Hierarchical Reinforcement Learning for Pedagogical Policy Induction

Guojing Zhou, Hamoon Azizsoltani, Markel Sanz Ausin, Tiffany Barnes, and Min Chi

AIED 2019

How can RL be used to help intelligent tutors make decisions about what actions to take?

The promise of RL for intelligent tutoring systems

Typically for an ITS to do its job it needs information about students, tasks, and some way to decide when and which tasks or resources to provide to the students (a policy)

- E.g., basic BKT has information about tasks and infers student knowledge, all of which can be used by another algorithm to make decisions

The promise of RL for intelligent tutoring systems

Typically for an ITS to do its job it needs information about students, tasks, and some way to decide when and which tasks or resources to provide to the students (a policy)

Rather than using an *a priori* policy, RL can be used to **learn** an optimal policy

The promise of RL for intelligent tutoring systems

Typically for an ITS to do its job it needs information about students, tasks, and some way to decide when and which tasks or resources to provide to the students (a policy)

Rather than using an *a priori* policy, RL can be used to **learn** an optimal policy

Furthermore, some RL approaches **can also infer information about students and tasks** (see Bassen et al., 2020)

The challenges of using RL for intelligent tutoring systems

The challenges of using RL for intelligent tutoring systems

A. “The lack of simulation-based-environments to train data-hungry RL methods,

The challenges of using RL for intelligent tutoring systems

A. “The lack of simulation-based-environments to train data-hungry RL methods,
B. the need for large (often unbounded) state space representations,

The challenges of using RL for intelligent tutoring systems

A. “The lack of simulation-based-environments to train data-hungry RL methods,
B. the need for large (often unbounded) state space representations,
C. the limited observability of the environment’s state (i.e., the student’s knowledge),

The challenges of using RL for intelligent tutoring systems

A. “The lack of simulation-based-environments to train data-hungry RL methods,
B. the need for large (often unbounded) state space representations,
C. the limited observability of the environment’s state (i.e., the student’s knowledge),
D. significantly delayed and noisy outcome measures, and

The challenges of using RL for intelligent tutoring systems

A. “The lack of simulation-based-environments to train data-hungry RL methods,
B. the need for large (often unbounded) state space representations,
C. the limited observability of the environment’s state (i.e., the student’s knowledge),
D. significantly delayed and noisy outcome measures, and
E. concerns about the robustness, interpretability, and fairness of RL methods when applied to the critical domain of ED” (Singla et al., 2021, p. 1)

Challenges addressed by Bassen et al., 2020

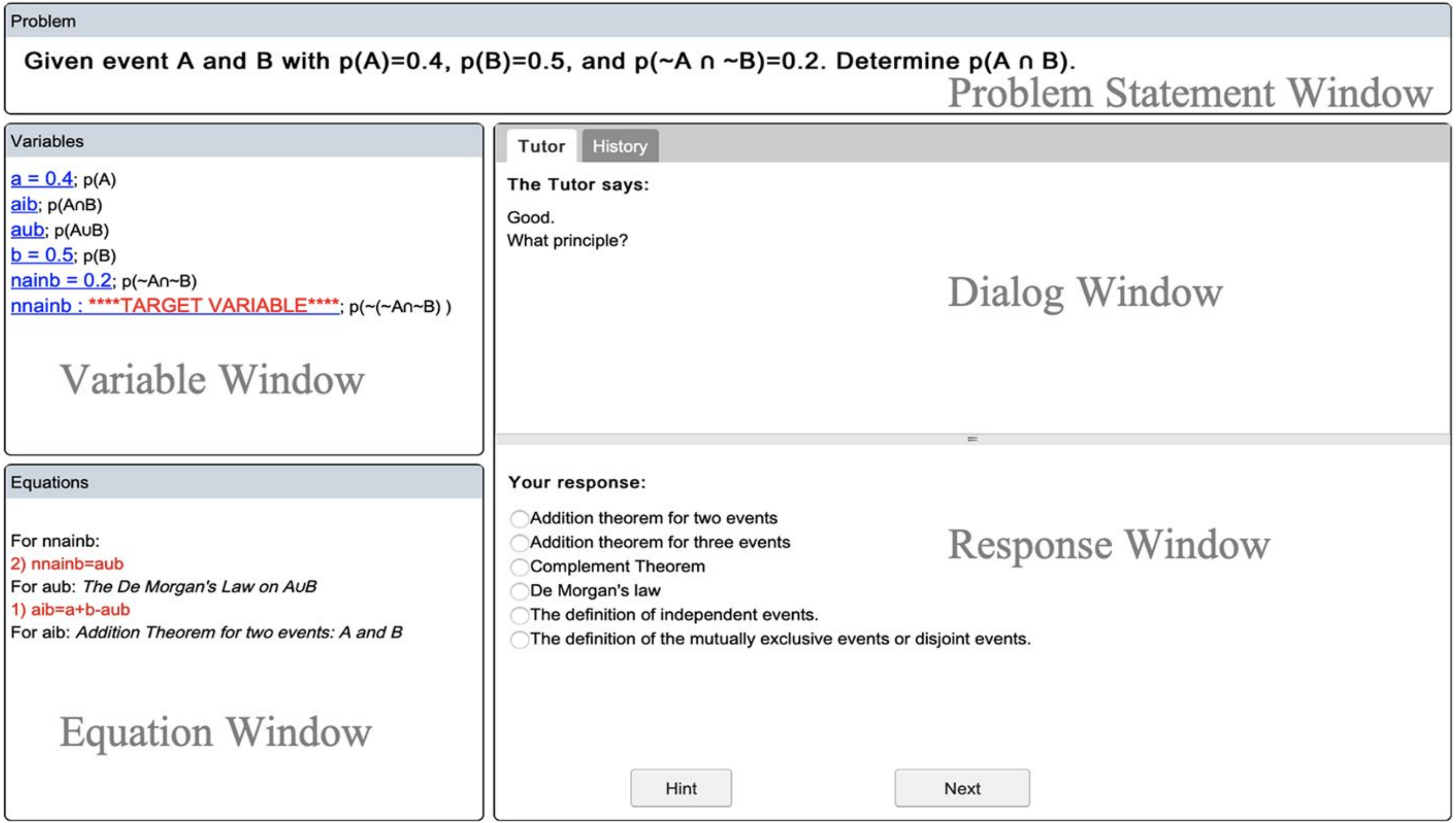
A. “The lack of simulation-based-environments to train data-hungry RL methods,
B. the need for large (often unbounded) state space representations,
C. the limited observability of the environment’s state (i.e., the student’s knowledge),
D. significantly delayed and noisy outcome measures, and
E. concerns about the robustness, interpretability, and fairness of RL methods when applied to the critical domain of ED” (Singla et al., 2021, p. 1)

