Neural Computing and Applications (2023) 35:24369–24388
https://doi.org/10.1007/s00521-023-08989-w

ORIGINAL ARTICLE

Sim-GAIL: A generative adversarial imitation learning approach of student modelling for intelligent tutoring systems

Zhaoxing Li[1] · Lei Shi[1,2] · Jindi Wang[1] · Alexandra I. Cristea[1] · Yunzhan Zhou[1]

Received: 25 January 2023 / Accepted: 22 August 2023 / Published online: 3 October 2023

Abstract

The continuous application of artificial intelligence (AI) technologies in online education has led to significant progress, especially in the field of Intelligent Tutoring Systems (ITS), online courses and learning management systems (LMS). An important research direction of the field is to provide students with customised learning trajectories via student modelling. Previous studies have shown that customisation of learning trajectories could effectively improve students' learning experiences and outcomes. However, training an ITS that can customise students' learning trajectories suffers from cold-start, time-consumption, human labour-intensity, and cost problems. One feasible approach is to simulate real students' behaviour trajectories through algorithms, to generate data that could be used to train the ITS. Nonetheless, implementing high-accuracy student modelling methods that effectively address these issues remains an ongoing challenge. Traditional simulation methods, in particular, encounter difficulties in ensuring the quality and diversity of the generated data, thereby limiting their capacity to provide intelligent tutoring systems (ITS) with high-fidelity and diverse training data. We thus propose Sim-GAIL, a novel student modelling method based on generative adversarial imitation learning (GAIL). To the best of our knowledge, it is the first method using GAIL to address the challenge of lacking training data, resulting from the issues mentioned above. We analyse and compare the performance of Sim-GAIL with two traditional Reinforcement Learning-based and Imitation Learning-based methods using action distribution evaluation, cumulative reward evaluation, and offline-policy evaluation. The experiments demonstrate that our method outperforms traditional ones on most metrics. Moreover, we apply our method to a domain plagued by the cold-start problem, knowledge tracing (KT), and the results show that our novel method could effectively improve the KT model's prediction accuracy in a cold-start scenario.

Keywords Student modelling · Generative adversarial imitation learning · Intelligent tutoring systems

1 Introduction

Intelligent tutoring systems (ITS) are increasingly incorporating artificial intelligence (AI) technologies, including machine learning and deep learning, which could effectively offer customised learning trajectories for each student based on their prior knowledge and learning activities [1]. Research in cognitive science has shown that there is a strong relationship between, amongst others, the sequence of learning materials and learning outcomes [2]. In a traditional online learning platform, there is only one single static linear learning trajectory provided to students. In this one-size-fits-all approach, students may lose their motivation and even drop out of the course, due to anxiety or boredom encountered in the learning process [3]. Research on customised learning trajectories for students has been

✉ Lei Shi
lei.shi@newcastle.ac.uk

Zhaoxing Li
zhaoxing.li2@durham.ac.uk

Jindi Wang
jindi.wang@durham.ac.uk

Alexandra I. Cristea
alexandra.i.cristea@durham.ac.uk

Yunzhan Zhou
yunzhan.zhou@durham.ac.uk

[1] Department of Computer Science, Durham University, Durham, UK

[2] Open Lab, School of Computing, Newcastle University, Newcastle upon Tyne, UK

emerging in the ITS field. However, developing an ITS that can provide students with customised learning trajectories requires a large amount of data for training the system, which is time-consuming and costly [4], long known to be requiring a large amount of manual labour from education providers (instructors, authors, etc.) [5]. Although many mature ITSs have sufficient data to train algorithms, a large number of emerging ITSs are still suffering from a lack of training data in the early stages of development, also known as the *cold start problem* [6].

To tackle these challenges, previous studies have proposed various methods for simulating student learning trajectories (i.e., generating massive student learning behavioural data) that can be used to train an ITS. Early simulated student behaviour proposals stemmed from the aim at automatic validation of educational interventions via a sandbox method [7]. More recently, Jarboui et al. [8] attempted to model student trajectory sequences into a Markov Decision Process, but in real educational scenarios, only a few ITS can provide all the feature data consistent with a Markov Decision Process (e.g., the reward function of the ITS agent). Zimmer et al. [9] defined reward functions to build reinforcement learning agents to generate student trajectories, but this method requires building different reward functions for different datasets, which makes it difficult to generalise. Besides, humans' psychological responses to learning trajectories and reward mechanisms are difficult to simulate. This leads to circumstances where student simulation methods may not be able to simulate student learning trajectories sufficiently. Anderson et al. [10] proposed a student simulation method based on behavioural cloning (BC), the simplest form of Imitation Learning, which aims to solve the abovementioned problems where the reward is sparse and hard to define [11]. Whilst promising, BC-based methods only learn from the few features collected in student data, and the actions that algorithms are able to model can be very limited.

Motivated by the gap in prior literature identified above, the research question of this paper is: *How to build an efficient student simulation method that can generate massive student learning data, which can be used for ITS training?*

To answer this research question, we propose *Sim-GAIL*, a generative adversarial imitation learning (GAIL) approach to student modelling. Our Sim-GAIL method can be used to generate simulated student data to solve the lack of data and cold-start problems in ITS training.

Furthermore, to showcase its efficiency, we compare our Sim-GAIL with the two main student modelling methods used in the ITS field, the RL-based and the BC-based student modelling approach, using data from the very recent and largest ITS dataset, EdNet [12]. We extract action and state features to train the models. We analyse and compare performance using action distribution evaluation, cumulative reward evaluation (CRE), and two offline-policy evaluation (OPE) methods, which include importance sampling (IS) and Fitted Q Evaluation (FQE). Moreover, we apply our method's generated data in an ITS cold-start scenario. The experimental results show that our method outperforms the two traditional RL-based and BC-based baseline methods and could improve the training efficiency of the ITS in a cold-start scenario.

The *main contributions* of this work lie in the following three aspects:

1. We propose Sim-GAIL, a student modelling approach, to generate simulation data for ITS training.
2. It is the first method, to the best of our knowledge, that uses Generative Adversarial Imitation Learning (GAIL) to implement student modelling to address the challenge of lacking training data and the cold-start problem.
3. The experiments demonstrate that a trained Sim-GAIL could simulate real student learning trajectories very well. Our method outperforms traditional RL-based and BC-based methods on most metrics and can improve the training efficiency in cold-start scenarios.

Thus, the *advantages of Sim-GAIL* include its ability to effectively generate data resembling real student behaviours, address the cold-start problem, demonstrate superior performance on various metrics, efficiently converge to an optimal policy, and offer scalability and generality across different datasets and applications.

This paper is structured as follows. Section 2 introduces the background of reinforcement learning, imitation learning (including behavioural cloning), and student modelling. Section 3 demonstrates the dataset, data preprocessing, and model architecture. Section 4 outlines the experiments and baseline models. Section 5 discusses the evaluation methods and the experimental results based on action distribution, offline policy (OP) evaluation, expected cumulative rewards (ECR) evaluation, and knowledge tracing (KT). Section 6 discusses our findings and future works. Section 7 draws conclusions.

2 Background and literature review

Before analysing current competitors of the proposal for student modelling for generating training data presented in this paper, we show the current state of the underlying methodologies: Markov decision process, reinforcement learning, imitation learning, and finally, the method at the basis of our proposal, generative adversarial imitation learning.

2.1 Markov decision process and reinforcement learning

The Markov decision process (MDP) is the standard method for sequential decision-making (SDM) [13]. The sequential decision-making models can generally be seen as an instance of the Markov decision process. Reinforcement learning is also typically regarded as an MDP [14]. Therefore, in this section, we introduce MDP and then reinforcement learning.

2.1.1 Markov decision process (MDP)

MDP is a mathematical model of sequential decision used to generate stochastic policies and rewards, achievable by an agent in an environment where the system state exhibits Markov properties [15]. MDPs are represented as a set of interacting objects, namely agents and environments, with components including states, actions, policies, and rewards. In an MDP model, the agent observes the present state of the environment and takes actions on the environment in accordance with the policy, thereby changing the state of the environment and getting rewards. The ultimate goal of the agent is to reach the maximum cumulative reward, which is achieved using a reward function [16]. Figure 1 shows the structure of the MDP.

