









Individualised Mathematical Task Recommendations Through Intended Learning Outcomes and Reinforcement Learning

Alexander Pögel¹ , Katja Ihsberner¹ , Norbert Pengel² ,
Milos Kravcik³ , Martin Grüttmüller¹ , and Wolfram Hardt⁴ 

¹ Leipzig University of Applied Sciences, Karl-Liebknecht-Straße 132,
04277 Leipzig, Germany

{alexander.poegelt,katja.ihsberner,martin.gruettmueller}@htwk-leipzig.de

² Leipzig University, Marschnerstraße 29a, 04109 Leipzig, Germany
norbert.pengel@uni-leipzig.de

³ Educational Technology Lab, German Research Center for Artificial Intelligence
(DFKI), Alt-Moabit 91C, 10559 Berlin, Germany
milos.kravcik@dfki.de

⁴ Department of Computer Engineering, Chemnitz University of Technology,
Straße der Nationen 62, 09111 Chemnitz, Germany
wolfram.hardt@informatik.tu-chemnitz.de

Abstract. Guiding students towards achieving the Intended Learning Outcomes (ILOs) of an academic module as part of a mentoring process presents a significant challenge, as it is important not only to emphasize the necessary skills, but also to consider the ongoing personal progress towards achieving a learning outcome. In addition, most educational content is presented in a ‘one-size-fits-all’ way, without taking into account the individual needs of students. In this paper we present a recommendation system based on Reinforcement Learning (RL) that derives its suggestions from the students’ progress towards achieving the ILOs and the current relevance of the ILOs, according to the specific didactic design of the module. The taxonomy model proposed by Anderson and Krathwohl, serves as the groundwork for abstracting ILO progress, temporal relevance, and the affiliation of recommendation items. In the process of creating a recommendation pool, experts identified the mathematical concept and the taxonomy level addressed by existing e-assessments in order to identify their possible association with ILOs. The RL agent utilizes this dynamic measurement of the student’s ILO progress - measured by the Bayesian knowledge tracing algorithm - to improve its recommendations, contributing to the ongoing personalisation of learning paths. In our evaluation, which utilized a test set of 129 mathematical tasks, the tested RL algorithms significantly outperformed a random baseline, underscoring the potential of this approach to enhance personalized learning within the realm of higher education mathematics.

Keywords: Recommender System · Reinforcement Learning · Intended Learning Outcomes · Mathematical Tasks

1 Introduction

In the evolving landscape of higher education, the demand for delivering personalized learning paths tailored to the distinct needs and capabilities of each student is paramount, as it has been shown to significantly enhance learning outcomes, student satisfaction, motivation, and engagement [6]. However, the development of these personalized learning paths demands significant effort and continuous adaptation from educators, who must meticulously specify the Intended Learning Outcomes (ILOs), which define the knowledge and skills students should have acquired on successful completion of the module, and select and arrange the learning content accordingly to ensure that the learning objects contribute effectively to the achievement of the learning objectives [4]. Moreover, educators must continually monitor and adjust students' learning trajectories to optimize educational impact. To support this complex process, a variety of strategies and methods from the field of Intelligent Tutoring Systems (ITS) offer promising ways to create personalized learning experiences.

This paper presents a novel approach to the design of personalised learning in higher education mathematics by designing a system that focuses on the students' progress towards achieving ILOs, the current relevance and cognitive demand level of topics (based on the didactic model of the module), and the desired challenge level of the students. The aim of this process is to recommend relevant and suitable mathematical tasks, contributing to the effective and efficient achievement of the ILOs.

This work introduces a recommender system leveraging Reinforcement Learning (RL), with a focus on applying and comparing the Proximal Policy Optimization (PPO) [13] and the Deep Q Network (DQN) [11] algorithms. By evaluating their performance in recommending from a collection of 129 mathematical tasks across 312 topics, we aim to establish a benchmark against a random selection method. This comparison provides insights into the potential of combining ILO and RL to personalize learning by aligning task recommendations with students' progress and the challenge level they seek, directly contributing to the achievement of ILOs set by educators.