Challenges addressed by Zhou et al., 2019

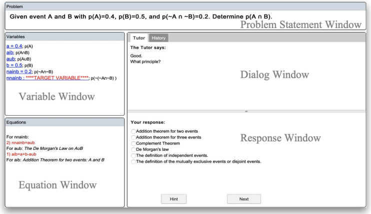
A. “The lack of simulation-based-environments to train data-hungry RL methods,
B. the need for large (often unbounded) state space representations,
C. the limited observability of the environment’s state (i.e., the student’s knowledge),
D. significantly delayed and noisy outcome measures, and
E. concerns about the robustness, interpretability, and fairness of RL methods when applied to the critical domain of ED” (Singla et al., 2021, p. 1)

Zhou et al.’s Problem

Pyrenees


Problem
Given event A and B with p(A)=0.4, p(B)=0.5, and p(~A ∩ ~B)=0.2. Determine p(A ∩ B).
Problem Statement Window
Variables
a = 0.4; p(A)
aib; p(A∩B)
aub; p(A∪B)
b = 0.5; p(B)
nainb = 0.2; p(~A∩~B)
nnainb : ****TARGET VARIABLE****; p(~(~A∩~B))
Variable Window
Tutor
History
The Tutor says:
Good.
What principle?
Dialog Window
Equations
For nnainb:
2) nnainb=aub
For aub: The De Morgan's Law on A∪B
1) aib=a+b-aub
For aib: Addition Theorem for two events: A and B
Equation Window
Your response:
Addition theorem for two events
Addition theorem for three events
Complement Theorem
De Morgan's law
The definition of independent events.
The definition of the mutually exclusive events or disjoint events.
Response Window
Hint
Next

Pyrenees



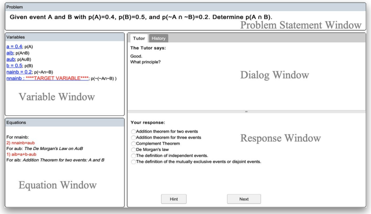
Problem types:

- Worked examples
- Faded worked examples
- Problem solving

For each problem type, the step types:

- Elicit
- Tell

Pyrenees



Problem types:

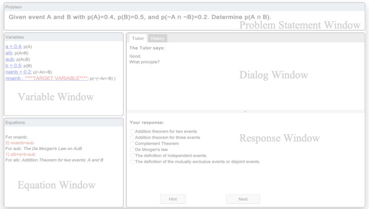
- Worked examples
- Faded worked examples
- Problem solving

For each problem type, the step types:

- Elicit
- Tell

	What it is	**Elicit / tell**
Worked example	The student observes how the tutor solves a problem	All-tell
Faded worked example	The student and the tutor co-construct the solution	Mix of tell and elicit
Problem solving	The student solves the problem	All-elicit

Pyrenees



Problem types:

- Worked examples
- Faded worked examples
- Probl

How to sequence these activities?

For each problem type, the step types:

- Elicit
- Tell

	What it is	Elicit / tell
Worked example	The student observes how the tutor solves a	All-tell
Faded worked example	The student and the tutor co-construct the solution	Mix of tell and elicit
Problem solving	The student solves the problem	All-elicit

Pyrenees

Problem types:

- Worked examp
- Faded worked
- Probl

For each problem t
step types:

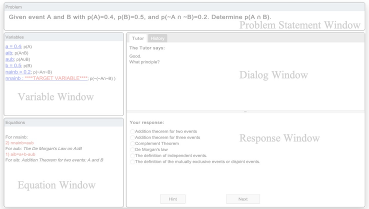
- Elicit
- Tell

Prior research has investigated the effectiveness of WE, PS, FWE, and their various combinations [14–17,21,23,26,31,33]. When focusing on PS and WE, Mclaren et al. found no significant difference in learning performance between studying WE-PS pairs and doing PS-only, but the former spent significantly less time than the PS-only [16]. In a subsequent study, Mclaren et al. compared three conditions: WE-only, PS-only and WE-PS pairs [15]. Similarly, no significant differences were found among them in terms of learning gains, but the WE condition spent significantly less time than the other two; and no significant time on task difference was found between PS-only and WE-PS pairs.

Several studies were conducted comparing different combinations of WE, PS, and FWE. Renkl et al. compared WE-FWE-PS with WE-PS pairs, and the former significantly outperformed the latter on student learning performance while no significant difference was found between them on time on task [21]. Similarly, Najar et al. compared adaptive WE/FWE/PS with WE-PS pairs [17].