2.1.2 Reinforcement learning (RL)

RL is a type of machine learning method that enables an agent to learn a policy by taking different actions in an interactive environment, in order to maximise cumulative rewards. It could be defined as the tuple of $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where S is defined as the state of the environment, A represents actions of the agent, $\mathcal{P} : S \times \mathcal{A} \times S \rightarrow [0, 1]$ represents the transition probabilities of actions from the current state to the next state and $R : S \times \mathcal{A} \times S \rightarrow \mathbb{R}$ denotes the reward function. The goal of an RL agent is to achieve maximum cumulative rewards. However, the drawback of traditional RL methods lies in its computational overhead, brought by repeated interactions between the agent and the environment.

2.2 Imitation learning (IL)

Different from RL, where the agent learns by interacting with the environment to obtain the maximum rewards, IL is a method of learning policy that involves emulating the behaviour of experts' trajectories [17], instead of leveraging an explicit reward function as in RL.

2.2.1 Behavioural cloning (BC)

BC considers the learning of policy under supervised learning settings, leveraging state-action pairs [18, 19]. Albeit simple and effective, BC suffers from the heavy reliance on extremely large amounts of data [20, 21], without which a distributional mismatch, often referred to as covariate shift [22, 23], would occur, due to compounding errors and stochasticity in the environment during test time.

2.2.2 Apprenticeship learning (AL)

Different from BC, AL instead tries to identify features of the expert's trajectories that are more generalisable and to find a policy that matches the same feature expectations with respect to the expert [24]. Its goal is to find a policy that performs no worse than the expert across a class of cost functions. The main limitation of AL is that it cannot imitate the expert trajectory well, due to the restricted class of cost functions. Specifically, when the true cost function does not lie within the cost function classes, the agent cannot be guaranteed to outperform the expert.

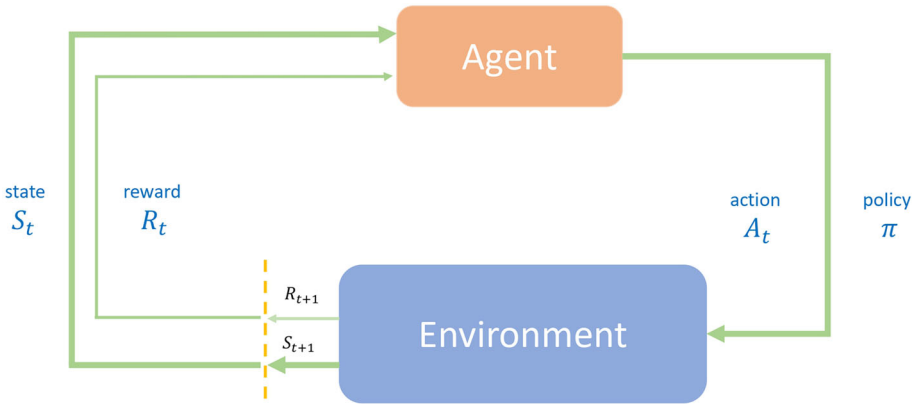

Fig. 1 Framework of the Markov decision process

2.3 Generative adversarial imitation learning (GAIL)

GAIL addresses the drawbacks of RL and AL effectively [20], by borrowing the idea of Generative Adversarial Networks (GANs) [25]. It is derived from a type of Imitation Learning, called Maximum Causal Entropy Inverse Reinforcement Learning (MaxEntIRL) [26].

Figure 2 shows the mechanism of GAIL. Integrating GANs into imitation learning allows for the Generator never to be exposed to real-world examples, enabling agents to learn only from experts' demonstrations. In GAIL, the Discriminator is trained with the objective of distinguishing the generated trajectories from real trajectories, while the Generator, on the other hand, attempts to imitate the real trajectories, to fool the Discriminator into thinking it is actually one of them.

2.4 Student modelling

As the traditional one-size-fits-all approach can no longer satisfy student learning needs, it leads to increased demands for customised learning [27, 28]. Various student modelling methods have been proposed, which are generally classified as integrating expert knowledge-based or data-driven methods [29, 30]. Knowledge-based methods refer to utilising human knowledge to address issues that would normally require human intelligence [7, 31]. Data-driven methods simulate students' learning trajectories through massive student learning records data [6, 32, 33].

2.4.1 Expert knowledge-based methods

The majority of the studies in this field involve building different forms of student models, to train a reinforcement learning (RL) agent [34]. Glesias et al. proposed a Markov Decision Process based on expert knowledge, to train student models [35]. Doroud et al. [34] suggested an RL-based agent method rooted in cognitive theory, to optimise the sequencing of the knowledge components (KCs). The reward function of this method is based on pre- and post-test scores, taken as a metric, and termed Normalised Learning Gain (NLG). However, this metric needs evaluation by human participants, which is excessively human

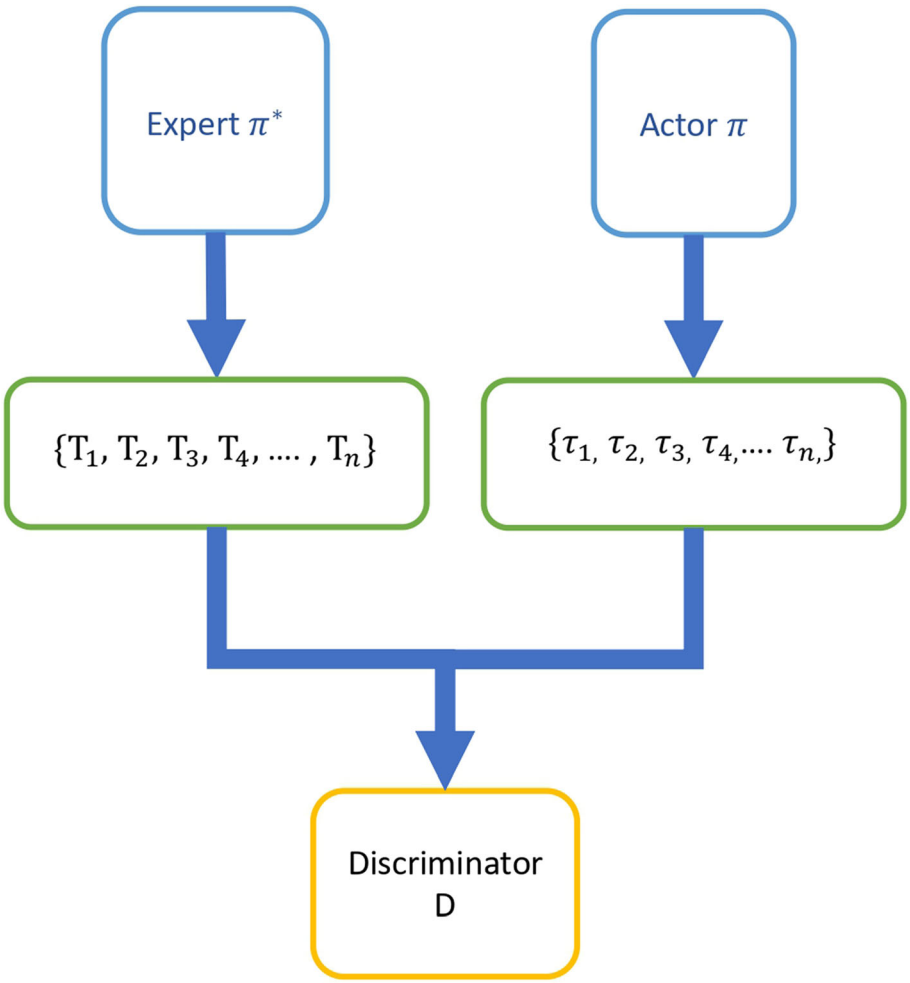

Fig. 2 Mechanism of generative adversarial imitation learning

resource-intensive. Yudelson et al. [36] proposed a 'Student Simulation' method based on Bayesian Knowledge Tracing (BKT), which could train a 'sim student' to imitate real students' mastery of different knowledge. Segal et al. suggested a student simulation method based on the Item Response Theory (IRT) [37], which could respond to different reactions to courses at different difficulty levels [38]. Azhar et al. [39] introduced an application of reinforcement learning (RL) for optimising the learning sequence modelling of online learning materials, which is an end-to-end pipeline to automatically derive and evaluate robust representations of students' interactions and policies for content sequencing in online education.