In the following sections, we first provide a reference to related work. Subsequently, we thoroughly introduce the design and implementation of our approach. Finally, we describe the experiments conducted and discuss them, before concluding the paper.

2 Related Work

A large part of the education technology research has focused on the implementation of the increasingly data-driven systems. However, people tend to trust humans more than algorithms, especially if the task is considered subjective or it requires consideration of individual uniqueness [8]. Therefore, it is crucial to give a close attention to specific learner characteristics in this process. A comprehensive meta-analysis [7] showed that digital tool use had a positive effect on

student learning outcomes and can enhance learning in secondary school mathematics and science.

In recent research on adaptive learning recommendation systems, RL has been utilized to personalize educational processes effectively. An example of individualized educational support is a task recommender for the domain of mathematics based on RL and Item Response Theory (IRT) [12]. The recommendation used the estimated total score and item difficulty estimates derived from IRT. The results suggested that this method allowed for personalized and adaptive recommendations of items within the user-selected threshold while avoiding those with an already achieved target score.

Another approach is a knowledge graph-based, context-aware, recommender system algorithm, which was influenced by agent exploration in RL, for creating sequential learning-path recommendations [1]. The evaluation showed an enriched recommendation based on the learners' context, as well as a better discovery of relevant educational content.

[9] aimed at goal-oriented learning path recommendation and pointed out that previous methods still failed to recommend effective goal-oriented paths due to the under-utilizing of goals. Therefore they presented a Graph Enhanced Hierarchical Reinforcement Learning (GEHRL) framework for goal-oriented learning path recommendation. The framework divides learning path recommendation into two parts: sub-goal selection (planning) and sub-goal achieving (learning item recommendation). They employed a high-level agent as a sub-goal selector to select sub-goals for the low-level agent to achieve. Experiments demonstrated state-of-the-art performance of the framework.

Various RL-based strategies in educational recommendation systems typically utilize a Markov decision framework combined with specific RL algorithms to solve it. For example, [2] integrates the Markov framework with Deep Deterministic Policy Gradients to tailor online course recommendations to individual learner profiles. Similarly, [15] also employs a Markov framework alongside DQN to optimize learning paths by analyzing behavioral data. Additionally, [16] uses a Markov decision process enhanced with DQN to dynamically adapt content recommendations.

3 Design and Implementation

In the context of education, the principle of “Constructive Alignment” emphasizes the importance of aligning the ILOs (which reflect the goals and expectations of the educator for the students), learning activities (objects) and assessment of a module, to ensure that the designed learning experiences are effectively contributing to achieving the desired educational objectives [4]. To operationalize the mapping of ILOs, various models exist, with one of the most well-known being the revised Bloom’s taxonomy by Anderson and Krathwohl [3]. This, comprising 6 cognitive process dimensions (remember, understand, apply, analyze, evaluate, create) and 4 knowledge dimensions (factual knowledge, conceptual knowledge, procedural knowledge, metacognitive knowledge), serves as the foundation for

this work, as it provides a structured framework for classifying learning objectives and associated learning activities (objects). In the context of this research, ILOs are technically defined by a mathematical concept and a certain taxonomy level. It is essential to note that this model is hierarchical, meaning that addressing a specific level implies addressing all levels below it.

Within the domain of higher education mathematics described here, a vast, university-wide pool of tasks exists, utilized, maintained, and further developed by universities in Saxony, which comprises over 5000 mathematics tasks in the area of higher education mathematics that, once classified according to these taxonomies, can be effectively aligned with ILOs to achieve targeted educational objectives. The logical and structured progression of mathematical concepts allows for the strategic reuse of tasks across different learning objectives. This not only demonstrates the interconnectedness of mathematical topics, but also significantly improves the efficiency of resource utilization. By repurposing tasks, educators can enhance the use of existing educational materials, reducing the necessity to develop new tasks for each distinct learning objective and promoting a more sustainable approach to curriculum development. Additionally, within the existing task pool, a significant portion lacks student outcome data, limiting the applicability of conventional recommendation techniques. However, the approach proposed in this paper, focusing on the taxonomy-classification of tasks, enables dynamic and cross-module application even in the absence of outcome data.

