately reduced to the givens.

In pre- and post-test, students solve 14 and 20 open-ended problems, where each pre-test problem has an isomorphic post-test problem. The answers are graded in a double-blind manner by experienced graders using a partial-credit rubric, where grades are based *only* on accuracy in the $[0, 100]$ range. The *pre-* and *post-test* scores are the average grades in their respective sections.

4 Methods

As students can choose to switch problem-solving strategies *only* on the logic tutor, our interventions are provided in the logic training section [1–3, 5]. It was shown that *StrTime* students frequently follow the desired behavior of switching ***early*** to *BC* on the logic tutor, their *StrOnly* peers switch ***late***, and the *Default* students make ***no*** switches and stick to the default strategy [4, 6]. Additionally, we found that providing metacognitive interventions that recommend switching to BC —referred to as Nudges— or present problems directly in BC —known as Direct Presentation— for *Default* and *StrOnly* students cause them to catch up with their *StrTime* peers [2, 3, 5]. Therefore, we conducted two experiments to investigate different ways to present such metacognitive interventions for *Default* and *StrOnly* students.



(a) Exp. 1: RFC−Static (b) Exp. 2: DRL−Adaptive

Fig. 2. Training on the Modified Logic Tutor

4.1 Experiment 1: RFC−Static

We utilized a RFC that could early predict a student's metacognitive group —*Default*, *StrOnly* or *StrTime*— based on the incoming competence of the logic tutor. The RFC was previously shown to be 96% accurate [5].

After early prediction, we trained *StrTime* students on the original logic tutor with all problems presented by default in forward chaining, while *Default* and *StrOnly* students were assigned the modified tutor, as shown in Figure 2a. Specifically, two worked examples (WE) on BC were provided, where the tutor showed a step-by-step solution, and six problems were presented in BC by default (Direct Presentation). We expected the WEs and six problems to teach students *how* and *when* to use BC. Note that we selected the colored problems in Figure 2a based on our data's historical switches to BC [6]. This experiment provided a static metacognitive intervention —Direct Presentation— which was preferred to Nudges due to its prior success with *Default* and *StrOnly* students [3].

4.2 Experiment 2: DRL−Adaptive

We leveraged DRL to provide adaptive metacognitive interventions —*Nudge*, *Direct Presentation*, or *No Intervention*— regardless of the RFC's metacognitive group prediction. We trained *Experimental* students on the modified tutor shown in Figure 2b. Figure 3 shows an example of a nudge, which is prompted after a number of seconds sampled from a probability distribution of prior students' switch behavior [6]. Since our interventions included the no-intervention option, we intervened in as many problems as possible. The WEs from Experiment 1 were kept, as they are vital for teaching students ***how*** to use BC. We did not intervene in the last training problem at each level, as it is used to evaluate the student's improvement on that level.

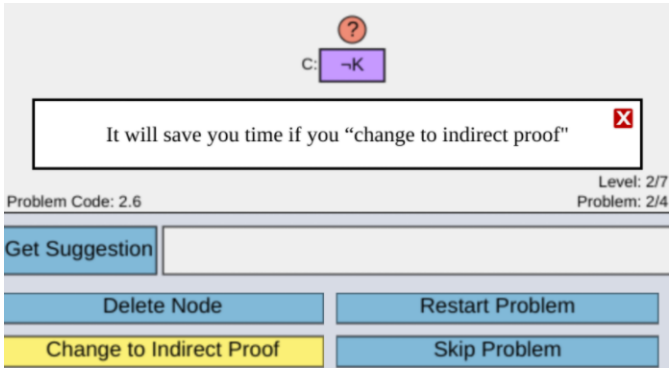


Fig. 3. Strategy Switch Nudge

Training Corpus and Policy Induction: To train our DRL agent, we utilized data collected from four previous studies consisting of 867 students [2, 3, 5, 6] and

performed a $80-20$ train-test split. The dataset consisted of a record per each student on a logic training problem represented as (**state**, **action**, **reward**). The *state* is the feature vector comprising 152 features that capture temporal, accuracy-based and hint-based behaviors. The *action* is either Nudge, Direct Presentation, or No Intervention. The *reward* is the immediate problem score on the logic tutor, as stated earlier in Section 3.