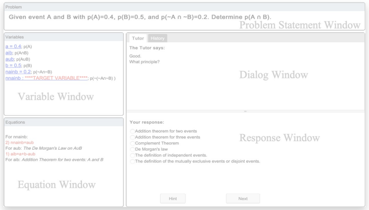
How to sequence these activities?

conditions: WE-FWE-PS, FWE, and PS-only [23]. Their results showed that FWE outperformed WE-FWE-PS, which in turn outperformed PS-only, and no significant time on task difference was found among the three conditions. Note that in their study, the order of WE, FWE, and PS were fixed in WE-FWE-PS; while in FWE, the tutor used an adaptive pedagogical policy, expert rules combined with data-driven student models. In short, previous studies have shown that alternating among WE, PS, and FWEs can be more effective than only alternating between WE and PS; however, it is not clear whether the former can be more effective than only using FWEs. On the other hand, prior research either used a fixed policy (WE-FWE-PS) or hand-coded expert rules combined with data-driven student models to make decisions. In this work, we applied an offline, off-policy HRL framework to derive a hierarchical pedagogical policy directly from empirical data. Its effectiveness is directly compared against another data-driven FWE policy induced by applying one of the state-of-the-art



	Elicit / tell
	All-tell
e	Mix of tell and elicit
e	All-elicit

Pyrenees



Problem types:

- Worked examples
- Faded worked examples
- Probl

For each p
step types:

- Elicit
- Tell

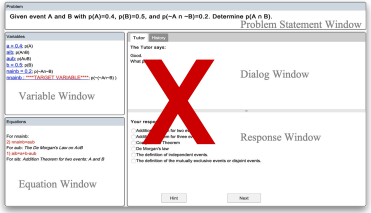
	What it is	Elicit / tell
Worked example	The student observes how the tutor solves a	All-tell
example	solution	of tell and it
Problem solving	The student solves the problem	All-elicit

How to sequence these activities?

💡 Let’s try DQN! 💡

But first... Inferring Intermediate Rewards with Gaussian Processes

DQN (and RL generally) works better when immediate rewards are available



💡 Use Gaussian process regression to infer intermediate rewards 💡

Azizsoltani, Kim, Ausin, Barnes, and Chi, 2019

Gaussian Processes (Generally)

“For a given set of training points, there are potentially infinitely many functions that fit the data. Gaussian processes offer an elegant solution to this problem by assigning a probability to each of these functions [1]. The mean of this probability distribution then represents the most probable characterization of the data.”

Görtler, J., Kehlbeck, R., & Deussen, O. (2019). A visual exploration of Gaussian processes. Distill, 4(4), e17.

Gaussian Processes (In the context of Zhou's problem)

Given delayed rewards and features for each state, use GP regression to learn a function that assigns intermediate rewards.

Gaussian Processes (In the context of Zhou's problem)

Given delayed rewards and features for each state, use GP regression to learn a function that assigns intermediate rewards.

- Normalized learning gain

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

Gaussian Processes (In the context of Zhou's problem)

Given delayed rewards and features for each state, use GP regression to learn a function that assigns intermediate rewards.

- Normalized learning gain

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

- 142 features in each state
  - Autonomy: amount of work done, ...
  - Temporal: avgStepTime, ...
  - Problem solving: problemDifficulty, nPrincipleInProblem, ...
  - Performance: pctCorrectPrin, ...
  - Hints: nHint, ...

Azizsoltani, Kim, Ausin, Barnes, and Chi, 2019

Two empirical studies were performed to evaluate the effectiveness of DQN-Del in Spring 2018 and DQN-Inf in Fall 2018, respectively... In each study, the effectiveness of the corresponding RL-induced policy was compared against the Random policy. The students were randomly assigned into the two conditions while balancing their incoming competence. Overall, the results from both experiments showed no significant difference between the DQN-Del and Random in Spring 2018 and between the DQN-Inf and Random in Fall 2018 on any measures of learning performance. Therefore, despite the fact that our theoretical results showed that the ECRs of the two RL induced policy look very reasonable, **our empirical results showed they are no better than the Random policy**.

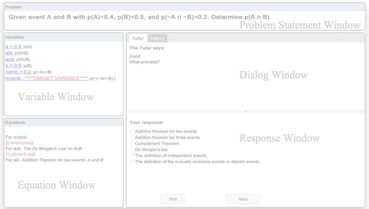
Azizsoltani, Kim, Ausin, Barnes, and Chi, 2019

Two empirical studies were performed to evaluate the effectiveness of DQN-Del in Spring 2018 and DQN-Inf in Fall 2018, respectively... In each study, the effectiveness of the corresponding RL-induced policy was compared against the Random policy. The students were randomly assigned into the tw

☹️ DQN didn't work for this problem

. Overall, the results from both experiments showed no significant difference between the DQN-Del and Random in Spring 2018 and between the DQN-Inf and Random in Fall 2018 on any measures of learning performance. Therefore, despite the fact that our theoretical results showed that the ECRs of the two RL induced policy look very reasonable, **our empirical results showed they are no better than the Random policy.**

Pyrenees



Problem types:

- Worked examples
- Faded worked examples
- Probl

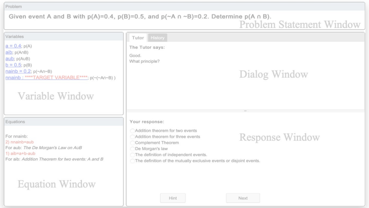
↩️ How to sequence these activities?