2.4.2 Data-driven methods

Compared with integrating expert knowledge-based methods, data-driven methods could better simulate real students' learning trajectories and more effectively reduce biases [13]. There have been some studies [40–42] aiming to build student simulation methods based on data-driven MDP approaches. For example, Beck et al. proposed a population student model (PSM) based on a linear regression model that could simulate the probability of the student's correct response [43]. However, this method requires a high-quality dataset from real ITS platforms. Limited by the quantity of high-quality datasets, the previous data-driven model struggled to keep up with the expanding requirements of ITS development. Li et al. proposed a student behaviour simulation method based on a Decision Transformer, to generate student behaviour data for ITS training [6, 33]. Emond et al. [44] proposed an adaptive instructional system (AIS) as a self-improvement system. It presented a methodological approach that incorporates three concurrent research activities: Bayesian networks modelling of learning processes, knowledge elicitation from expert instructors, and the use of simulated learners and tutors for exploring AIS design options. On the other hand, with the further development of ITS research, more and more high-quality datasets, such as EdNet [12], have been published in recent years, which can be used to achieve a high-quality data-driven student simulations. However, collecting data like the EdNet dataset is extremely time-consuming and labour-intensive. How to improve the effectiveness of ITS with small data volumes or in a cold-start scenario is still a problem that needs to be addressed.

3 Method

In this section, we introduce the methodology for the research described in this paper. First, we describe the EdNet dataset we use, in Sect. 3.1. In Sect. 3.2, we show how we preprocess the data in EdNet, to obtain the features we need. We then articulate the framework of our SIM-GAIL method in Sect. 3.3.

3.1 Dataset

We adopt EdNet [12], the largest dataset in the ITS field, for our experiments. This dataset comprises students' interaction log data with an ITS, which can be used to extract the state and action representation. EdNet is a massive benchmark dataset of interactions between students and a MOOC learning platform called SANTA.[1] SANTA is a TOEIC (Test of English for International Communication) learning platform in South Korea, and the EdNet dataset was collected by *Riiid! AI Research*.[2] There are 131,417,236 interaction logs collected from 784,309 students in 13,169 exercises over two years, as shown in Table 1. The interaction logs for each student are recorded in an independent CSV (Comma-Separated Values) file. EdNet is a four-layer hierarchical dataset, structured from KT1 to KT4, according to the granularity of interactive actions. KT1 only contains simple information, such as question and answer pairs and elapsed time. Based on the information in KT1, to provide correlation information between student behaviour and question-and-answer sequences, EdNet adds detailed action records to KT2, such as watching video lectures and reading articles. In KT3, actions such as choosing response options and reviewing explanations are added to KT2, which can be used to infer the influence of different learning activities on students' knowledge states. KT4 includes the finest detailed action information, such as purchasing courses, and pausing and playing video lectures, which could be used to investigate the impact of sparse key actions on overall learning outcomes.

3.2 Data preprocessing

The problems involving decision-making processes are transformed into MDPs in general [8] (see Sect. 2.1.1). In this experiment, we view the students' sequential decision-making trajectories as a Markov Decision Process. Extracting the *action space* and *state space* of the real students' data is essential for building an effective student simulation method using MDP. Next, we show how we

[1] https://www.aitutorsanta.com.

[2] https://www.riiid.co.

Table 1 Statistics of the EdNet

Number of interactions	131,417,236
Number of students	784,309
Number of exercises	13,169

explore the data and extract the *action space* and *state space*.

3.2.1 Action space

There are 13,169 questions, 1021 lectures, and 293 kinds of skills in EdNet [12]. However, there are no criteria for

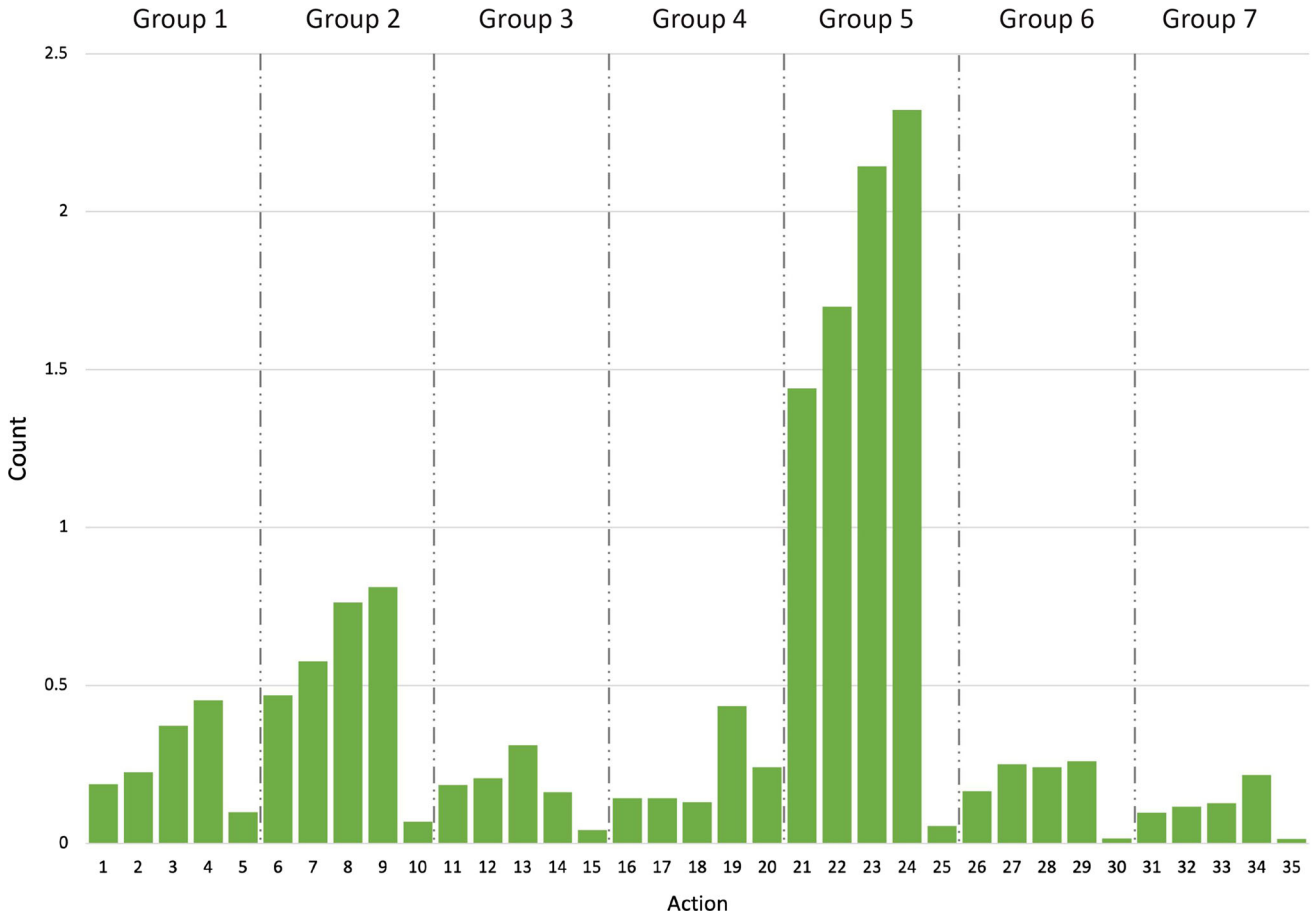

Fig. 3 Analysis of the action distribution of the EdNet dataset

Table 2 State feature representation

State feature	Description
'correct_so_far'	The ratio of correct responses
'av_time'	The cumulative average of the elapsed time
'av_fam'	Average familiarity of all parts
'topic_fam'	Familiarity with the current part
'prev_correct'	Numbers of correct answers in previous responses
'steps_in_part'	Counts of student learning steps
'lects_consumed'	Numbers of lectures a student has learnt