Our goal is to show that DRL works with our metacognitive interventions **rather** than *which* DRL algorithm is better with our interventions. We preferred DRL to RL due to its prevailing success in educational domains [15, 21]. To select our DRL algorithm, we had to avoid a relatively simple algorithm such as Deep Q-Network (DQN), which overestimates action values [18] and may result in underfitting. Furthermore, we needed to avoid sophisticated DRL algorithms, such as autoencoders and actor-critic approaches, so that DRL does not overshadow the impact of our metacognitive interventions. In other words, a sophisticated DRL algorithm yielding an optimal policy would be acknowledged likely for its sophistication rather than for the metacognitive interventions it provided. Thus, we leveraged Double-DQN (DDQN), which solves the overestimation issue in DQN by **decoupling** the action *selection* from the action *evaluation* in two different neural networks [25]. The resulting modified Bellman equation becomes:

$$Q(s, a; \boldsymbol{\theta}) = r + \gamma\, Q(s', argmax_{a'} Q(s', a', \boldsymbol{\theta}); \boldsymbol{\theta}^{-}) \tag{1}$$

where r is the reward; γ is the discount factor; s and s' refer to the current and next states; a and a' denote the current and next actions. Specifically, DDQN uses the **main** ($\boldsymbol{\theta}$) neural network to *select* the action with the highest Q-value for the next state and then *evaluates* the Q-value of that action using the **target** ($\boldsymbol{\theta}^{-}$) neural network. After hyperparameter tuning, we picked the model with the lowest mean squared error loss. The deployed policy had two hidden layers with 16 neurons each, $1e$-3 learning late, $9e$-1 discount factor, 32 batch size, a synchronization frequency of 4 steps between main and target neural networks, and was trained until convergence ($\approx$ 2000 epochs).

5 Experiments Setup

The two experiments took place in an undergraduate Computer Science class at North Carolina State University in the Spring and Fall of 2022, respectively. The participants were assigned each tutor as a class assignment and told that completion is required for full credit. In both experiments, students were assigned to the logic tutor following the standard procedure of pre-test, training (Fig. 2a for Exp. 1 and Fig. 2b for Exp. 2), and post-test. We trained students on the probability tutor six weeks later, where no interventions were provided. Each probability training problem was randomly assigned for the student to solve on their own, for the tutor to present it as a worked example (WE), or both to collaborate in the form of collaborative problem-solving. Note that on both tutors, the problem order is the same for all students.

Exp. 1 (RFC) Participants: A total of 121 students finished both tutors and were classified by the RFC into 47 *Default*, 48 *StrOnly* and 26 *StrTime*. *Default* and *StrOnly* were randomly assigned to *Experimental (RFC)* and *Control (Ctrl)* conditions. *Experimental* received interventions on logic training (Fig. 2a), while *Control* and *StrTime* received no interventions. We had 24 $Default_{RFC}$, 25 $StrOnly_{RFC}$, 23 $Default_{Ctrl}$, 23 $StrOnly_{Ctrl}$ and 26 *StrTime* students.

Exp. 2 (DRL) Participants: A total of 112 students finished both tutors and were randomly assigned to *Experimental (DRL)* and *Control (Ctrl)* conditions. *Experimental* received interventions on logic training (Fig. 2b), while *Control* received no interventions. To investigate whether DRL would help students with different incoming metacognitive skills, we used the *RFC to ensure even distribution across conditions and metacognitive groups for comparison purposes*. We found that our DRL policy provided no interventions for $StrTime_{DRL}$ students 94% of the time. As a result, we combined $StrTime_{DRL}$ and $StrTime_{Ctrl}$ into *StrTime*. We had 22 $Default_{DRL}$, 24 $StrOnly_{DRL}$, 22 $Default_{Ctrl}$, 22 $StrOnly_{Ctrl}$ and 22 *StrTime* students.

