Adaptively Scaffolding Cognitive Engagement with Batch Constrained Deep Q-Networks

ANONYMIZED

Abstract. Scaffolding student engagement is a central challenge in adaptive learning environments. The ICAP framework defines levels of cognitive engagement with a learning activity in terms of four different engagement modes—Interactive, Constructive, Active, and Passive—and it predicts that increased cognitive engagement will yield improved learning. However, a key open question is how best to translate the ICAP theory into the design of adaptive scaffolding in adaptive learning environments. Specifically, should scaffolds be designed to require the highest levels of cognitive engagement (i.e., Interactive and Constructive modes) with every instance of feedback or knowledge component? To answer this question, in this paper we investigate a data-driven pedagogical modeling framework based on batch-constrained deep Q-networks, a type of deep reinforcement learning (RL) method, to induce policies for delivering ICAP-inspired scaffolding in adaptive learning environments. The policies are trained with log data from 487 learners as they interacted with an adaptive learning environment that provided ICAP-inspired feedback and remediation. Results suggest that adaptive scaffolding policies induced with batch-constrained deep Q-networks outperform heuristic policies that strictly follow the ICAP model without RL-based tailoring. The findings demonstrate the utility of deep RL for tailoring scaffolding for learner cognitive engagement.

Keywords: Deep Reinforcement Learning, Cognitive Engagement, ICAP, Adaptive Learning Environments.

1 Introduction

Adaptive learning environments provide scaffolding in the form of hints, feedback and remediation to improve learning experiences. However, designing effective scaffolds is challenging. Determining how and when to deliver scaffolding in different situations is critical to enabling effective learning experiences [22]. A key factor in adaptive scaffolding is the cognitive engagement of learners. Chi and Wylie [7] describe *cognitive engagement* as an “active learning” process that involves higher-order thinking (e.g., analyzing, synthesizing, evaluating) and is distinguished from motivational or emotional perspectives on engagement.

The ICAP framework provides a taxonomy for categorizing different modes of cognitive engagement [7]. ICAP predicts that learning activities requiring higher levels of cognitive engagement (e.g., peer dialogue, writing a summary) yield improved learning outcomes compared to activities that involve lower levels of cognitive engagement (e.g., listening passively, highlighting text). There is strong evidence in support of the

ICAP theory, and it has been used to guide the design of lesson plans [6] and adaptive learning technologies [20], but it is less clear how to translate ICAP into the design of individual scaffolds. High levels of cognitive engagement require time and student motivation. A direct translation of ICAP may not be optimal for every scaffold and knowledge component in an adaptive learning environment. This raises a natural question: should ICAP be operationalized by adaptively scaffolding cognitive engagement, eliciting higher-order thinking at key moments with the aim of enhancing overall learning outcomes, and, if so, how should we devise models for adaptively scaffolding cognitive engagement?

Recent years have seen growing interest in using reinforcement learning (RL) to induce policies for scaffolding a range of learning processes in adaptive learning environments [5, 8]. Deep RL, which combines RL and deep neural networks, has shown particular promise for this task [3, 32]. For example, Ju and colleagues [15] investigated an adversarial deep RL framework to identify critical moments in students' learning processes. Ausin et al. [3] used offline deep Q-networks with Gaussian process-based inferred rewards to train policies to decide when to provide problem-solving opportunities or worked examples to students in an adaptive learning environment for introductory logic. Results from these studies have demonstrated the promise of deep RL for pedagogical models in adaptive learning environments. However, to date there has been limited work investigating methods for adaptively scaffolding cognitive engagement with deep RL techniques.

In this paper, we introduce a data-driven pedagogical modeling framework based on batch constrained deep Q-networks, a type of deep RL method, to induce policies for scaffolding cognitive engagement in adaptive learning environments. The policies drive ICAP-based feedback and remediation following instructional videos and embedded assessments in a learning environment for training operational command skills. The policies are induced using interaction log data from 487 learners as they engaged with the adaptive learning environment. We compare scaffolding policies induced with batch constrained deep Q-networks with heuristic policies that strictly follow the ICAP model without RL-based tailoring.