For our technical implementation, we utilized two principal reinforcement learning algorithms: Proximal Policy Optimization (PPO) [13] and Deep Q Network (DQN) [11]. The PPO, an On-Policy method, i.e. it directly optimizes the policy currently making decisions, is known for its balance between performance and interpretability. It uses a trust region approach to ensure minimal deviation from the previous policy while seeking improvements [13]. Conversely, DQN is an Off-Policy method, optimizing a policy that is separate from the one generating the current data, that learns from a broader collection of past interactions through experience replay and fixed Q-targets [11]. This approach not only allows DQN to leverage historical data for learning but also enhances stability and efficiency in the learning process. Both PPO and DQN are model-free methods, meaning they learn optimal policies directly from interaction with the environment without constructing a model of the environment, which is particularly advantageous in complex or unknown environments where modeling the dynamics can be challenging. Both methods were implemented through the Ray RLlib library [10], providing a comprehensive framework for managing reinforcement learning experiments, and the Gymnasium package [17], offering a standardized interface for simulating a wide array of environments, thereby enabling the effective training and evaluation of our models.

3.1 Reinforcement Learning Environment

The presented recommendation problem of assigning optimal tasks for the efficient and targeted achievement of ILOs was modelled as a Markov Decision

Process (MDP), a common strategy in recommender systems known for its effectiveness in sequential decision-making and long-term outcome optimization [14]. With MDP, an agent aims to select an action a from the set of all possible actions A in a state s from a set of states S in order to reach a new state s' . This modelling approach is based on the assumption that the transition to the subsequent state s' depends exclusively on the current state s and not on previous states. This assumption is known as the Markov assumption and forms the basis of the MDP. For the state transitions, the agent considers the transition probabilities P and the set of reward functions R associated with these transitions. These reward functions are used to reward or penalise the agent for state changes. The objective of this method is to identify a policy that maximises the expected total reward. Encapsulating the state space, action space, reward functions and transition probabilities in the tuple (S, A, R, T) provides a comprehensive definition of the MDP [18].

State Space S . The state space, also known as the agent's observation space, defines all possible states that can be assumed by the agent's environment. In our modelling, each state s can be defined by a tuple (ut, pl, st) , where:

1. ut : Is a list of tasks that the student has already solved.
2. pl : Is a prioritization list representing the current relevance of each concept, influenced by the progress in the course and its didactic design. This is implemented as a dictionary, where each element includes:
 - (a) The position in the prioritization list.
 - (b) A taxonomy mapping determining the desired cognitive level at which the concept should be addressed in relation to the current state of the course.
3. st : Is the representation of a student, also represented as a dictionary. Each student contains:
 - (a) A challenge level indicating how much the student desires to be challenged.
 - (b) Progress on each concept. This is expressed by a taxonomy mapping describing the extent of the student's mastery of the concept at each cognitive level.

Action Space A . The action space describes all actions that the agent can choose in a given state. In our model, this encompasses all tasks that can be recommended.

Reward Function R . The reward function evaluates an action in a given state by a numerical value. In our model, we have integrated four different rewards that reflect the adaptation of a task to the required relevance, the contribution to the student's progress, the selection of new tasks for the student and the correspondence between the difficulty of a task and the desired challenge level of the student. The assignment of the Relevance Reward ranges from 0 to 100, and the Difficulty Reward and Reward for New Tasks functions range from 0 to 10. The Progress Reward typically ranges from 0 to 100 but can exceed 100 upon the achievement of ILOs, reflecting significant learning milestones.