6 Results

6.1 Experiment 1: RFC−Static

Table 1. Experiment 1 (RFC−Static) Results

	Experimental (RFC)		*Control*		
	$Default_{RFC}$ ($N = 24$)	$StrOnly_{RFC}$ ($N = 25$)	$Default_{Ctrl}$ ($N = 23$)	$StrOnly_{Ctrl}$ ($N = 23$)	*StrTime* ($N = 26$)
Logic Tutor					
Pre	57.8 (20)	56.5 (17)	55.9 (20)	55.7 (21)	57.4 (20)
Iso. Post	76.5 (14)	**83.9 (9)***	72.3 (16)	73 (15)	81.8 (11)*
Iso. NLG	0.18 (.15)	**0.32 (.11)***	0.14 (.29)	0.16 (.32)	0.31 (.17)*
Post	73.6 (13)	**80.5 (9)***	69.1 (13)	71.6 (11)	79.8 (10)*
NLG	0.16 (.12)	**0.31 (.12)***	0.12 (.31)	0.13 (.3)	0.28 (.16)*
Probability Tutor					
Pre	75.5 (16)	74.2 (15)	73.8 (14)	74.8 (16)	76.1 (16)
Iso. Post	75.2 (17)	**93.6 (5)***	70.9 (14)	72.6 (15)	91.2 (7)*
Iso. NLG	0.04 (.34)	**0.35 (.16)***	-0.03 (.29)	-0.02 (.31)	0.28 (.18)*
Post	74.6 (19)	**92.7 (7)***	69.5 (16)	70.9 (17)	90.3 (8)*
NLG	0.02 (.37)	**0.32 (.18)***	-0.07 (.32)	-0.04 (.36)	0.25 (.17)*

In a row, bold is for the highest value, and asterisk means significance over no asterisks.

Table 1 compares the groups' performance in Experiment 1. We show the mean and standard deviation of pre- and post-test scores, isomorphic scores, and the learning outcome in terms of the normalized learning gain (*NLG*) [6, 14] defined

as ($NLG = \frac{Post-Pre}{\sqrt{100-Pre}}$), where 100 is the maximum test score. We refer to pre-test, post-test and NLG scores as *Pre*, *Post* and *NLG*, respectively. The RFC was 97% accurate in classifying students who received no interventions —$Default_{Ctrl}$, $StrOnly_{Ctrl}$ and *StrTime*. On each tutor, a one-way ANOVA using group as factor found no significant difference on *Pre*: $F(4, 116) = 0.21$, $p = .93$ for logic and $F(4, 116) = 0.38$, $p = .82$ for probability.

To measure the students' improvement on isomorphic problems, repeated measures ANOVA tests were conducted (one for each group on each tutor) using {*Pre*, *Iso. Post*} as factor. We found that $StrOnly_{RFC}$ and *StrTime* learned significantly with $p < .0001$ on both tutors, while $Default_{RFC}$ and the control groups did not perform significantly higher on *Iso. Post* than *Pre* on either tutor. These findings verify the RFC's accuracy, as *StrTime* learned significantly on both tutors, while *Control* did not, despite each receiving no interventions.

On both tutors, a one-way ANCOVA using *Pre* as covariate and group as factor found a significant effect on *Post*: $F(4, 115) = 7.4$, $p < .0001$, $\eta^2 = 0.56$. for logic and $F(4, 115) = 8.2$, $p < .0001$, $\eta^2 = 0.63$. for probability. Follow-up pairwise comparisons with Bonferroni adjustment ($\alpha = .05/10$) showed that $StrOnly_{RFC}$, *StrTime* > $Default_{RFC}$, $Default_{Ctrl}$, $StrOnly_{Ctrl}$ on both tutors. Similar patterns were found on *NLG* using ANOVA. In essence, while no significant difference was found between $StrOnly_{RFC}$ and *StrTime*, each significantly outperformed the remaining three groups.