2 Related Work

RL provides a natural framework for inducing data-driven scaffolding models to improve student learning experiences. Wang conducted a study with 30 students learning software development concepts in a dialogue-based tutoring system and found that RL-based teaching assistants derive improved teaching strategies by inducing policies through adaptively exploring partially observable states [30]. Georgila and colleagues found that RL-based models can foster increased confidence among learners through adaptive scaffolding to support the development of interpersonal skills [10]. Their results suggested that the induced policies matched, or outperformed heuristic scaffolding models designed by human experts. Similar findings have been reported in other studies and learning environments investigating RL-based pedagogical models [23, 34].

Deep RL techniques show promise for inducing adaptive scaffolding policies that require high-dimensional input state representations [1, 31]. Previous work has shown that employing deep RL techniques to induce scaffolding policies trained with simulated students can yield effective models. For example, Wang and colleagues [31] found that adaptively scaffolding student learning in a narrative-centered learning environment for middle school microbiology using deep RL models trained with simulated students outperforms baseline methods. Other work has investigated offline deep RL methods, where RL models are trained with previously collected data rather than simulations. For example, Azizsoltani and colleagues found that inferring immediate rewards using Gaussian process estimation in deep RL-based pedagogical models can significantly improve learning gains in students [4]. However, deep RL techniques have not been used to induce policies for adaptively scaffolding cognitive engagement with ICAP-inspired feedback and remediation in an adaptive learning environment.

ICAP is a cognitive engagement framework that classifies learning activities into four modes of cognitive engagement: *passive*, *active*, *constructive*, and *interactive* engagement [7]. The ICAP framework predicts that as students become more actively engaged with learning materials, moving from passive to active to constructive to interactive behaviors their learning will increase. Support for the ICAP framework has been found in a number of studies [18, 33, 20]. Mitrovic et al. found that using interactive visualization and prompts to enforce constructive engagement in a video-based learning environment led to high levels of confidence and lower levels of frustration during the learning episode compared to students who engaged in passive learning behaviors [20]. Few studies have investigated whether ICAP-inspired scaffolding applied at a step-based or micro-loop level in adaptive learning environments facilitates learning or how ICAP inspired remediation should be adaptively scaffolded within an adaptive learning environment to support student learning [26, 27]. There has been limited work investigating adaptive scaffolding of cognitive engagement using deep RL techniques in adaptive learning environments.

3 Dataset

To induce data-driven pedagogical models for delivering ICAP-inspired feedback and remediation, we utilize log data collected from an online study involving 487 learners (54% male, 42% female) recruited through Amazon Mechanical Turk who interacted with an adaptive learning environment for training operational command skills. The learning environment was built using ANONYMIZED, an open-source domain-independent framework for designing, deploying, and evaluating adaptive learning technologies [25]. The learning environment includes a series of instructional videos that cover core concepts and principles associated with operational command. Following each video, learners answered a series of multiple-choice questions. An incorrect response to a question prompted the learning environment to deliver ICAP-inspired feedback and remediation that required the learner to either (1) passively re-read a transcription of the video that was just presented in the lesson video, (2) re-read the transcription of the video and actively highlight the portion of text that answered the recall question that was just missed, or (3) re-read the video transcription and constructively summarize the answer to the question in their own words. The learning environment did not have built-in support for the interactive mode of engaging with feedback and remediation, so that component of ICAP was omitted. The active and constructive remediation prompts included expert highlighting/summaries that asked students to self-evaluate the accuracy of their responses. The learning environment also included a "no remediation" prompt that provided learners with a simple feedback message stating they incorrectly answered the question.

After completing a remediation exercise, learners were presented again with the previously attempted question. If they answered the question correctly, then they advanced to the next question or video lesson. Learners continued to receive remediation until they correctly answered the question. The learning environment utilized a random policy to determine the type of ICAP-inspired remediation learners received after each missed question, although a software error caused passive and no remediation instances to be under-sampled.

In all, learners completed 39 embedded assessments, which were distributed across four units that typically take 1-2 hours in total to complete. The adaptive learning environment also included a set of web-based surveys designed to collect demographic information and a set of pre- and post-test items that measured student learning as a result of completing the course.