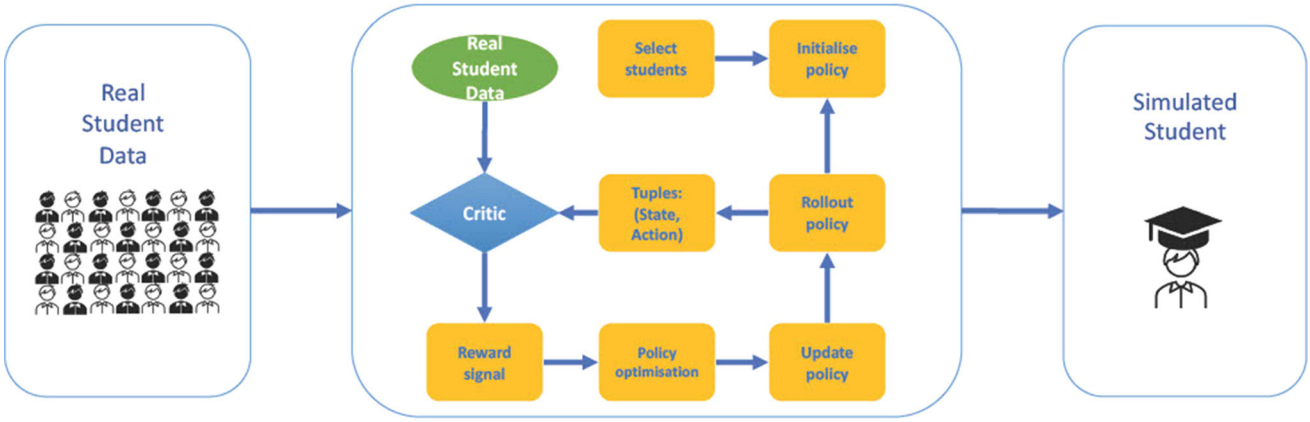

Fig. 4 The Sim-GAIL pipeline

separating these courses into different parts. Bassen et al. [4] proposed a method to group knowledge concepts, based on the assumption that each part was grouped by domain experts' experience. Inspired by this method, we divide the lectures and questions space of the agent into 7 groups. However, as the division into 7 groups is of a too coarse granularity for the action space, we further use the method proposed in [38], and divide the difficulty of the questions from 1 to 4 by the answer correctness rate, obtained by comparing the students answer logs and the correct answers. Some lectures lack a difficulty ranking and are therefore assigned a default difficulty value of 0. Hence, all action spaces are divided into 5 difficulty levels, with 7 groups, and thus 35 action types in total. Figure 3 shows the distribution of the 35 types of actions in EdNet. In each group, the action types include 4 questions from difficulty levels 1 to 4, and 1 lecture. Taking Group 1, for example, actions 1 to 4 correspond to questions with different difficulty levels, and action 5 corresponds to lectures where the difficulty level cannot be defined, which is set as 0. As shown in Fig. 3, the rest of the groups follow this pattern.

3.2.2 State space

EdNet records the interaction data for each student with the system, in separate CSV files, via UNIX timestamps. Therefore, most of the state features obtained from EdNet are longitudinal and temporal. Previous works have shown that different state feature choices could make a large difference in the performance of the algorithms [40, 45]. We select state features that are widely chosen in similar simulated student works [4, 35, 40, 42]. Transitions between these selected states represent students' learning trajectories. Table 2 shows the *features* we select from EdNet: 'correct_so_far' is the proportion of the correct answer to the number of all activities attempted; 'av_time' is the cumulative average of the elapsed time spent on each action; 'av_fam' denotes the average familiarity of the 7 groups; 'topic_fam' denotes the familiarity with the current group; 'prev_correct' denotes the number of correct answers in the previous group; and 'steps_in_part' counts student learning steps in the current group. Compared to previous works [4, 40], we select more *state features*, which could potentially simulate the students' trajectories in real situations more effectively.

3.3 Sim-GAIL model architecture

Our Sim-GAIL model is built upon generative adversarial imitation learning (GAIL) [20], which aims to solve the problem of Imitation Learning of having difficulty in dealing with constant regularisation and not being able to match occupancy measures in large environments. Equation (1) demonstrates the optimal negative log loss, distinguishing between the pair: state π and action π_E.

$$\psi^*_{\mathrm{GA}}(\rho_\pi - \rho_{\pi_E}) = \max_{D\in(0,1)^{S\times\mathcal{A}}} \mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1-D(s,a))], \tag{1}$$

where ψ_{GA^*} is the average of the real trajectories' data, and D is the discriminative classifier. Using causal entropy H as the policy regulariser, the following procedure can be derived:

$$\underset{\pi}{\text{minimise}}\, \psi^*_{\mathrm{GA}}(\rho_\pi - \rho_{\pi_E}) - \lambda H(\pi) = D_{\mathrm{JS}}(\rho_\pi, \rho_{\pi_E}) - \lambda H(\pi). \tag{2}$$

This equation combines imitation learning (IL) and generative adversarial networks (GAN) [25]. Generator S generates trajectories that are passed to Discriminator D. The Generator's goal is to make it less likely for the Discriminator to differentiate the real trajectories and those generated by the Generator, whilst the Discriminator's goal is to distinguish between them. The Generator achieves the best learning effect when the Discriminator fails to

recognise the generated trajectories. Lastly, ρ_{π_E} in Eq. (1) is the occupancy measure of the real trajectories.

$$\mathbb{E}_{\pi}[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi) \quad (3)$$

There is a function approximation of π and D. TRPO [46] is used to find a saddle point (π, D), which decreases the value of Expression (3). To decrease the expected cost, we use the cost function $c(s,a) = \log D(s,a)$. As classified by Discriminator, the cost function will move toward real trajectories-like regions of the state-action space, to achieve the training goal of Discriminator.

Figure 4 shows the pipeline of Sim-GAIL. Real student data from EdNet is processed by the methods introduced in Sect. 3.2 and fed into the GAIL module (middle part) to create a simulation policy that could be used for training the 'sim student' (right part). The middle part is described in Algorithm 1. We start by initialising the policy θ and Discriminator D. At each iteration, we sample real student trajectories from the dataset and update the Discriminator parameters using the Adam gradient [47]. Then, we take a policy update step using the TRPO rule, to decrease the expected cost [46]. At last, we take a KL-constrained natural gradient step, to train the Discriminator.

'correct_so_far', 'ave_time', 'av_fam', 'topic_fam', 'pre_correct', 'step_in_part', and 'lects_consumed'. Then, a subset of the trajectories is randomly picked and controlled with the policy. After that, for each student's trajectory, a set of action-state pairs, are extracted from the observation policy. The policy outputs a student action, responding to the state feature at each timestamp. In this way, we created the simulation that represents the ground-truth policy, used to train other methods on.

For the experimental setup, we use an auto-encoder to process the data. Sim-GAIL is implemented using the PyTorch framework. We train the model on the seven features mentioned before using the 1000 students' interaction logs.

4.2 Baseline models

Among the few studies that could be selected as baseline methods, the current state-of-the-art top performers so far are the Behavioural Cloning based method proposed by Torabi [48] and the Reinforcement Learning-based method proposed by Kumar [49]. Therefore, we use these two methods as the baselines for the experiments.

Algorithm 1 Algorithm of Sim-GAIL.