6.2 Experiment 2: DRL−Adaptive

Table 2. Experiment 2 (DRL−Adaptive) Results

	Experimental (DRL)		*Control*		
	$Default_{DRL}$ ($N = 22$)	$StrOnly_{DRL}$ ($N = 24$)	$Default_{Ctrl}$ ($N = 22$)	$StrOnly_{Ctrl}$ ($N = 22$)	*StrTime* ($N = 22$)
Logic Tutor					
Pre	55.6 (21)	56.1 (21)	55.2 (19)	56.4 (23)	58.2 (19)
Iso. Post	91.9 (5)*	**92.1 (4)***	72.7 (18)	74.1 (17)	83.4 (12)*
Iso. NLG	0.46 (.12)*	**0.48 (.09)***	0.18 (.3)	0.14 (.27)	0.35 (.11)*
Post	**87.7 (5)***	87.6 (5)*	70 (15)	69.7 (16)	80.2 (11)*
NLG	**0.45 (.12)***	0.44 (.14)*	0.16 (.33)	0.1 (.31)	0.31 (.15)*
Probability Tutor					
Pre	76.9 (15)	74.6 (16)	75.2 (15)	76.7 (13)	78.6 (14)
Iso. Post	94.5 (5)*	**96.1 (3)***	73.9 (10)	71.4 (14)	89.1 (7)*
Iso. NLG	0.36 (.11)*	**0.43 (.13)***	-0.02 (.18)	-0.06 (.21)	0.24 (.15)*
Post	94.1 (6)*	**95.6 (4)***	71.8 (11)	68.6 (17)	87.7 (8)*
NLG	0.34 (.13)*	**0.39 (.16)***	-0.07 (.24)	-0.1 (.25)	0.22 (.19)*

In a row, bold is for the highest value, and asterisk means significance over no asterisks.

Table 2 compares the groups' performance in Experiment 2. The RFC was 98% accurate in classifying students who received no interventions —$Default_{Ctrl}$,

$StrOnly_{Ctrl}$ and $StrTime$. A one-way ANOVA using group as factor found no significant difference on *Pre*: $F(4, 107) = 0.18$, $p = .95$ for logic and $F(4, 107) = 0.45$, $p = .77$ for probability. We conducted repeated measures ANOVA for each group on each tutor using {*Pre*, *Iso. Post*} as factor. We found that $Default_{DRL}$, $StrOnly_{DRL}$ and $StrTime$ learned significantly with $p < .0001$ on both tutors, unlike $Default_{Ctrl}$ and $StrOnly_{Ctrl}$.

A one-way ANCOVA using *Pre* as covariate and group as factor found a significant effect on *Post* on both tutors: $F(4, 106) = 9.3$, $p < .0001$, $\eta^2 = 0.61$ for logic and $F(4, 106) = 10.6$, $p < .0001$, $\eta^2 = 0.74$ for probability. Subsequent Bonferroni-corrected analyses ($\alpha = .05/10$) revealed that $Default_{DRL}$, $StrOnly_{DRL} > StrTime > Default_{Ctrl}$, $StrOnly_{Ctrl}$ on both tutors. For instance, $Default_{DRL}$ had significantly higher *Post* than $StrTime$ ($t(42) = 3.9$, $p < .001$, $d = 0.88$ for logic and $t(42) = 3.7$, $p < .001$, $d = 0.91$ for probability) and $Default_{Ctrl}$ ($t(42) = 6.6$, $p < .0001$, $d = 1.6$ for logic and $t(42) = 7.1$, $p < .0001$, $d = 2.5$ for probability). Similar patterns were observed using ANOVA on *NLG*. In brief, the two DRL groups benefited equally from our policy, significantly outperformed their control peers, and surprisingly surpassed $StrTime$ students.

6.3 Post-hoc Analysis

To compare the results between the two experiments, we performed a Shapiro-Wilk normality test for each metric for each group in Tables 1 and 2. The results showed no evidence of non-normality ($p > .05$). Therefore, we conducted independent samples t-test for every two identical groups between Tables 1 and 2. Specifically, we found no significant difference between the $StrTime$ groups across the two tables[1]. Similarly, no such difference was observed between the $Default_{Ctrl}$ groups or between their $StrOnly_{Ctrl}$ peers.