For each problem type, the step types:

- Elicit
- Tell

	What it is	Elicit / tell
Worked example	The student observes how the tutor solves a	All-tell
Faded worked example	The student and the tutor co-construct the solution	Mix of tell and elicit
Problem solving	The student solves the problem	All-elicit

Pyrenees



Problem types:

- Worked examples
- Faded worked examples
- Probl

For each p
step types:

- Elicit
- Tell

	What it is	Elicit / tell
Worked example	The student observes how the tutor solves a	All-tell
example	solution	of tell and it
Problem solving	The student solves the problem	All-elicit

↩️ How to sequence these activities?

💡 Let’s try HRL! 💡

RL algorithms are inefficient

“**RL algorithms tend to have poor sample efficiency, often requiring hundreds of episodes to learn simple policies, or hundreds of thousands of episodes to learn more complicated ones** [37]… We overcome this challenge for RS by using proximal policy optimization (PPO), a policy gradient method that leverages deep neural networks to more efficiently learn a scheduling policy [33]” (Bassen et al., 2020, p. 4).

“Prior research in online RL pedagogical policy induction has mainly relied on simulations or simulated students (computational learner models that imitate the learning process of students). **One reason for that is online approaches often need large amounts of exploration to learn an effective policy, which is often too expensive to carry out with real students**” (Zhou et al., 2021)

HRL breaks one problem into a hierarchy of sub-problems

"It has been widely shown that HRL can be more effective and data-efficient than flat RL approaches [6,11,18,22,37]. HRL generally breaks down a large decision-making problem into a hierarchy of small sub-problems and induces a policy for each of them. Since the sub-problems are small, they usually require less data to find the optimal policies. For example, Cuayhuitl et al. induced navigation policies [6] at 3 levels: buildings, floors, and corridors, showing that HRL converged to an optimal policy in much fewer iterations."

Formulation of HRL: Markov Decision Process (MDP)

In RL, an MDP describes a stochastic control process and formally corresponds to a 4-tuple: <S,A,T,R>.

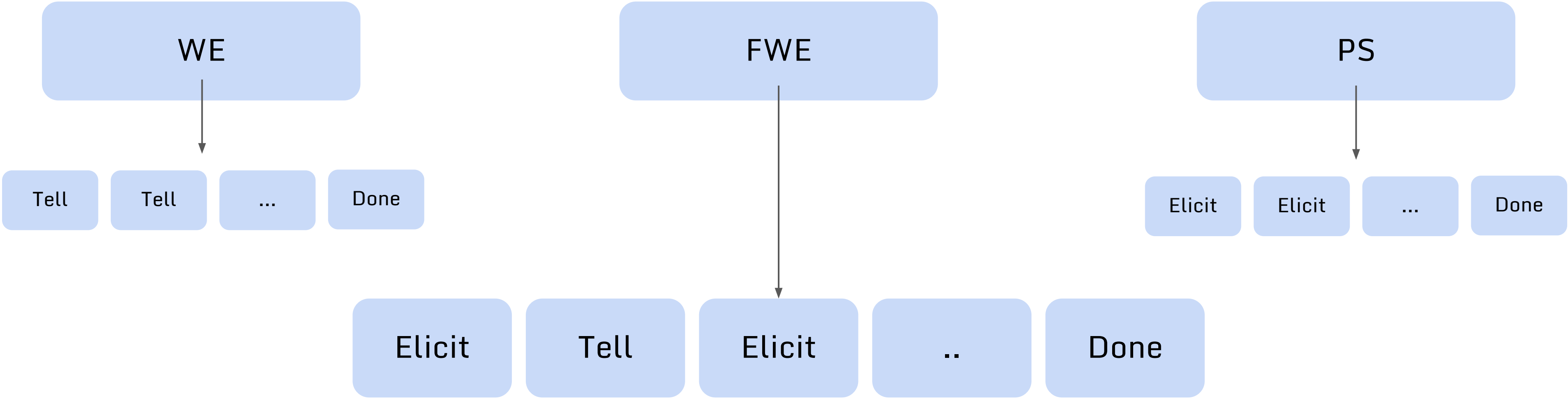
- S: States are vector representations (containing 142 state features)
- A: Actions selected from {Elicit, Tell}
- R: Reward function
- T: Transition probabilities (S -> S’)

Formulation of HRL: Discrete Semi-Markov Decision Process (SMDP)

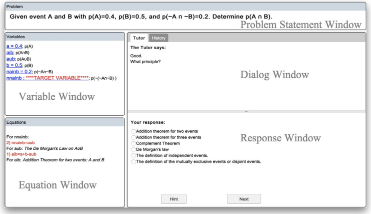
An SMDP adds the idea of *options* or *complex activities* to the MDP formulation. The complex activities are distinct from the primitive actions in that a complex activity may contain multiple primitive actions.

Formulation of HRL: Discrete Semi-Markov Decision Process (SMDP)

An SMDP adds the idea of *options* or *complex activities* to the MDP formulation. The complex activities are distinct from the primitive actions in that a complex activity may contain multiple primitive actions.