1. Reward based on the relevance of the selected task is determined by the correspondence between the concepts and the addressed cognitive level recommended by the task and the prioritization list. The reward is computed for each concept, and subsequently, the average across all concepts is considered as the final relevance reward. For each concept i in a recommended task, the reward is calculated as follows:

- (a) Calculation of the *Concept Relevance Factor (CRF)*:

$$CRF_i = \frac{|\text{prioritization list}| - \text{index of concept}_i}{|\text{prioritization list}|}$$

- (b) Calculation of the *Taxonomy Relevance Factor (TRF)*: For each concept i that is both in the task and the prioritization list, the *TRF* is calculated to reflect the alignment between the cognitive levels of the tasks and the requirements from the prioritization list. For cognitive levels j that match exactly ($j \in \text{matching levels}$), a factor of 1 is used. If a task addresses a concept at a lower cognitive level ($j \in \text{lower levels}$) than specified in the prioritization list, a factor of 0.5 is used. Conversely, addressing a concept at a higher cognitive level ($j \in \text{higher levels}$) than specified results in a factor of -1 , penalizing the misalignment.

$$TRF_i = \frac{\sum_{j \in \text{matching levels}} 1 + \sum_{j \in \text{lower levels}} 0.5 + \sum_{j \in \text{higher levels}} -1}{|J|}$$

- (c) Calculation of the *General Relevance Reward (RR)*:

$$RR_i = (TRF_i \cdot CRF_i \cdot \text{weight}_i) \cdot 100$$

2. The reward for contributing to the student's learning progress, whose calculation is described in Subsect. 3.2, is calculated specifically for concepts that appear in the prioritization list and at cognitive levels that are addressed by these prioritized concepts. The calculation is performed for each relevant concept i , with a greater impact on improvements in more relevant concepts.

- (a) Calculate progress: Utilizing the Bayesian Knowledge Tracing (BKT) algorithm, which accounts for the possibility of regression as well as advancement in learning progress, we calculate the difference in student's mastery level before and after completing a recommended task. However, to ensure the reward is positive, negative values are set to 0. The difference is calculated on average over each cognitive level j :

$$diff_i = \frac{\sum_j \max(0, \text{value after}_{i,j} - \text{value before}_{i,j})}{|J|}$$

- (b) Calculation of *General Progress Reward (PR)*:

$$PR_i = diff_i \cdot 100$$

- (c) Calculation of *Achieved Bonus*: This bonus is applied for each cognitive level j where progress exceeds 90% ($j \in \text{achieved}$), denoting mastery. If mastery at any cognitive level regresses below this threshold, the bonus is retracted.

$$PR_i = PR_i + \sum_{j \in \text{achieved}} CRF_i \cdot 100$$

3. The *Difficulty Reward (DR)* is calculated based on the task difficulty for fulfilling the student's challenge level.

$$DR = \begin{cases} 10 & \text{if challenge level} \geq \text{difficulty} \\ 0 & \text{else} \end{cases}$$

4. The *New Task Reward (NTR)* is awarded for each task recommended to the student that they haven't completed yet.

$$NTR = \begin{cases} 10 & \text{if tasks have not yet been completed by the student} \\ 0 & \text{else} \end{cases}$$

Transition Probability T . The transition probability quantifies the chance of moving from one state to another when an action is performed in a particular state. In our context, this represents the probability that the progress of a particular student will change as a result of the recommendation of a task.

3.2 Assessing Student Progress

In intelligent tutoring systems, the Bayesian Knowledge Tracing (BKT) algorithm was initially designed to track how students acquire skills over time. Its purpose is to estimate the likelihood of a student mastering a given skill based on their performance in tasks or tests requiring that skill [5].

Extending BKT's application beyond its original scope, we utilize it to monitor students' progression towards achieving ILOs within academic modules. This adaptation allows us to assess each student's advancement for specific concepts covered by recommended tasks and at every cognitive level associated with these concepts (encompassing all levels below), thereby offering a tailored approach to enhancing educational content recommendations.