The resulting dataset included a total of 4,998 instances of ICAP-inspired feedback and remediation. On average, learners received 10 instances of remediation while completing the course (*SD*=12.7; *min*=1, *max*=113). Table 1 summarizes the distribution of remediation instances encountered throughout the course. A paired t-test showed that the pre-test scores (*M*=4.18, *SD*=2.30, *min*=0, *max*=11) and the post-test scores (*M*=8.32, *SD*=2.96, *min*=0, *max*=12) were significantly different (*p*<0.001), implying the adaptive learning environment improved knowledge of operational command concepts and skills among the participants.

Table 1. Distribution of ICAP-based remediation instances.

Remediation	Total	Chapter 1	Chapter 2	Chapter 3	Chapter 4
None	470 (9.40%)	155 (3.10%)	141 (2.82%)	141 (2.82%)	33 (0.66%)
Passive	445 (8.90%)	145 (2.90%)	136 (2.72%)	127 (2.54%)	37 (0.74%)
Active	2074 (41.50%)	684 (13.65%)	587 (11.74%)	639 (12.79%)	166 (3.32%)
Constructive	2009 (40.20%)	626 (12.53%)	606 (12.12%)	611 (12.22%)	166 (3.32%)

4 Adaptive Scaffolding with Batch Constrained Deep Q-Networks

4.1 Deep RL-based Pedagogical Model Architecture

To devise data-driven policies for adaptively scaffolding students' cognitive engagement, we utilized batch constrained deep Q-networks, a variant of the Q-learning algorithm that leverages deep neural network-based function approximation and offline RL techniques. Q-learning is a model-free RL algorithm where the goal is to learn an optimal policy π^* based on optimal action-value function $Q^*(s,a)$ estimated from sample data without use of an explicit model of the task environment [28]. Starting from state $s \in \mathbb{S}$ and taking action $a \in \mathbb{A}$ while getting reward $r \in \mathbb{R}$, the Q values are defined as the expected cumulative reward following a policy π that generates a set of actions at each successive state.

Deep Q-networks (DQNs) approximate Q values using deep neural networks, which provide an effective mechanism for handling large state and action spaces [21]. DQNs follow an off-policy learning approach that involves iteratively sampling from a finite experience buffer to greedily estimate Q values according to the Bellman equation. A loss function (Equation 1) is defined to train a deep neural network to estimate the model's Q-values:

$$Loss(\theta) = \mathbb{E}[(y - Q(s,a;\theta))^2] \qquad (1)$$

where θ represents the set of weights in the neural network, γ represents the RL discount factor, $s' \in \mathbb{S}$ is the next state, and $a' \in \mathbb{A}$ is the next action.

A variant of DQNs is the *Double DQN*, which uses two separate networks to reduce overestimation bias in the DQN by separating the action selection and action evaluation components of the model [11]. This provides improved stabilization and convergence while the model is trained. In Double DQNs, two neural networks with identical architectures are used, namely, the target network ($\bar{\theta}$) and the online network (θ). The online network is trained on every iteration while the target network is frozen for a fixed number of iterations. During training, the online network is used to select the next action and the target network is used to evaluate the Q value of the action:

$$y = r + \gamma Q(s', argmax_{a' \in \mathbb{A}} Q(s',a';\theta); \bar{\theta}) \qquad (2)$$

DQNs are often used with an *experience replay buffer* to keep track of a finite set of recent training observations [21]. During training, transitions are sampled from the buffer randomly. Prioritized experience replay [24] prioritizes the sampling of transitions based upon the current temporal difference errors. This additional priority makes the network more data efficient [12] by ensuring quick convergence. Priority is calculated as follows:

$$t_{priority} = \left|\{r + \gamma Q(s', argmax_{a' \in \mathbb{A}} Q(s',a';\theta); \bar{\theta})\} - \{Q(s,a;\theta)\}\right|^{\omega} \qquad (3)$$

Here, $t_{priority}$ is the priority of a transition t and ω is a hyperparameter.