Formulating the problem in terms of HRL



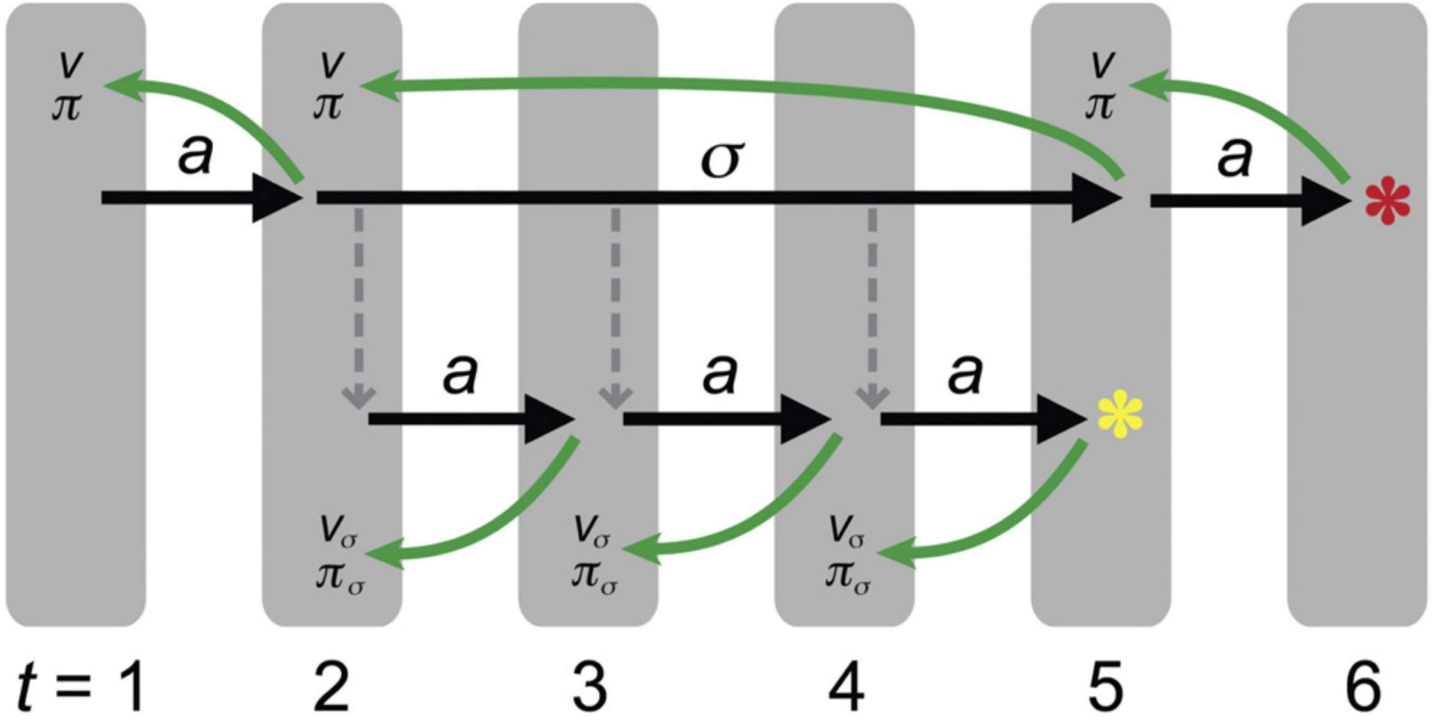
	What it is	Elicit / tell
Worked example	The student observes how the tutor solves a problem	All-tell
Faded worked example	The student and the tutor co-construct the solution	Mix of tell and elicit
Problem solving	The student solves the problem	All-elicit