The main objective was to compare our *static RFC-based* and *adaptive DRL-based* interventions. First, we compared the interventions' distribution within each experiment. The RFC experimental students received static interventions; hence, the distribution is identical between $Default_{RFC}$ and $StrOnly_{RFC}$. For DRL students, $Default_{DRL}$ received 94(33%) Nudges, 65(23%) Direct Presentation and 127(44%) No Intervention, while $StrOnly_{DRL}$ received 82(26%) Nudges, 74(24%) Direct Presentation and 156(50%) No Intervention. A chi-square test showed no significant difference in the distribution of interventions between the experimental DRL groups: $\chi^2(2, N = 598) = 3.2$, $p = .2$.

Second, we compared the learning performance on both tutors. On the logic tutor, a two-way ANCOVA using *Pre* as covariate, and condition {*RFC*, *DRL*} and metacognitive group {*Default*, *StrOnly*} as factors, found a significant interaction effect on *Post*: $F(1, 90) = 37.9$, $p < .0001$, $\eta^2 = 0.78$. There was also a main effect of condition: $F(1, 90) = 28.4$, $p < .0001$, $\eta^2 = 0.59$ in that the *DRL* groups significantly outperformed their *RFC* peers. Follow-up Bonferroni-corrected analyses ($\alpha = .05/6$) confirmed that $Default_{DRL}$,

[1]This holds for all metrics, such as *Pre*, *Post* and *NLG*.

$StrOnly_{DRL}$ >$Default_{RFC}$, $StrOnly_{RFC}$. For example, $StrOnly_{DRL}$ had significantly higher *Post* than $StrOnly_{RFC}$: $t(47) = 3.6$, $p < .001$, $d = 0.98$. Similar results were found using ANOVA on *NLG*.

On probability, a two-way ANCOVA using *Pre* as covariate and the same two factors found a significant interaction effect on *Post*: $F(1, 90) = 29.1$, $p < .0001$, $\eta^2 = 0.46$. Subsequent pairwise analyses with Bonferroni adjustment ($\alpha = .05/6$) revealed that $Default_{DRL}$, $StrOnly_{DRL}$, $StrOnly_{RFC} > Default_{RFC}$. For instance, $StrOnly_{RFC}$ had significantly higher *Post* than $Default_{RFC}$: $t(47) = 6.4$, $p < .0001$, $d = 1.3$. Similar results were observed on *NLG* using ANOVA.

7 Discussions & Conclusions

We showed that leveraging DRL to provide adaptive metacognitive interventions closed the gap between metacognitive groups and caused them to surpass their control peers. Surprisingly, our DRL policy allowed the experimental students to outperform *StrTime* students significantly. On the other hand, using a RFC-based approach to provide static interventions only benefited a subset of students —*StrOnly*— who caught up with their *StrTime* peers.

It is also evident that DRL prepared the experimental students for future learning [9] by outperforming their control peers on both tutors. In other words, the experimental students outperformed their peers on probability based on interventions they received on logic. This finding suggests that they acquired backward-chaining skills in logic and transferred them to probability, where they received no interventions.

Limitations and Future Work There are at least two caveats in our work. First, splitting students into experimental and control conditions resulted in relatively small sample sizes. Second, the probability tutor supported only one strategy, which restricted our intervention ability to the logic tutor. The future work involves comparing the RFC-based and DRL-based approaches within one study. Additionally, we aim to make the probability tutor support forward chaining, like the logic tutor.