In batch RL, also known as offline RL, the experience replay buffer remains fixed. This approach is often necessary in RL applications in adaptive learning environments, where a training corpus is collected from students prior to employing RL, and additional data collection is not feasible during the RL process. With limited data, DQNs often suffer from divergence issues due to extrapolating Q values outside of the data distribution. Batch constrained DQNs restrict such extrapolation errors by only allowing actions that are evident in the available data using a probabilistic sampling technique [9]:

$$y = r + \gamma Q(s', argmax_{a' | (a'|s') / \max_{\hat{a}} \pi_b(\hat{a}|s') > \tau} Q(s',a';\theta); \bar{\theta}) \qquad (4)$$

Here, π_b is the policy used to collect the data and τ is a probability threshold.

Deep neural networks within batch constrained DQNs can be implemented using different neural architectures. In this work, we implement two of the most commonly used neural network architectures: fully connected layers and long short-term memory (LSTM) layers. In fully connected layers, each neuron is a perceptron that calculates a weighted sum of the input units to produce an output value through an activation function. All the inputs are connected to all the neurons in the first layer, all of the output of the first layer is fully connected to input neurons of the second layer, and so on until the final output layer is reached.

LSTMs are a specialized version of recurrent neural networks that use long term temporal dependencies to avoid common issues in neural networks such as the vanishing and exploding gradient problems [13]. An LSTM unit consists of a memory cell state and three gates: a forget gate, an input gate, and an output gate. These pieces together control the flow of information during model training. Notably batch constrained DQNs with LSTM networks support sequential input representations, which enables them to keep track of (and forget) previous inputs and hidden states.

4.2 States, Actions and Reward

To formalize the task of inducing a policy for scaffolding cognitive engagement in an adaptive learning environment, we defined a Markov decision process, which involves controlling a set of actions $\mathbb{A}$ based on some state $s \in \mathbb{S}$ to optimize the accumulation of reward $r \in \mathbb{R}$. Markov decision process provide a standardized mathematical representation for RL tasks. We define the state ($\mathbb{S}$), action ($\mathbb{A}$) and reward ($\mathbb{R}$) components of the Markov decision process as follows.

State ($\mathbb{S}$). We devise the state representation by extracting 31 features from learners' log data, which are divided into 3 groups: (1) survey features, (2) video playback features, and (3) remediation engagement features. The survey features include gender, age, education level, content familiarity, domain interest, and pre-test score on a content knowledge assessment. Four video playback features are extracted: time spent on the last video, average time spent on videos, whether learners received automated feedback about their time spent on the last video, and time spent on the feedback. Twenty remediation features are extracted: the previous type of remediation delivered, the total number of remediation instances delivered for each ICAP category, the average time spent engaged in the remediation activity, the average time spent on all previous remediation activities, and features reflecting how long learners spent answering the recall questions. We normalize each feature to range between [0,1] to improve stability when training the DQNs. Batch constrained DQNs require a discrete state space to calculate the probability of each state-action pair. Therefore, we cluster the set of state-action pairs into 5 groups using k-means clustering. We visually select the number of clusters using the distortion elbow method [17]. These clusters are used when sampling for the batch constrained DQN action probabilities.

Action ($\mathbb{A}$). At each pedagogical decision point, there are 4 possible actions (i.e., ICAP-based remediations) that can be selected: constructive, active, passive, or no remediation. Brief descriptions of the action types are given in Section 3.

Reward ($\mathbb{R}$). Each participant completed a 12-item pre- and post-test to assess content knowledge about operational command concepts. We use learners' pre- and post-test scores to calculate normalized learning gains (NLG) associated with each sequence of ICAP-inspired remediation instances [19]. Note that, for each student, we will only have a single NLG value at the end of their episode, which is the delayed reward [4]. Our reward value is a real number and ranges between 0 to 100.

4.3 Evaluation Metrics

In batch constrained DQNs, the optimal policy (π^*) usually has a significantly different distribution of state-action pairs than the behavioral policy (π_b) that was used to collect the training data. Performing RL policy evaluation with data collected under a different policy is known as *off-policy evaluation* [29]. We use two key metrics, Expected Cumulative Reward and Doubly Robust.

Expected cumulative reward (ECR) computes the average expected reward associated with a particular policy π beginning at the initial state $s_i \in \mathbb{S}$ in a given dataset D of the RL task. Specifically, ECR reports the average Q value over all initial states as follows:

$$ECR(D) = \frac{1}{N} \sum_{i=1}^{N} \max_{a \in \mathbb{A}} Q(s_i, a) \qquad (5)$$

where N is the total number of episodes and s_i is an initial state for the ith episode.