Require: Real students trajectories, $\tau_E \sim \pi_E$; initiating the policy θ and Discriminator D

1: **for** each $i = 0, 1, 2, \ldots$ **do**
2: Sample student trajectories $\tau_i \sim \pi_{\theta_i}$
3: Update the parameters w_i to w_{i+1} in Discriminator
4: $\hat{\mathbb{E}}_{\tau_i}\left[\nabla_w \log\left(D_w(s,a)\right)\right] + \hat{\mathbb{E}}_{\tau_E}\left[\nabla_w \log\left(1 - D_w(s,a)\right)\right]$
5: Take a policy step from θ_i to θ_{i+1} with cost function $\log\left(D_{w_{i+1}}(s,a)\right)$
6: $\hat{\mathbb{E}}_{\tau_i}\left[\nabla_\theta \log \pi_\theta(a \mid s) Q(s,a)\right] - \lambda \nabla_\theta H\left(\pi_\theta\right)$
7: where $Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}\left[\log\left(D_{w_{i+1}}(s,a)\right) \mid s_0 = \bar{s}, a_0 = \bar{a}\right]$
8: **end for**

4 Experiments

In this section, we introduce the experimental setup in our Sim-GAIL method and the two baseline methods that serve as comparator.

4.1 Sim-GAIL

In order to simulate the real student learning behaviour on a real platform, we build a simulator, to play back the real student learning trajectories from EdNet, selected using a stochastic policy. Specifically, we first sample the real student trajectories from Ednet. The state includes

4.2.1 Behavioural cloning (BC)

The first baseline is the behavioural cloning (BC)-based method, proposed by Torabi [48]. This model has shown good performance in the task of simulating users' behaviour from observations. Similarly, we employ a mixture regression (MR) approach [50], which is a Gaussian mixture of the actions and states, to process the data features. For fairness of the comparison, we use the same action-state pair data to train the Sim-GAIL and BC-methods, with data extracted from EdNet (see Sect. 4.1). The supervised learning method is applied to train the policy and Adam optimisation [47], with a batch size of 128.

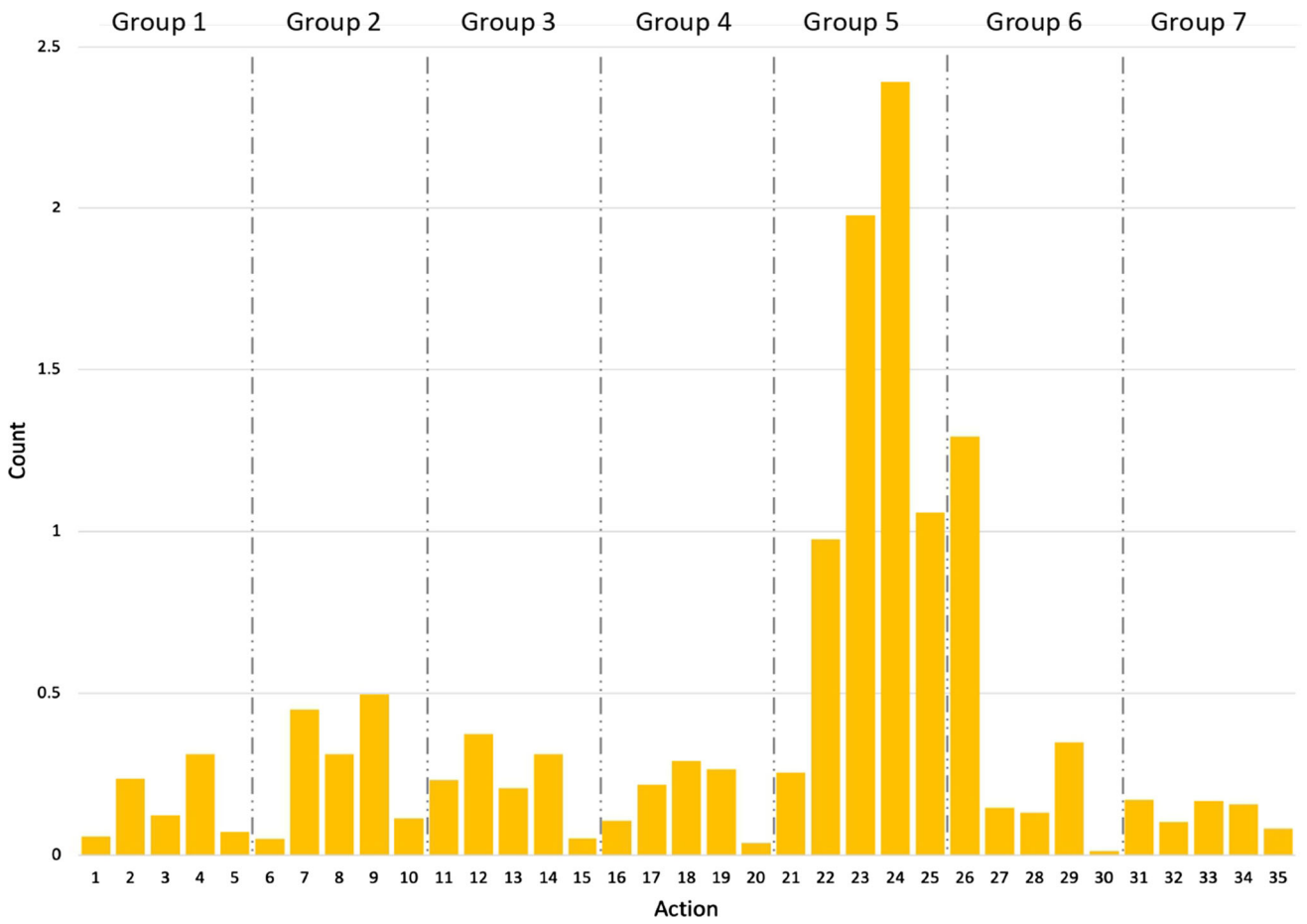

Fig. 5 Action distribution of the Sim-GAIL model

4.2.2 Reinforcement learning (RL)

The second baseline is the reinforcement learning (RL)-based method proposed by Kumar [49], which uses the conservative Q-learning (CQL) approach. EdNet does not contain any students' prior- or post-test scores. Hence, we use the method proposed by Azharet al. [51] to build a reward function, based on the historical logs of students' scores. More specifically, we use the correctness of the students' responses as the reward function. If a student's response is correct, a positive reward will be given; otherwise, a negative reward will be provided. Moreover, we integrate the difficulty levels of the questions. We set the rewards from 1 to 4, based on the difficulty level of the activity. Thus, if the student's responses match the correct answers, they get a positive reward of 1 to 4; and if no, they receive a negative reward of -1 to -4. The Dynamic Programming (DP) [52] method is used to train the model. More specifically, we utilise a Policy Iteration (PI) method to train the agent. This process could be separated into two repeated stages: the first is evaluating the value of every state in the finite MDP according to the current policy. The second is using the Bellman Optimality equation [53] to make the policy iteration based on the current policy.

5 Evaluation

Our evaluation includes two parts: The first part compares the Sim-GAIL with the two baseline models, and the second part uses Knowledge Tracing models to evaluate the effect of the Sim-GAIL.