The BKT relies on four primary parameters, which can be adapted to our context as follows:

- P_{init} : Initial probability of a student having achieved an ILO before attempting the recommended task
- $P_{transit}$: Probability of transitioning from not achieving to achieving an ILO upon attempting the recommended task
- P_{slip} : Chance of a student making an error despite having achieved the ILO
- P_{guess} : Likelihood of a correct answer without achieving mastery of the ILO

In our implementation, we assume that a student has not previously made progress in any ILO of the module, starting with $P_{init} = 0$. In each recommendation iteration, the test's success is used to estimate the new level of progress, becoming the new P_{init} in the subsequent iteration. To determine $P_{transit}$ the concept weight, the cognitive level addressed in the recommended task and the task difficulty are multiplied together. P_{slip} and P_{guess} were assessed by mathematical experts on the basis of the task structure. The complete calculation is performed as follows:

For a correct solution to the recommended task, we determine $P_{obs=correct}$ using the formula:

$$P_{obs=correct} = \frac{P_{init} \cdot (1 - P_{slip})}{P_{init} \cdot (1 - P_{slip}) + (1 - P_{init}) \cdot P_{guess}}$$

If the solution is incorrect, $P_{obs=wrong}$ is calculated using the following formula:

$$P_{obs=wrong} = \frac{P_{init} \cdot P_{slip}}{P_{init} \cdot P_{slip} + (1 - P_{init}) \cdot (1 - P_{guess})}$$

Subsequently, these calculated probabilities are employed to assess the student's progress within the specific concept i and cognitive level j :

$$Progress_{i,j} = P_{obs} + (1 - P_{obs}) \cdot P_{transit}$$

4 Experiments

4.1 Recommendation Pool

In order to create a recommendation pool to train the RL agent and measure students' progress in the ILOs, an experienced maths expert evaluated 129 existing online maths exercises taken from the learning management system OPAL and the integrated examination software ONYX. This selected set of tasks represents all the exercises available to students in a first-semester Bachelor's course at HTWK Leipzig. The 129 exercises reflect the breadth of the module and cover a total of 312 different concepts. Various criteria were taken into account when selecting the exercises:

id Reference of the task

name Name of the task

link Direct link to the task for the presentation of the recommendation

difficulty Overall difficulty of the task

vector of weights Refers to the differentiated assignment of significance or prominence to different concepts within a task

vector of concepts List of all concepts addressed in the task

vector of process dimensions Mapping of the addressed cognitive process dimension per concept

vector of knowledge dimensions Mapping of the addressed knowledge dimension per concept

slip probability Probability that a student who already has the skills required in the task can fail the task

guess probability Probability that a student who does not have the skills required in the task can solve the task by guessing

To obtain a detailed insight into the classification and assignment of tasks, refer to the following representative example:

Task: Replace the question mark with one of the following quantifier to make the following statement true:

$$\forall p \in \mathbb{N} : ?q \in \mathbb{Z} : \frac{p}{q} \in \mathbb{Q}$$

- A. \forall ... for all
- B. \exists ... there exists (at least one)
- C. \nexists ... there exists none
- D. None of the above

The task engages students with concepts such as quantifiers, specifically the existential and universal quantifiers, and sets, including natural, real and rational numbers. It is classified under the cognitive process dimension of “understand” and the knowledge domain of “conceptual knowledge” for all its concepts, requiring the identification of the correct quantifier to make the statement true. The probability to slip is 0.1 due to the simplicity of the task and the absence of input fields other than the single choice boxes. The guess probability is 0.25, reflecting a chance of guessing the correct answer among the provided options. Overall, the task is classified as having low difficulty, with a rating of 0.1 (10% difficult), making it accessible for those with a foundational understanding of the involved mathematical concepts.