Doubly Robust (DR) evaluation [14] is an alternative technique that combines the low variance of importance sampling estimation and the low biases of model-based estimation into a single metric according to the following equation:

$$DR(D) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{\infty} \gamma^t \prod_{\mathrm{l}=0}^{t} \frac{\pi(a_l|s_l)}{\pi_b(a_l|s_l)} \left(R_t^i - Q\left(s_t^i, a_t^i\right)\right) + \gamma^t \prod_{l=0}^{t-1} \frac{\pi(a_l|s_l)}{\pi_b(a_l|s_l)} V(s_t^i) \ (6)$$

The Q function and the value function (V) are based on a given policy π. Doubly Robust provides unbiased estimates if a model is accurate and/or provides low variance estimates if the behavior policy is known.

5 Results

To devise deep RL-based policies for adaptively scaffolding cognitive engagement, we created pedagogical models with batch constrained double DQNs (BCQs) using prioritized experience replay buffers. All models and analyses were implemented in Python using the Scikit-learn and Keras packages with Tensorflow backend. We select $\tau = 10\%$ for the batch constrained hyperparameter because the training data contains 8.9% passive remediations and 9.4% no remediations, resulting in any τ greater than 12% never applying constrained sampling and any value below 8% always enforcing random sampling. We use $\omega = 0.5$ as our priority exponent following prior work [12] and use $\gamma = 0.95$ with a minibatch size of 128. We copy the parameters from the online network to the target network every 100 epochs. All models are trained for 20,000 epochs, repeated 5 times with different random seeds. We do not split our dataset into a training and testing set for validation as it is not necessary in batch RL [29].

We compare two alternative neural network architectures in the BCQ models: fully connected neural networks and long short-term memory networks, which we refer to as FC and LSTM, respectively. The FC BCQ models utilize four fully connected layers with 128 units per layer using ReLU activation functions. The LSTM BCQ models utilize 3 LSTM layers and a fully connected layer at the end, each with 128 units using ReLU activation functions. The output layer always uses a linear activation function to output Q values. For both architectures, a learning rate of 0.001 was utilized with the Adam optimizer [16] and L2 regularization.

For both BCQ models we explore three types of input: input with only the current state (FC-1 or LSTM-1), input with the current state and the previous state (FC-2 or LSTM-2), and input with the current state and previous two states (FC-3 or LSTM-3). In the FC BCQ models, the input states are concatenated and encoded as a single observation. In LSTM BCQ, the input states are provided sequentially.

We include three baseline models for the purpose of comparison in our analyses. All baseline models use a similar architecture as the FC BCQ models, but instead of following a greedy approach to action selection based upon the Bellman equation during model training (i.e., select the action with the maximum Q value in the next state), each baseline model takes a predetermined action while learning the Q functions as follows:

$$y = r + \gamma Q(s', a^{base}; \bar{\theta}) \qquad (7)$$

Here a^{base} is a constructive remediation for the constructive baseline model, and a^{base} is an instance of no remediation for the no remediation baseline model. For the random baseline, a^{base} involves selecting an ICAP-inspired remediation according to a random policy. We select these baselines to serve as heuristic-based models that strictly follow ICAP without RL-based tailoring. τ is set to 0 for the baseline models.

We investigated the learning curves of different models based upon their ECR values. We found LSTM-2 BCQ performs slightly better than the LSTM-1 and LSTM-3 models, whereas little difference was observed between the three FC BCQ models. Based upon these findings, we focus our remaining analyses on FC-1 and LSTM-2.

As shown in Fig. 1, FC-1 and LSTM-2 both perform better than the three baseline models that strictly follow ICAP-inspired heuristics. Among the baselines, we observe that a random policy yields the lowest ECR and a constructive policy yields the highest.

To investigate the models' behavior after they converge to a stable policy, we measured their performances across two metrics during the last 2000 epochs (10% of the total training). Results are shown in Fig. 2. We observe that FC-1 and LSTM-2 score higher than all other baselines in terms of both the ECR and DR evaluation metrics. Running a pairwise Tukey HSD test reveals significant differences between all models except between FC-1 and LSTM-2 for both ECR and DR metrics. These results suggest there are no observed significant differences between the FC-1 BCQ and LSTM-2 BCQ models, but both significantly outperform all of the baselines.