In the first part of the evaluation, to better evaluate Sim-GAIL and its performance relative to traditional models, we develop our own comprehensive evaluation framework. Since the most critical elements for a Markov Decision Process are *action*, *reward*, and *policy*, as shown in Fig. 1, we build a novel framework, to evaluate the efficiency of Sim-GAIL and two baseline models from these three aspects, respectively. In particular, we identify action distribution, to evaluate the *action*, expected cumulative rewards, to evaluate the *reward*, and offline policy, to evaluate the *policy*. The first metric, the action distribution, is the similarity of distributions between the generated

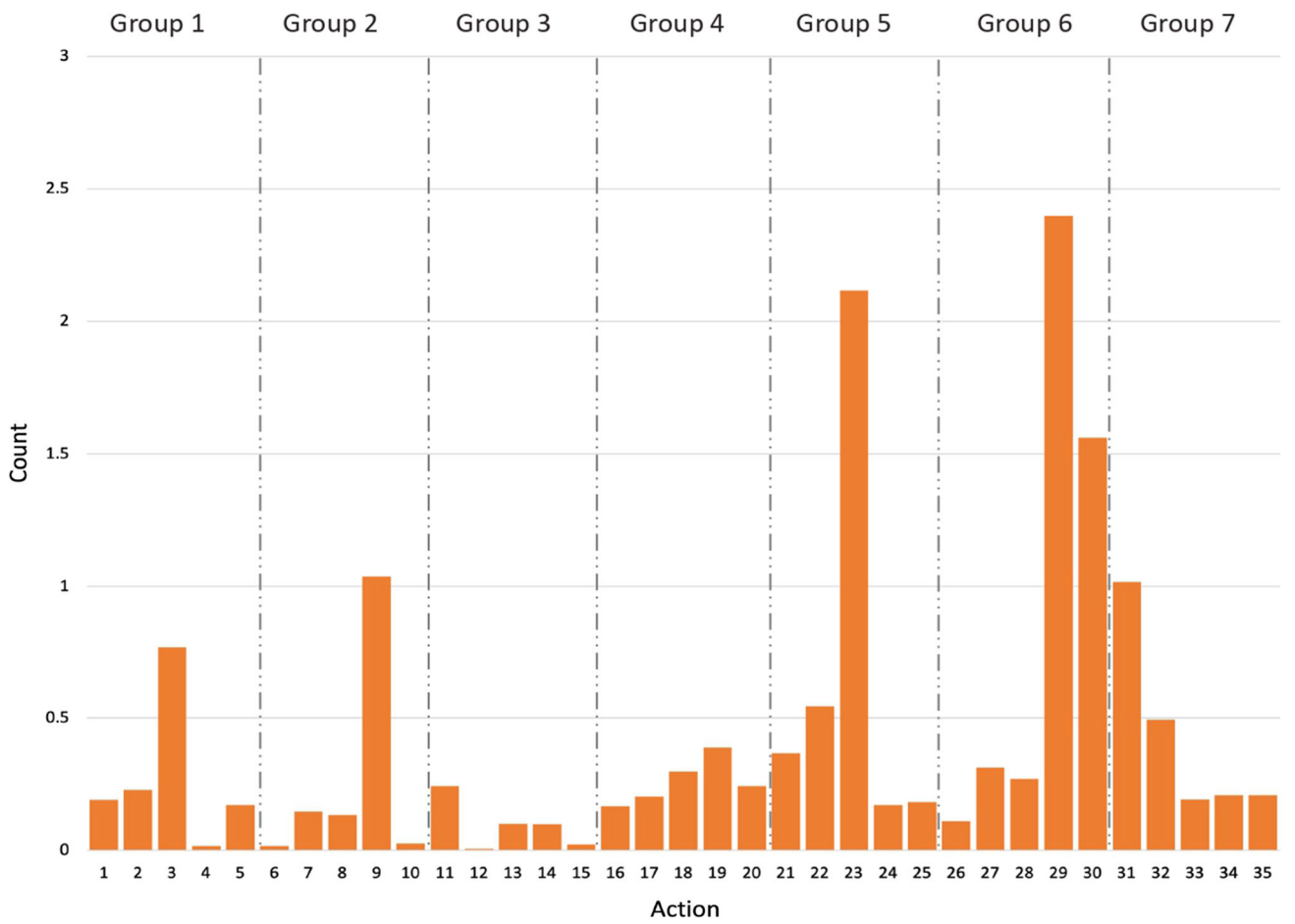

Fig. 6 Action distribution of the reinforcement learning-based model

actions and the real actions from the historical (ground-truth) data. We compare this metric amongst Sim-GAIL, the BC-based method, and the RL-based method with the original data, by using the Kullback–Leibler divergence method, which is generally used to measure the difference between two distributions [54]. Second, we compare the expected cumulative rewards (ECR) for each of these three methods. Third, we use two off-line policy evaluation (OPE) methods, including Importance Sampling (IS) and Fitted Q Evaluation (FQE), to compare the policy of these three methods. Our comprehensive and nuanced evaluation framework is aimed at delivering a more detailed and informative assessment of Sim-GAIL and its performance relative to traditional models.

In the second part of the evaluation, we use three state-of-the-art Knowledge Tracing models to evaluate Sim-GAIL, to test whether our method could be efficaciously applied in a real-world cold-start scenario. We apply the generated data to a widely used ITS technique called knowledge tracing (KT) to verify the effectiveness of our model. KT could be used to predict the students' next actions, based on their historical behavioural trajectories [6]. We apply the generated data in three state-of-the-art KT models, i.e., SSAKT, SAINT, and LTMTL, to test if the generated data mixed with the original data could improve their accuracy, when training on only a small set of student data.

5.1 Action distribution evaluation

As mentioned in Sect. 3.2, we obtain the action distribution of EdNet by allocating the 35 actions into seven groups, resulting in five actions per group, as shown in Fig. 3. We can observe that actions 21, 22, 23, and 24 have higher frequencies than other actions. This pattern also appears in the action distribution generated by Sim-GAIL (shown in Fig. 5). The major difference in action distributions between the real data from EdNet and those generated by Sim-GAIL is that action 25 (i.e., one of the lecture actions) in the latter is not close to the average value of 0. In addition, action 26 in Sim-GAIL also exhibits a higher frequency. Figure 6 shows the action distribution of the simulated students generated by the RL-based method. The highest frequencies fall into groups 5 and 6, while group 6