Higher level

↕️

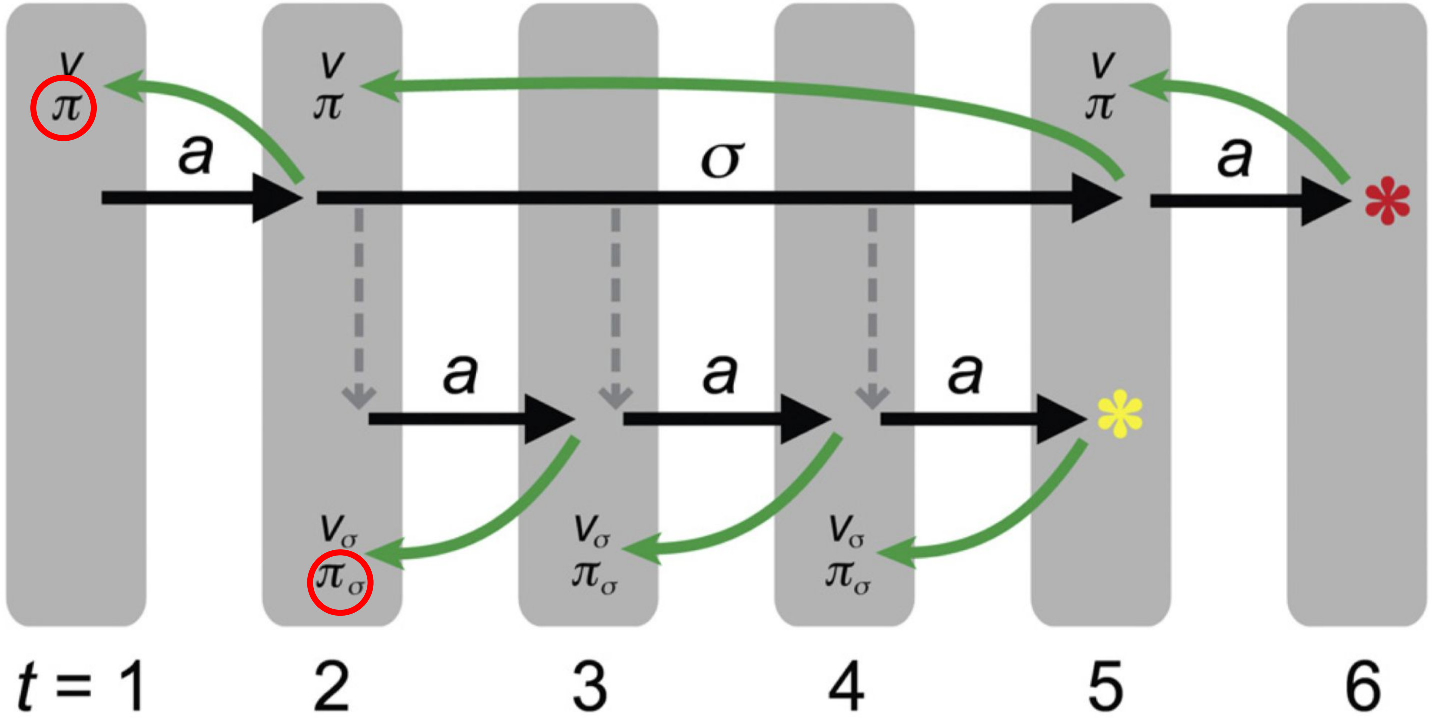
Lower level

Hierarchical Reinforcement Learning



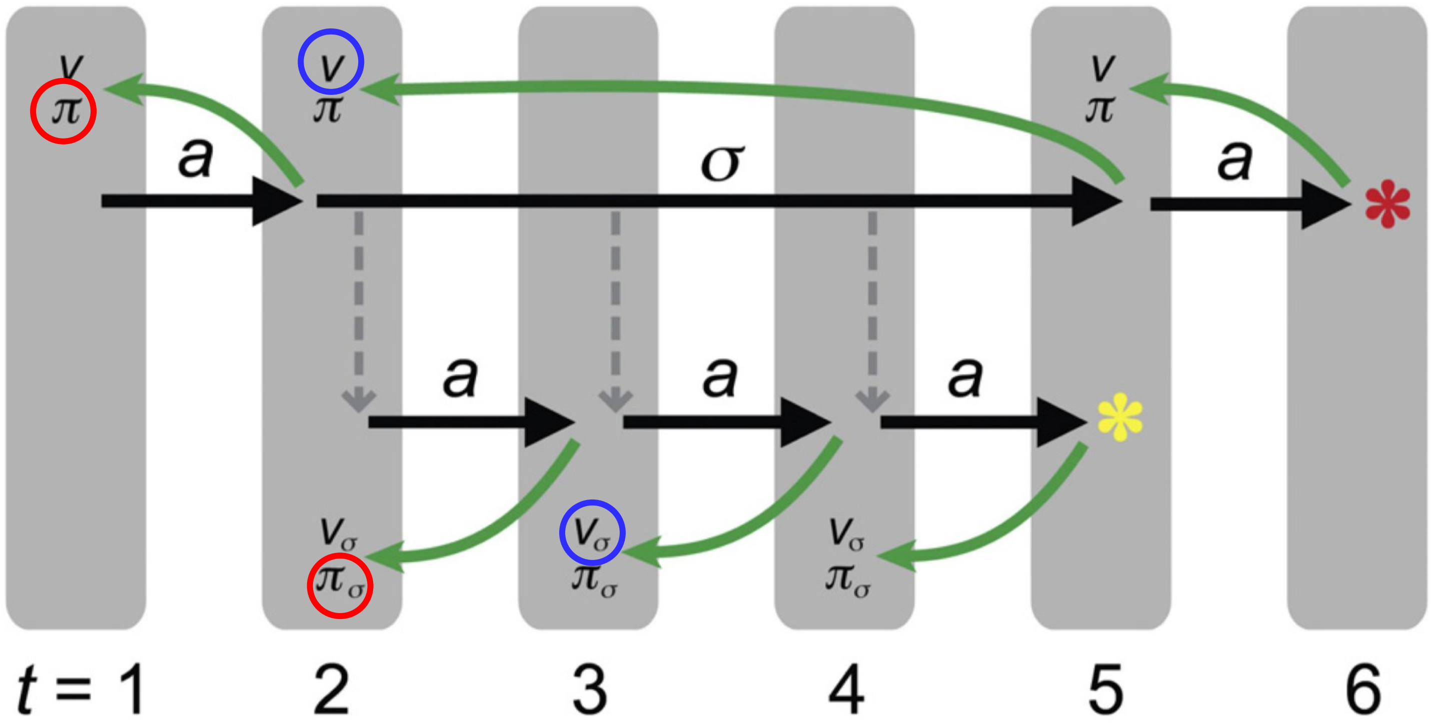
[The promise of hierarchical reinforcement learning]

Hierarchical Reinforcement Learning



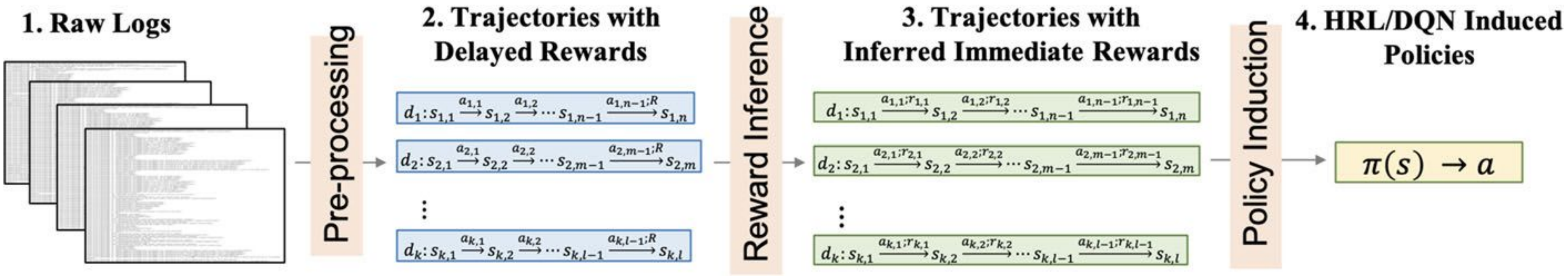
[The promise of hierarchical reinforcement learning]

Hierarchical Reinforcement Learning



[The promise of hierarchical reinforcement learning]

Summary of the HRL policy induction procedure



Methods and Findings

Participants and procedure

Students (N=180) were assigned into three conditions: HRL, DQN, and Random.