Fig. 3 shows the number of times each type of ICAP-inspired remediation was selected by the different models. The left figure represents FC-1 BCQ and the right represents LSTM-2 BCQ. Once again, we use the last 2000 epochs for this analysis. For

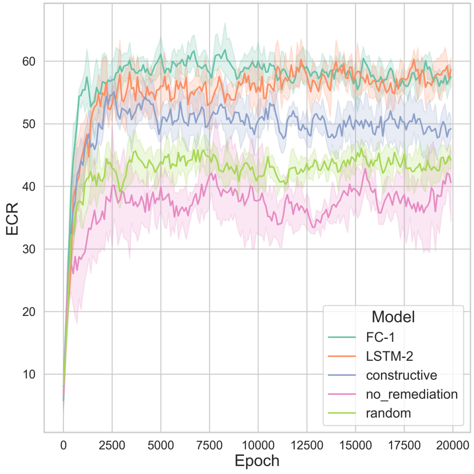

Fig. 1. Comparison between BCQ and ICAP-inspired baseline models.

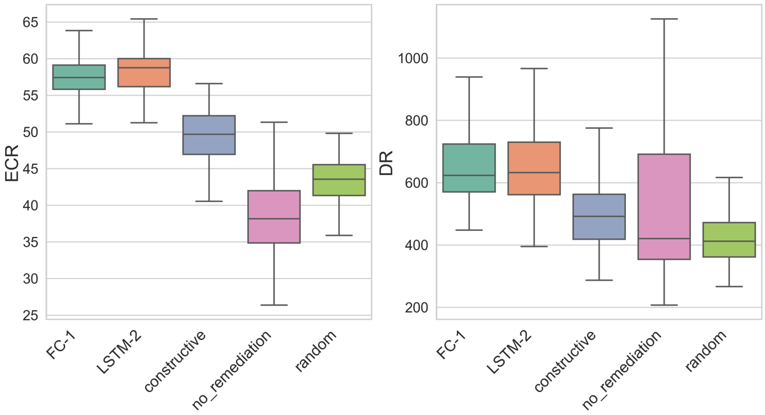

Fig. 2. Performance comparison between BCQ and baseline models during the last 2,000 training epochs.

FC-1, we find that the most frequently selected action is Constructive, with Active being the second most frequent and Passive being the third. (All pairwise comparison *p*-values are less than 0.05). For LSTM-2, there are no significant differences observed between the number of Active, Passive and No Remediation actions. However, both models recommend Constructive remediation significantly more often than other types of remediation (*p*=0.001).

An important observation is that the constructive baseline performs the same or better than the no remediation and random baselines, respectively. We also find that FC-1 BCQ and LSTM-2 BCQ both perform significantly better than the constructive baseline on the ECR and DR metrics. Overall, the results support the ICAP model, and they also suggest that improved learning outcomes may be possible by combining ICAP with RL-based tailoring using batch constrained DQNs.

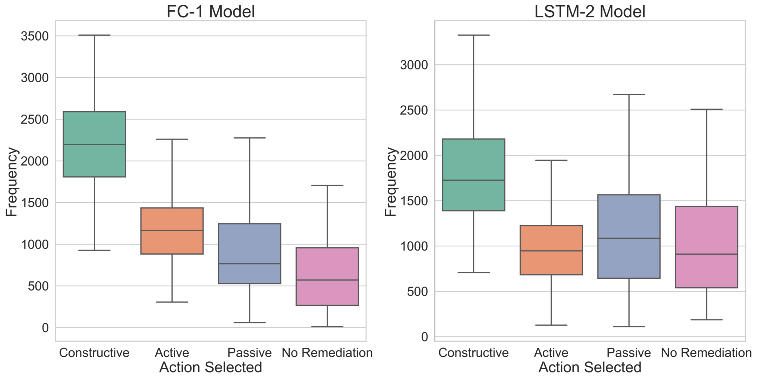

Fig. 3. Frequency of different remediation types selected by BCQ models during the final 2000 training epochs.