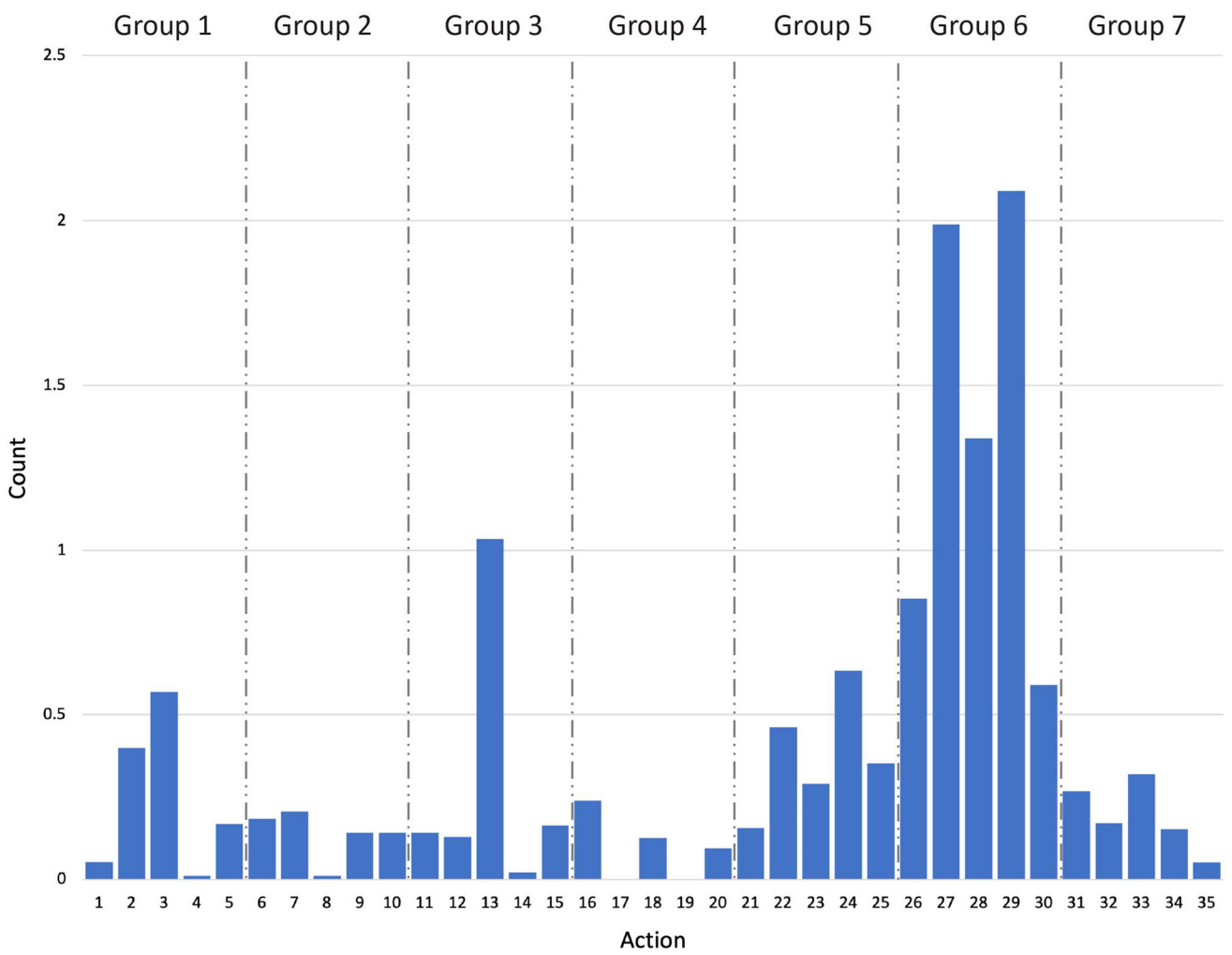

Fig. 7 Action distribution of the behavioural cloning-based model

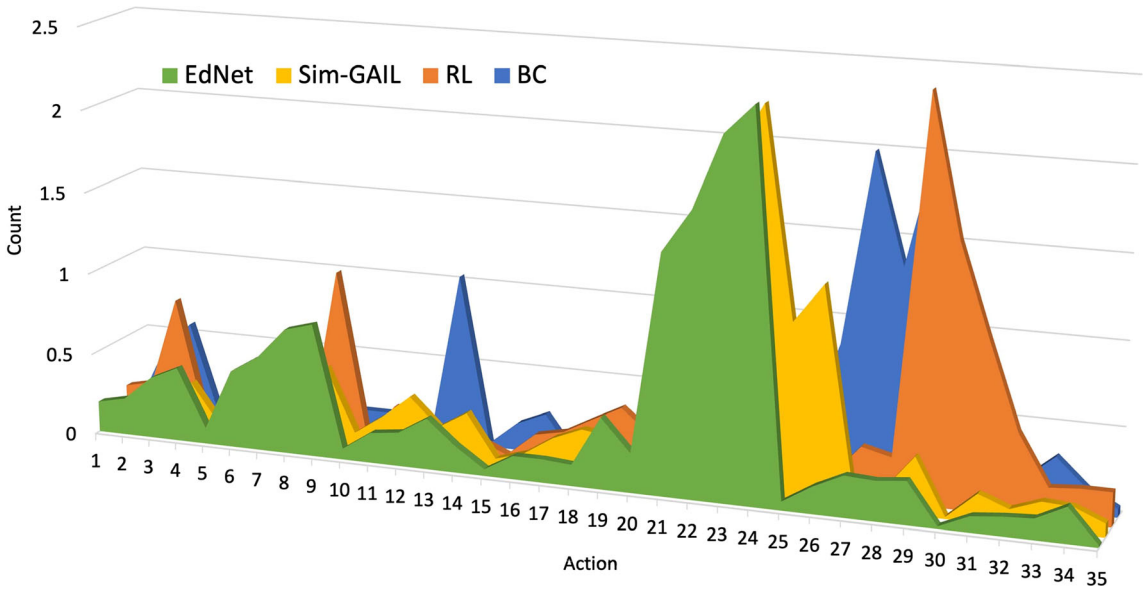

Fig. 8 Comparison of different models' actions distribution

Table 3 Kullback–Leibler divergence of action distribution

Model	Sim-GAIL	RL	BC
KL value	0.297	0.408	0.391

contains most of the high-frequency actions. Unlike the action distribution of real data, the clustering of each group can not be clearly identified in the action distribution of the RL-based method. Figure 7 shows the action distribution of the simulated students generated by the behavioural cloning (BC)-based method. Within this distribution, actions in group 6 illustrate the highest frequencies, indicating that actions in group 6 are the most frequent ones. Figure 8 compares the action distribution amongst the data generated by these three different student simulation methods. We can see that the BC-based method outperforms the RL-based method in this metric, and the action

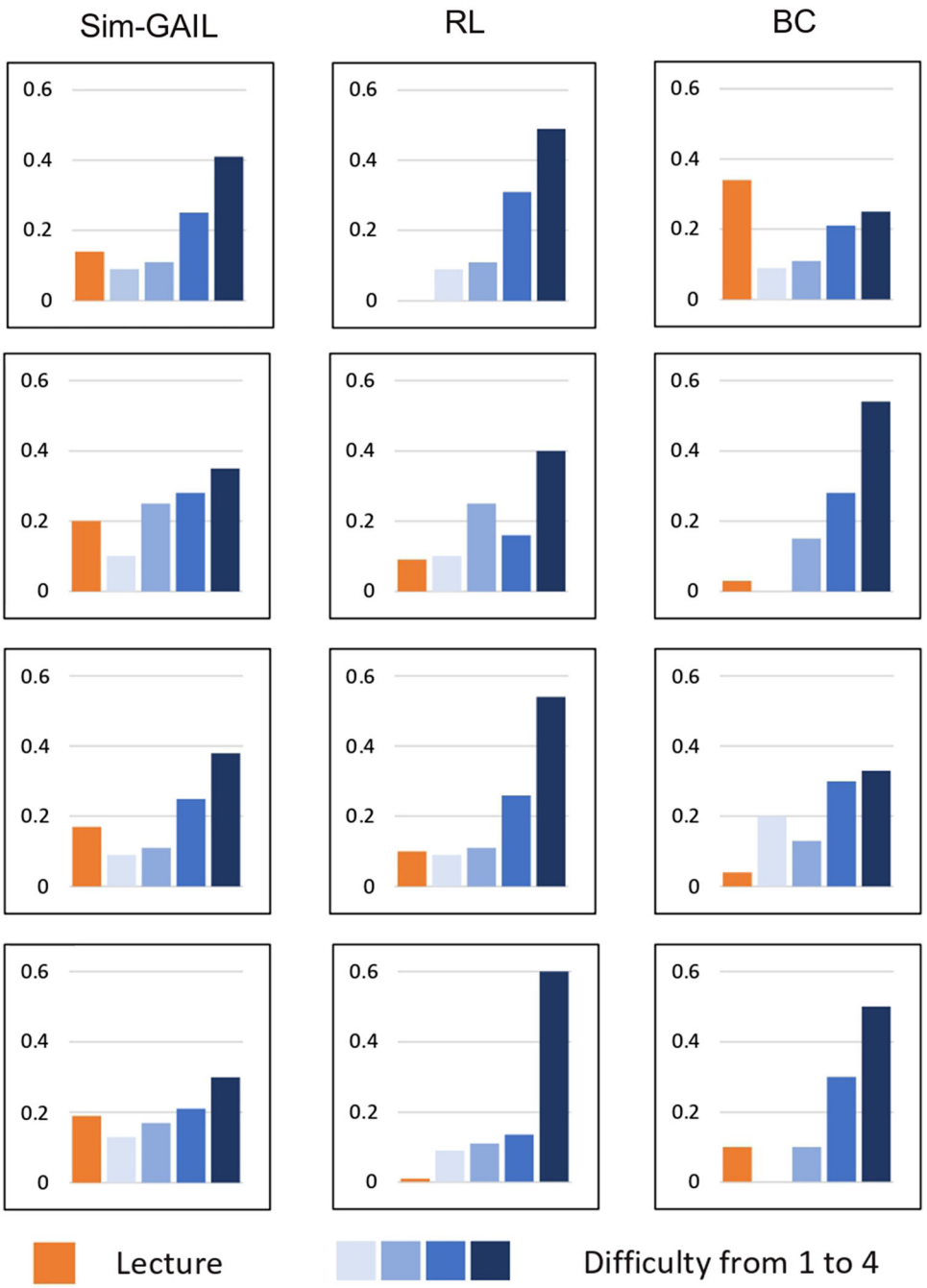

Fig. 9 Action distribution of the state feature 'topic_fam' from simulated students generated by three different methods. The horizontal axis is the value of 'topic_fam' 1–4, the vertical axis is the normalised counts of the actions, the orange bar represents the lecture consumption, and the blue bar represents questions, from easy to difficult. The difficulty is represented by hue strength