- 128 students completed the study. Completion rate did not differ by condition

Students went through four phases: textbook, 14 question pre-test, training on the ITS, and 20 question post-test.

- During training, all three conditions received the same 12 problems in the same order
- 14 of the post-test problems were isomorphic to the pre-test questions, the rest were multiple-principle problems

Partial-Credit grading criteria

Each problem score was defined by the proportion of correct principle applications evident in the solution.

- A student who correctly applied 4 of 5 possible principles would get a score of 0.8

All of the tests were graded in a double-blind manner by a single experienced grader.

Results

Condition	Pre	Iso post	Full post	Adj post	NLG	Time (hours)
HRL(44)	66.4(18.8)	85.8(14.6)	75.3(16.9)	77.7(10.3)	14.3(19.2)	2.19(.64)
DQN(45)	73.9(13.6)	85.2(13.1)	74.2(14.6)	71.2(12.0)	−2.2(29.4)	1.81(.58)
Random(39)	66.3(18.9)	80.5(19.5)	69.0(19.6)	71.4(13.8)	−0.1(35.0)	1.97(.52)

Results

Condition	Pre	Iso post	Full post	Adj post	NLG	Time (hours)
HRL(44)	66.4(18.8)	85.8(14.6)	75.3(16.9)	77.7(10.3)	14.3(19.2)	2.19(.64)
DQN(45)	73.9(13.6)	85.2(13.1)	74.2(14.6)	71.2(12.0)	−2.2(29.4)	1.81(.58)
Random(39)	66.3(18.9)	80.5(19.5)	69.0(19.6)	71.4(13.8)	−0.1(35.0)	1.97(.52)

DQN scored significantly higher than HRL: $t(125) = 2.06$, $p = 0.042$, $d = 0.46$ and Random: $t(125) = 2.01$, $p = 0.046$, $d = 0.46$; but there is no significant difference between HRL and Random: $t(125) = 0.02$, $p = 0.986$, $d = 0.00$.

Therefore, we mainly focus on comparing learning performances that consider the pre-test differences, that is, adjusted post-test and NLG.

Results

Condition	Pre	Iso post	Full post	Adj post	NLG	Time (hours)
HRL(44)	66.4(18.8)	85.8(14.6)	75.3(16.9)	77.7(10.3)	14.3(19.2)	2.19(.64)
DQN(45)	73.9(13.6)	85.2(13.1)	74.2(14.6)	71.2(12.0)	−2.2(29.4)	1.81(.58)
Random(39)	66.3(18.9)	80.5(19.5)	69.0(19.6)	71.4(13.8)	−0.1(35.0)	1.97(.52)

- Significant differences from pre to iso-post for all three conditions with large effect sizes
- No tests reported which compared conditions

Results

Condition	Pre	Iso post	Full post	Adj post	NLG	Time (hours)
HRL(44)	66.4(18.8)	85.8(14.6)	75.3(16.9)	77.7(10.3)	14.3(19.2)	2.19(.64)
DQN(45)	73.9(13.6)	85.2(13.1)	74.2(14.6)	71.2(12.0)	−2.2(29.4)	1.81(.58)
Random(39)	66.3(18.9)	80.5(19.5)	69.0(19.6)	71.4(13.8)	−0.1(35.0)	1.97(.52)

- No tests reported comparing pretest to full posttest

Results

Condition	Pre	Iso post	Full post	Adj post	NLG	Time (hours)
HRL(44)	66.4(18.8)	85.8(14.6)	75.3(16.9)	77.7(10.3)	14.3(19.2)	2.19(.64)
DQN(45)	73.9(13.6)	85.2(13.1)	74.2(14.6)	71.2(12.0)	−2.2(29.4)	1.81(.58)
Random(39)	66.3(18.9)	80.5(19.5)	69.0(19.6)	71.4(13.8)	−0.1(35.0)	1.97(.52)

“Adjusted post-test scores were calculated by adjusting the full-post test scores using the pre-test scores based on a linear model generated by ANCOVA analysis” (Zhou et al., 2021)

- I think the adjusted posttest score is the score predicted by an ANCOVA when the pretest and condition are given
- HRL outperformed both conditions with medium effect sizes

Results

Condition	Pre	Iso post	Full post	Adj post	NLG	Time (hours)
HRL(44)	66.4(18.8)	85.8(14.6)	75.3(16.9)	77.7(10.3)	14.3(19.2)	2.19(.64)
DQN(45)	73.9(13.6)	85.2(13.1)	74.2(14.6)	71.2(12.0)	−2.2(29.4)	1.81(.58)
Random(39)	66.3(18.9)	80.5(19.5)	69.0(19.6)	71.4(13.8)	−0.1(35.0)	1.97(.52)