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

References

1. Abdelshiheed, M., Hostetter, J.W., Barnes, T., Chi, M.: Bridging declarative, procedural, and conditional metacognitive knowledge gap using deep reinforcement learning. In: CogSci (2023)
2. Abdelshiheed, M., Hostetter, J.W., Shabrina, P., Barnes, T., Chi, M.: The power of nudging: Exploring three interventions for metacognitive skills instruction across intelligent tutoring systems. In: CogSci. pp. 541–548 (2022)
3. Abdelshiheed, M., Hostetter, J.W., Yang, X., Barnes, T., Chi, M.: Mixing backward- with forward-chaining for metacognitive skill acquisition and transfer. In: AIED. pp. 546–552. Springer International Publishing (2022)

4. Abdelshiheed, M., Maniktala, M., Barnes, T., Chi, M.: Assessing competency using metacognition and motivation: The role of time-awareness in preparation for future learning. In: Design Recommendations for Intelligent Tutoring Systems, vol. 9, pp. 121–131. US Army CCDC Soldier Center (2022)
5. Abdelshiheed, M., Maniktala, M., Ju, S., Jain, A., Barnes, T., Chi, M.: Preparing unprepared students for future learning. In: CogSci. pp. 2547–2553 (2021)
6. Abdelshiheed, M., Zhou, G., Maniktala, M., Barnes, T., Chi, M.: Metacognition and motivation: The role of time-awareness in preparation for future learning. In: CogSci. pp. 945–951 (2020)
7. Belenky, D.M., et al.: Examining the role of manipulatives and metacognition on engagement, learning, and transfer. The Journal of Problem Solving **2**(2), 6 (2009)
8. de Boer, H., et al.: Long-term effects of metacognitive strategy instruction on student academic performance: A meta-analysis. Educ. Psychol. Rev. **24** (2018)
9. Bransford, J.D., Schwartz, D.L.: Rethinking transfer: A simple proposal with multiple implications. Review of research in education **24**(1), 61–100 (1999)
10. Detterman, D.K., Sternberg, R.J.: Transfer on trial: Intelligence, cognition, and instruction. Ablex Publishing (1993)
11. Fazio, L.K., et al.: Strategy use and strategy choice in fraction magnitude comparison. J Exp Psychol Learn Mem Cogn **42**(1), 1 (2016)
12. Fellman, D., et al.: The role of strategy use in working memory training outcomes. Journal of Memory and Language **110**, 104064 (2020)
13. Flavell, J.H.: Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. American psychologist **34**(10), 906 (1979)
14. Hostetter, J.W., Abdelshiheed, M., Barnes, T., Chi, M.: A self-organizing neuro-fuzzy q-network: Systematic design with offline hybrid learning. In: AAMAS (2023)
15. Ju, S., Zhou, G., Abdelshiheed, M., Barnes, T., Chi, M.: Evaluating critical reinforcement learning framework in the field. In: AIED. pp. 215–227. Springer (2021)
16. Krueger, P.M., Lieder, F., Griffiths, T.: Enhancing metacognitive reinforcement learning using reward structures and feedback. In: CogSci (2017)
17. Kuhn, D.: Metacognitive development. Curr Dir Psychol Sci **9**(5), 178–181 (2000)
18. Mnih, V., et al.: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015)
19. Richey, J.E., Zepeda, C.D., Nokes-Malach, T.J.: Transfer effects of prompted and self-reported analogical comparison and self-explanation. In: CogSci. vol. 37 (2015)
20. Roberts, M.J., Erdos, G.: Strategy selection and metacognition. Educational Psychology **13**, 259–266 (1993)
21. 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: AIED. pp. 472–485. Springer (2020)
22. Shabrina, P., Mostafavi, B., Abdelshiheed, M., Chi, M., Barnes, T.: Investigating the impact of backward strategy learning in a logic tutor: Aiding subgoal learning towards improved problem solving. IJAIED (2023)
23. Spörer, N., et al.: Improving students' reading comprehension skills: Effects of strategy instruction and reciprocal teaching. Learn. Instr. **19**(3), 272–286 (2009)
24. Sutton, R.S., et al.: Reinforcement learning: An introduction. MIT press (2018)
25. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: AAAI. vol. 30 (2016)
26. Zepeda, C.D., et al.: Direct instruction of metacognition benefits adolescent science learning, transfer, and motivation. J. Educ. Psychol. **107**(4), 954 (2015)
27. Zhou, G., Azizsoltani, H., Ausin, M.S., Barnes, T., Chi, M.: Hierarchical reinforcement learning for pedagogical policy induction. In: AIED. pp. 544–556 (2019)