4.2 Experimental Setup

In our study, we train the RL models and observe their performance in simulated environments with virtual students. For this purpose, the environments were created wherein the RL-agent recommends a task to a randomly generated student, for which an outcome is estimated. As described in Subsect. 3.1, each environment consists of a student and a prioritization list for relevant concepts. In our experimental setup, both are generated with random parameters at each initialisation. For the prioritization list, a selection is made by choosing a random number of concepts with a random selection of taxonomy levels at which these concepts should be addressed. In order to simulate the processes of real students, simulated students are generated by assigning them a random ability level that represents their overall capability in handling tasks and a random

progress, which is created by a random selection of concepts for which a random value of progress per cognitive level was determined. Additionally, a function was developed to determine whether a simulated student successfully completes a given task. This determination is based on calculating a success rate, which integrates the student’s ability level, the difficulty of the task, and the discrepancy between the student’s progress and the task’s requirements. The success rate is computed as follows:

$$\text{success_rate} = 0.5 + (a - 0.5) - (d - 0.5) - (\bar{g} - 0.3)$$

where a represents the student’s ability level, which varies from 0 to 1, d indicates the difficulty level of the task, also ranging from 0 to 1. \bar{g} denotes the average gap between the cognitive levels concepts are addressed in the task and the highest level in which the student has achieved any progress in these concepts, with values ranging from -1 to 1. Here, negative values indicate that the student’s mean progress exceeds the cognitive levels addressed in the task. If this success rate exceeds 0.5, the task completion is considered successful, and the student’s learning progress is updated as detailed in Subsect. 3.2.

4.3 Results

To evaluate the effectiveness of the trained models a comparative analysis was conducted against a random baseline across 1000 unique, randomly generated environments, as described in Subsect. 4.2. These settings were created to simulate diverse student progress levels and the relevance of ILOs, incorporating all 129 tasks as potential recommendations. Each model was required to issue a single recommendation per environment.

The outcomes, depicted in Fig. 1, are illustrated through two bar graphs, showcasing the average rewards received from the recommendations. The first graph offers a detailed breakdown by individual reward functions, revealing that PPO outperforms in ‘Relevance Reward’ and ‘Progress Reward’ categories, surpassing both DQN and the random baseline. The noticeable outperformance of both PPO and DQN in the ‘Progress Reward’ (PR) category compared to the random baseline may be attributed to their more effective selection of tasks that contribute to a student’s progression. All three approaches show similar performance in the ‘New Task Reward’ category. This can be attributed to the test environment setup, where each model was required to issue only a single recommendation per environment, ensuring that the recommended task is inherently new and thus all models invariably score the full 10 points in this category. However, for ‘Difficulty Reward’ (DR), DQN’s recommendations stand out, outperforming those of PPO and the random baseline, which suggests that DQN may have a better strategy for gauging or responding to task difficulty levels.

The second graph compares the total rewards obtained by the algorithms, highlighting that PPO achieves the highest improvement, outperforming the random baseline by an average of 36.61%. Meanwhile, DQN also demonstrates a notable advancement, being 30.49% better than the random selection. These

findings underscore the efficacy of both RL algorithms and highlight the potential of RL to enhance personalized learning pathways in higher education mathematics.