HRL significantly outperformed both conditions with medium effect sizes

Results

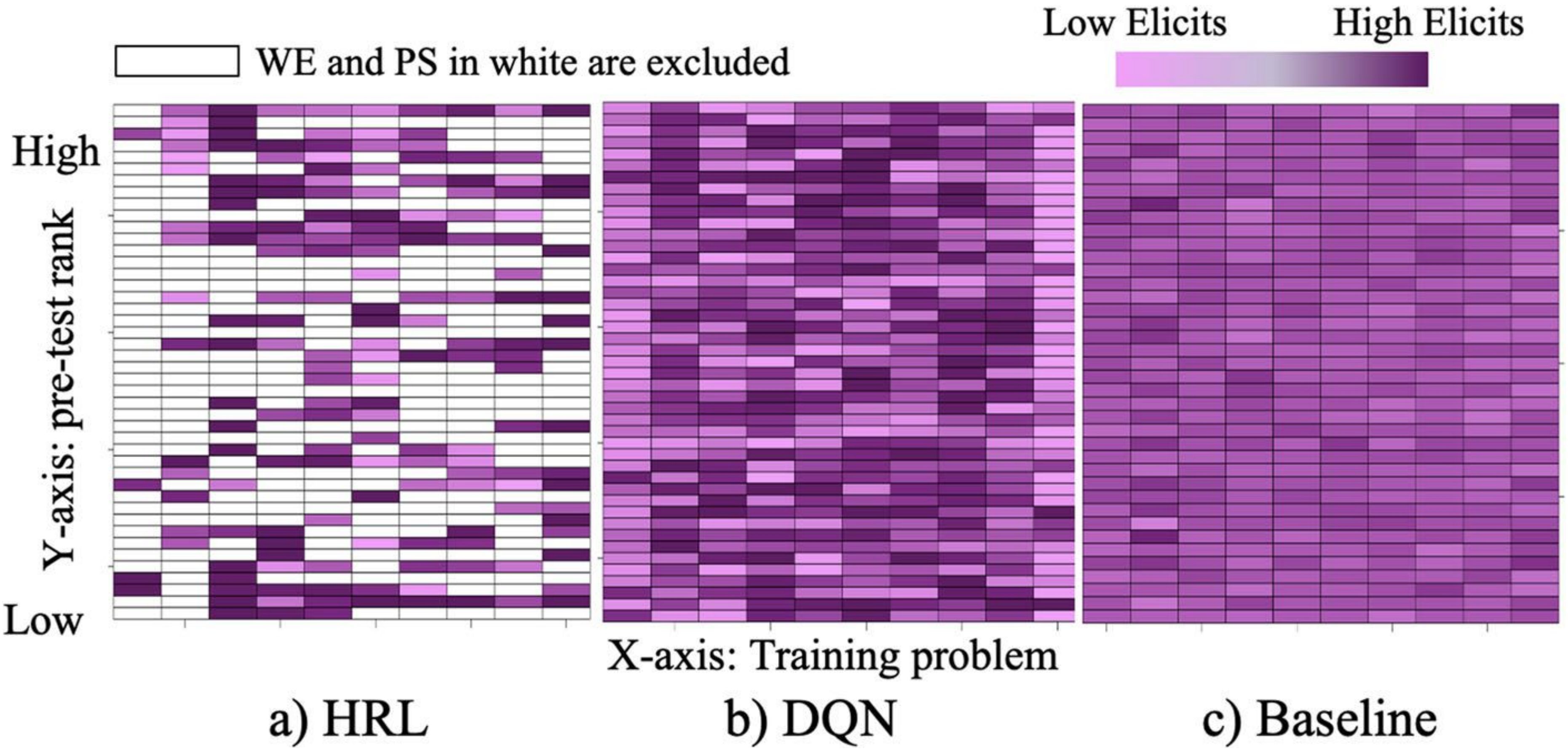
Condition	Pre	Iso post	Full post	Adj post	NLG	Time (hours)
HRL(44)	66.4(18.8)	85.8(14.6)	75.3(16.9)	77.7(10.3)	14.3(19.2)	2.19(.64)
DQN(45)	73.9(13.6)	85.2(13.1)	74.2(14.6)	71.2(12.0)	−2.2(29.4)	1.81(.58)
Random(39)	66.3(18.9)	80.5(19.5)	69.0(19.6)	71.4(13.8)	−0.1(35.0)	1.97(.52)

HRL spent significantly more time on task than DQN ($p < 0.05$) and marginally significantly more time on task than Random ($p < 0.07$)

Results

Condition	Elicit	Tell	Pct Tell
HRL	309.0(60.4)	88.7(66.1)	22.025(15.870)
DQN	205.8(51.6)	188.9(53.0)	47.794(12.974)
Random	200.5(15.9)	203.5(17.4)	50.354(2.482)

- The HRL policy was more likely to choose PS and FWE than WE
- The HRL condition received significantly less tell than the DQN condition: $t(125) = -10.00$, $p < 0.0001$, $d = 1.78$ and the Random condition: $t(125) = -10.60$, $p < 0.0001$, $d = 2.42$.
- The higher SDs of the two RL methods indicate personalization occured



WE and PS in white are excluded
Low Elicits
High Elicits
High
Y-axis: pre-test rank
Low
X-axis: Training problem
a) HRL
b) DQN
c) Baseline

Conclusions

“The results suggest that HRL can be more effective than flat RL in pedagogical policy induction. One possible explanation is that HRL has an explicit problem-level vision. At the problem level, HRL views a problem as an atomic action, and this abstraction has two potential advantages: (1) it aggregates the effects of all steps in a problem and (2) it converts a long step-level sequence into a short problem-level sequence. The aggregation of steps across a problem may provide HRL with a better estimation of the effect of taking a series of steps; while the problem sequence may give HRL a better view of the long-term effects of each problem. Theoretically, flat RL could learn the impact of a problem by aggregating step-level information, but there is no guarantee that it would.”