6 Conclusions and Future Work

Scaffolding cognitive engagement is a key challenge in adaptive learning environments. The ICAP framework predicts that increased cognitive engagement will yield improved learning, but it is unclear how to best translate the guidance provided by this theory into the design of scaffolding in adaptive learning environments. To address this, we utilized batch constrained deep Q-networks to induce policies for presenting learners with ICAP-inspired scaffolding in an adaptive learning environment. Empirical analysis of converged RL policies indicates that batch constrained deep Q-networks yield adaptive scaffolding policies that outperform heuristic-based policies which exclusively select constructive scaffolding, no scaffolding, or scaffolding at random. Policies induced with batch constrained deep Q-networks also consistently select constructive scaffolding more frequently than active, passive, or no scaffolding, which aligns with the ICAP model. These results (1) support the ICAP theory, and (2) suggest that adaptively scaffolding cognitive engagement based upon deep RL-induced policies is beneficial for optimizing student learning outcomes.

There are several promising directions for future work. First, it will be important to empirically evaluate the induced policies by implementing them in a run-time adaptive learning environment and investigating their impact on student learning outcomes. Second, future research should investigate the impact of additional reward and state features in deep RL-based policies to identify their impact on student learning and engagement. For example, auxiliary rewards and multi-objective reinforcement learning approaches could be used to induce policies that balance student learning and motivational factors in order to enhance learning outcomes and student engagement. Third, it will be instructive to examine how multimodal data such as video-based analysis of student engagement and eye-gaze patterns of student attentional states can be used to augment RL-induced policies for scaffolding student engagement.

References

1. Ai, F., Chen, Y., Guo, Y., Zhao, Y., Wang, Z., Fu, G.: Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. In: Proceedings of the 12th International Conference on Educational Data Mining, pp. 240–245. (2019).
2. Ausin, M.S., Maniktala, M., Barnes, T., Chi, M.: Exploring the impact of simple explanations and agency on batch deep reinforcement learning induced pedagogical policies. In: Proceedings of the 21st International Conference on Artificial Intelligence in Education. pp. 472–485. Springer, Heidelberg (2020). https://doi.org/10.1007/978-3-030-52237-7_38
3. Ausin, M.S., Azizsoltani, H., Barnes, T., Chi, M.: Leveraging deep reinforcement learning for pedagogical policy induction in an intelligent tutoring system. In: Proceedings of the 12th International Conference on Educational Data Mining. pp. 168–177 (2019).
4. Azizsoltani, H., Jin, Y.: Unobserved is not equal to non-existent: Using Gaussian processes to infer immediate rewards across contexts. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. pp. 1974–1980. (2019). https://doi.org/10.24963/ijcai.2019/273
5. Chi, M., VanLehn, K., Litman, D.: Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutorial tactics. In: Proceedings of 10th International Conference on Intelligent Tutoring Systems. pp. 224–234. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13388-6_27
6. Chi, M.T.H., Adams, J., Bogusch, E.B., Bruchok, C., Kang, S., Lancaster, M., et al.: Translating the ICAP theory of cognitive engagement into practice. Cognitive Science 42(6), 1777–1832 (2018). https://doi.org/10.1111/cogs.12626.
7. Chi, M.T.H., Wylie, R.: The ICAP framework: Linking cognitive engagement to active learning outcomes. Educational Psychologist 49(4), 219–243 (2014). https://doi.org/10.1080/00461520.2014.965823.
8. Doroudi, S., Aleven, V., Brunskill, E.: Where's the reward? A review of reinforcement learning for instructional sequencing. International Journal of Artificial Intelligence in Education 29(4), 568–620 (2019). https://doi.org/10.1007/s40593-019-00187-x
9. Fujimoto, S., Meger, D., Precup, D.: Off-policy deep reinforcement learning without exploration. In: Proceedings of the 36th International Conference on Machine Learning, pp. 2052–2062 (2019).
10. Georgila, K., Core, M.G., Nye, B.D., Karumbaiah, S., Auerbach, D., Ram, M.: Using reinforcement learning to optimize the policies of an intelligent tutoring system for interpersonal skills training. In: Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems, pp. 737–745, IFAAMAS, Richland, SC (2019). https://dl.acm.org/doi/abs/10.5555/3306127.3331763
11. Van Hasselt, H., Guez, A., and Silver, D.: Deep reinforcement learning with double q-learning. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, pp. 2094 – 2100. (2016).
12. Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., et al.: Rainbow: Combining improvements in deep reinforcement learning. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, pp. 3215–3222. (2018).
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation. 9(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
14. Jiang, N., Li, L.: Doubly robust off-policy value evaluation for reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning, pp. 652–661. (2016).