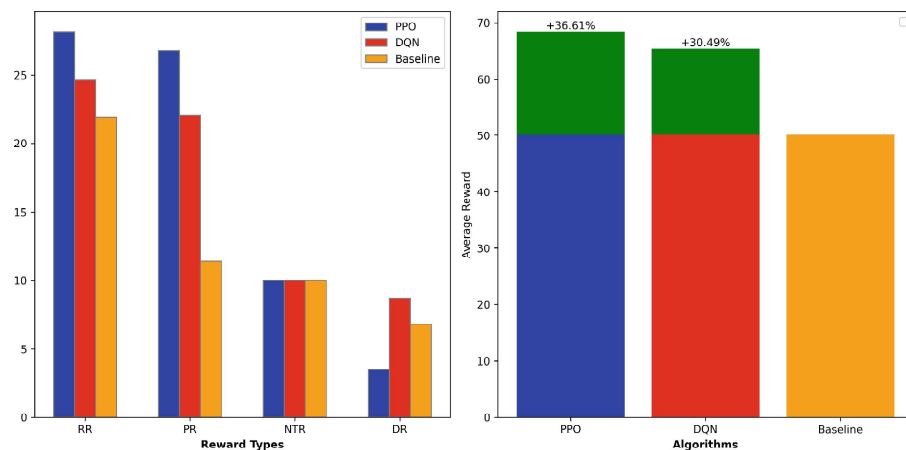


Fig. 1. Comparison of algorithms PPO and DQN, along with a random baseline, across four different reward functions (RR - Relevance Reward, PR - Progress Reward, NTR - New Task Reward, DR - Difficulty Reward) in the left bar charts. The right bar chart illustrates the cumulative reward comparison for PPO, DQN, and the baseline

5 Discussion

In this study, we introduced a RL-based recommendation system designed to support students in achieving ILOs. Our results showcase the system's efficacy, with the PPO algorithm outperforming a random baseline by 36.61%, and the DQN algorithm also showing a commendable performance improvement of 30.49% over the baseline. These findings highlight the practical utility of our approach in personalizing learning experiences.

Despite the promising outcomes, the reliance on expert-classified data introduces a potential for errors or subjective biases, suggesting a need for future studies to involve multiple independent raters to enhance the classification's reliability. A further limitation noted is the selection bias introduced by utilizing all tasks from a specific module as the recommendation pool. This approach does not account for the entire spectrum of possible tasks, potentially skewing the algorithms' performance when faced with a completely different set of tasks. Acknowledging this, future work should investigate the system's robustness and performance across a broader range of tasks, ensuring its effectiveness in universally enhancing learning outcomes. The results of this study, derived from a simulated environment, underscore the need for real-world testing to validate the RL-based system's effectiveness in actual educational settings. Conducting

practical tests will be crucial for future work to identify potential challenges and confirm the system’s impact on learning outcomes. Unlike related studies, such as [12], which base recommendations on statistical analyses, our approach provides the capability to recommend tasks for which no outcome data is available. Furthermore, the adaptability of the proposed methods, such as BKT and taxonomy classification, suggests broader applicability to diverse domains, opening avenues for future research. It is conceivable to extend this approach to create a generalized mathematics recommendation system suitable for various academic modules with distinct ILOs. Although our approach requires the classification of learning materials based on concepts and taxonomy levels, which poses a scalability challenge, it uniquely allows for module-transcendent use of these materials. In contrast to systems that recommend learning materials tailored to specific skills, our method enables a more individual selection of materials aligned with course-specific learning objectives. This facilitates contributions from multiple stakeholders, such as instructors, who can add to a shared pool of resources that others may use effectively.

6 Conclusion

This paper introduces an effective recommendation system designed for academic settings, assisting students attaining the predefined learning objectives of a module while considering the learner’s individual progress and specific preferences for challenging tasks. By comparing the PPO and DQN algorithms against a random baseline, we have shown that both algorithms are more efficient in selecting appropriate tasks, indicating the potential of RL for enhancing the relevance of educational content recommendations.

While the outcomes are encouraging, we acknowledge the study’s initial reliance on simulated environments. Future work will focus on real-world applications to better understand the system’s practical benefits and limitations. This step is crucial for assessing the system’s real impact on student learning and for making necessary adjustments to enhance its effectiveness.

By moving towards implementing and testing in actual educational settings, we aim to validate the system’s potential to personalize learning at a broader scale. This research contributes to the ongoing discussion on integrating AI in education, highlighting the importance of further exploration to fully realize its benefits.

Acknowledgments. The authors would like to thank the German Federal Ministry of Education and Research (BMBF) for their kind support within the project *Personalisierte Kompetenzentwicklung und hybrides KI-Mentoring* (tech4compKI) under the project id 16DHB2211.

References

1. Abu-Rasheed, H., Weber, C., Dornhöfer, M., Fathi, M.: Pedagogically-informed implementation of reinforcement learning on knowledge graphs for context-aware learning recommendations. In: Viberg, O., Jivet, I., Muñoz-Merino, P., Perifanou, M., Papathoma, T. (eds.) EC-TEL 2023. LNCS, vol. 14200, pp. 518–523. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-42682-7_35
2. Agrebi, M., Sendi, M., Abed, M.: Deep reinforcement learning for personalized recommendation of distance learning. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds.) WorldCIST 2019. AISC, vol. 931, pp. 597–606. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16184-2_57
3. Anderson, L.W., Krathwohl, D.R.: A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives, complete Addison Wesley Longman Inc., New York (2001)
4. Biggs, J.: Enhancing teaching through constructive alignment. *Higher Educ.* **32**(3), 347–364 (1996). <https://doi.org/10.1007/BF00138871>
5. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User-Adap. Interact.* **4**(4), 253–278 (1994). <https://doi.org/10.1007/BF01099821>
6. Fariani, R.I., Junus, K., Santoso, H.B.: A systematic literature review on personalised learning in the higher education context. *Technol. Knowl. Learn.* **28**(2), 449–476 (2023). <https://doi.org/10.1007/s10758-022-09628-4>
7. Hillmayr, D., Ziernwald, L., Reinhold, F., Hofer, S.I., Reiss, K.M.: The potential of digital tools to enhance mathematics and science learning in secondary schools: a context-specific meta-analysis. *Comput. Educ.* **153**, 103897 (2020)
8. Kizilcec, R.F.: To advance AI use in education, focus on understanding educators. *Int. J. Artif. Intell. Educ.* **34**, 1–8 (2023)
9. Li, Q., et al.: Graph enhanced hierarchical reinforcement learning for goal-oriented learning path recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 1318–1327 (2023)
10. Liang, E., et al.: RLlib: abstractions for distributed reinforcement learning. <https://doi.org/10.48550/arXiv.1712.09381>. <http://arxiv.org/abs/1712.09381>
11. Mnih, V., et al.: Playing Atari with deep reinforcement learning (2013). <https://doi.org/10.48550/arXiv.1312.5602>. <http://arxiv.org/abs/1312.5602>
12. Orsoni, M., Pögel, A., Duong-Trung, N., Benassi, M., Kravcik, M., Grützmüller, M.: Recommending mathematical tasks based on reinforcement learning and item response theory. In: Frasson, C., Mylonas, P., Troussas, C. (eds.) ITS 2023. LNCS, vol. 13891, pp. 16–28. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-32883-1_2
13. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. <https://doi.org/10.48550/arXiv.1707.06347>
14. Shani, G., Brafman, R.I., Heckerman, D.: An MDP-based recommender system. *J. Mach. Learn. Res.* **6**, 1265–1295 (2005)
15. Tan, C., Han, R., Ye, R., Chen, K.: Adaptive learning recommendation strategy based on deep q-learning. *Appl. Psychol. Meas.* **44**, 251–266 (2020). <https://doi.org/10.1177/0146621619858674>
16. Tang, X., Chen, Y., Li, X., Liu, J., Ying, Z.: A reinforcement learning approach to personalized learning recommendation systems. *Br. J. Math. Stat. Psychol.* **72**, 108–135 (2019). <https://doi.org/10.1111/bmsp.12144>. <https://onlinelibrary.wiley.com/doi/abs/10.1111/bmsp.12144>

17. Towers, M., et al.: Gymnasium, March 2023. <https://doi.org/10.5281/zenodo.8127026>. <https://zenodo.org/record/8127025>
18. Uther, W.: Markov decision processes. In: Sammut, C., Webb, G.I. (eds.) Encyclopedia of Machine Learning, pp. 642–646. Springer, New York (2010). https://doi.org/10.1007/978-0-387-30164-8_512