15. Ju, S., Zhou, G., Barnes, T., Chi, M.: Pick the moment: Identifying critical pedagogical decisions using long-short term rewards. In: Proceedings of the 13th International Conference on Educational Data Mining, pp. 126–136. (2020).
16. Kingma, D.P., and Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. (2014).
17. Kodinariya, T.M., Makwana, P.R.: Review on determining number of cluster in K-Means clustering. International Journal of Advance Research in Computer Science and Management Studies, 1(6), 90–95 (2013).
18. Lim, J., Ko, H., Yang, J.W., Kim, S., Lee, S., Chun, M.-S., et al.: Active learning through discussion: ICAP framework for education in health professions. BMC medical education. 19(1), Article 47 (2019). https://doi.org/10.1186/s12909-019-1901-7
19. Marx, J.D., Cummings, K.: Normalized change. American Journal of Physics. 75(1), 87–91 (2007). https://doi.org/10.1119/1.2372468
20. Mitrovic, A., Gordon, M., Piotrkowicz, A., Dimitrova, V.: Investigating the effect of adding nudges to increase engagement in active video watching. In: Proceedings of the 20th International Conference on Artificial Intelligence in Education. pp. 320–332. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-23204-7_27
21. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., et al.: Human-level control through deep reinforcement learning. Nature. 518(7540), 529–533 (2015). https://doi.org/10.1038/nature14236
22. van de Pol, J., Volman, M., Oort, F., Beishuizen, J.: The effects of scaffolding in the classroom: Support contingency and student independent working time in relation to student achievement, task effort and appreciation of support. Instructional Science. 43(5), 615–641 (2015). https://doi.org/10.1007/s11251-015-9351-z
23. Sawyer, R., Rowe, J., Lester, J.: Balancing learning and engagement in game-based learning environments with multi-objective reinforcement learning. In: Proceedings of the 18th International Conference on Artificial Intelligence in Education. pp. 323–334. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-319-61425-0_27
24. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. arXiv preprint arXiv:1511.05952. (2015).
25. AnonymousA.
26. AnonymousB.
27. AnonymousC.
28. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. 2nd edn. MIT Press, Cambridge, MA (2018).
29. Thomas, P., Brunskill, E.: Data-efficient off-policy policy evaluation for reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning. pp. 2139–2148 (2016).
30. Wang, F.: Reinforcement learning in a POMDP based intelligent tutoring system for optimizing teaching strategies. International Journal of Information and Education Technology. 8(8), 553–558 (2018). https://doi.org/10.18178/ijiet.2018.8.8.1098
31. Wang, P., Rowe, J., Min, W., Mott, B., Lester, J.: High-fidelity simulated players for interactive narrative planning. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. pp. 3884–3890 (2018). https://doi.org/10.24963/ijcai.2018/540
32. Wang, P., Rowe, J.P., Min, W., Mott, B.W., Lester, J.C.: Interactive narrative personalization with deep reinforcement learning. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. pp. 3852–3858 (2017). https://doi.org/10.24963/ijcai.2017/538

33. Wiggins, B.L., Eddy, S.L., Grunspan, D.Z., Crowe, A.J.: The ICAP active learning framework predicts the learning gains observed in intensely active classroom experiences. AERA Open. 3(2), 1–14 (2017). https://doi.org/10.1177/2332858417708567

34. Zhou, G., Yang, X., Azizsoltani, H., Barnes, T., Chi, M.: Improving student-system interaction through data-driven explanations of hierarchical reinforcement learning induced pedagogical policies. In: Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization. pp. 284–292. ACM, New York (2020). https://doi.org/10.1145/3340631.3394848