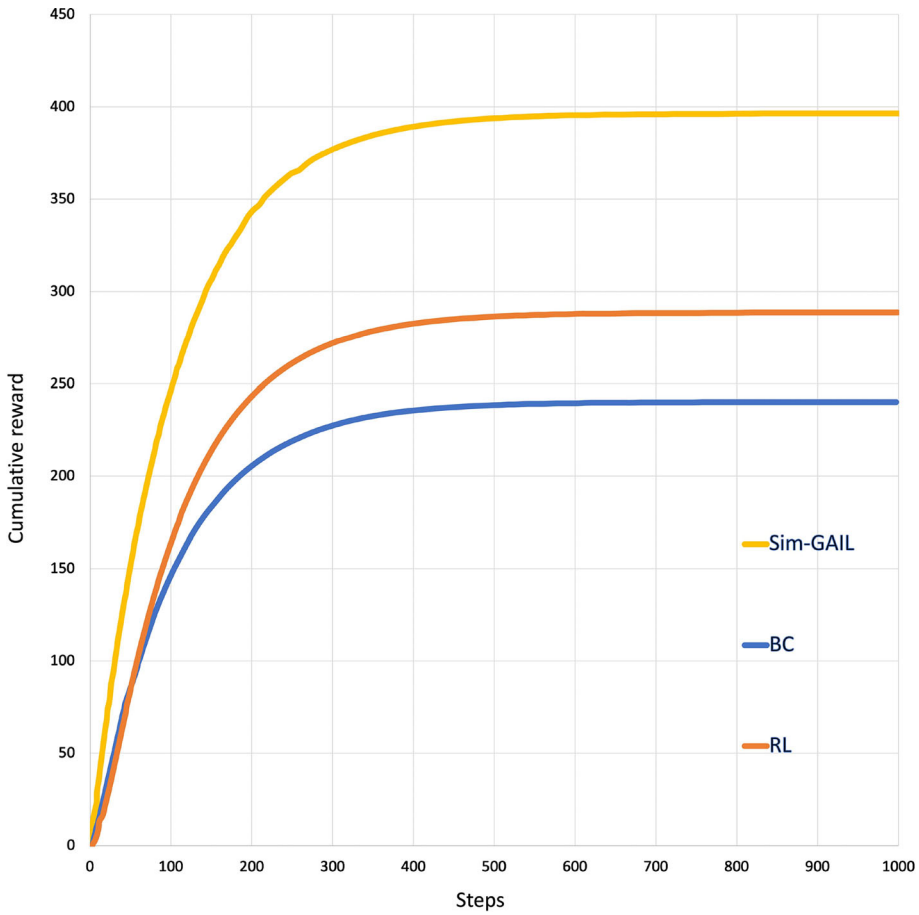

Fig. 10 Expected cumulative rewards evaluation

distribution of Sim-GAIL generated data is closest to the real data's distribution.

Moreover, we use the Kullback–Leibler divergence (KL) method to measure whether the action distribution generated by these three methods conforms to the real action distribution from EdNet. Table 3 shows the KL values of the distribution of the actions generated by these three methods and that of the real actions, respectively. The KL value between the action distribution of the data generated by Sim-GAIL; and the action distribution of the real data (ground truth) is the lowest (0.297), which suggests that the action distribution generated by Sim-GAIL is the closest to the real action distribution. Thus, it performs the best in this metric. The result also shows that the BC-based method (0.391) performs worse than Sim-GAIL but better than the RL-based method (0.408) in this metric.

The state 'topic_fam' represents a student's familiarity with the current topic. It is an important indicator that can reflect a student's mastery of knowledge. We compare the action distribution of the state value 'topic_fam' from simulated students generated by three different methods, which is shown in Fig. 9. From left to right is the distribution of simulated student actions in the state of 'topic-fam' generated by Sim-GAIL, RL-based method, and BC-based method. It can be seen that data generated by the RL-based method is the most distributed in the most difficulty-level actions (the darkest bar in each figure). Within this policy generated by RL, the method could obtain the highest rewards in the short term. However, the distribution of actions in the lecture (the orange bar) is minimal. Such a distribution does not match the real learning trajectories of students, because students need to learn new knowledge through attending lectures. The BC-based method has a more average distribution of actions on all difficulty-level actions. However, the distributions of lecture actions are unstable, which is also inconsistent with the real students' learning trajectories. The action distribution of the simulated student method based on Sim-GAIL is the most in line with the real students' trajectories action distribution, and the counts of students' actions between lectures and questions are relatively stable. This indicates that the simulated students generated by the Sim-GAIL method can balance the data distribution and optimal policy to achieve a better simulation effect.

Table 4 Importance sampling evaluation results

Model	OIS	PDIS	WIS
Behavioural cloning	6.59E+01	3.96E+01	0.970
Reinforcement learning	3.86E−02	3.25E+05	3.841
Sim-GAIL	7.35E−02	8.07E+03	4.753

5.2 Expected cumulative rewards evaluation

Expected cumulative rewards (ECR) represents the average of the expected cumulative rewards under a given policy [55]. ECR could effectively reflect the cumulative reward obtained by the method, which is a crucial indicator of the effect of the method. The equation for computing ECR is:

$$\mathrm{ECR} = \mathbb{E}_{s_0 \sim \mathcal{D}, \pi^*} \mathrm{Q}(\mathrm{s}_0, \pi^*(\mathrm{s}_0)), \tag{4}$$

where the $Q(s_0, a)$ function is the 'action value' of the action a selected by policy π in the initial state s_0. In this experimental setting, we set ECR to be simply equal to the unique initial state value $\mathrm{ECR} = \mathrm{V}_{\pi^*}(\mathrm{s}_0)$. We calculate the cumulative rewards for 100 rounds over 1000 steps starting from the initial state. The results of the expected cumulative rewards evaluation are shown in Fig. 10, and a higher ECR indicates better performance.

The ECR of Sim-GAIL grows the fastest among the three methods, suggesting its superior ability to accumulate rewards in the early stages of the simulation. This rapid growth could be attributed to the generative nature of the GAIL algorithm, which enables efficient exploration and exploitation of the simulation environment, leading to higher rewards. After 200 steps, Sim-GAIL's ECR reaches a plateau at around 400, indicating that the model has learned an optimal policy and further exploration does not significantly increase the total rewards. This illustrates the model's ability to converge to an optimal solution quickly, a key advantage in scenarios where computational resources or time are limited.

The RL method exhibits a slower ECR growth rate compared to Sim-GAIL. This could be due to the inherent challenge in reinforcement learning of balancing exploration and exploitation. Although RL eventually stabilises at a cumulative reward of approximately 290 after 500 steps, this indicates its lower efficiency compared to Sim-GAIL. BC displays the slowest ECR growth rate, stabilising at around 240 after 400 steps. This slower growth and lower final ECR compared to Sim-GAIL and RL reflect the limitations of the BC method, which may not fully capture the complex dynamics of the simulation environment.

These observations indicate that Sim-GAIL outperforms the traditional RL and BC methods in terms of ECR growth rate and final ECR value, highlighting the effectiveness of the GAIL approach in this context. This superior performance underscores the novelty and potential of our proposed Sim-GAIL as a powerful tool for generating simulated student data for ITS training.

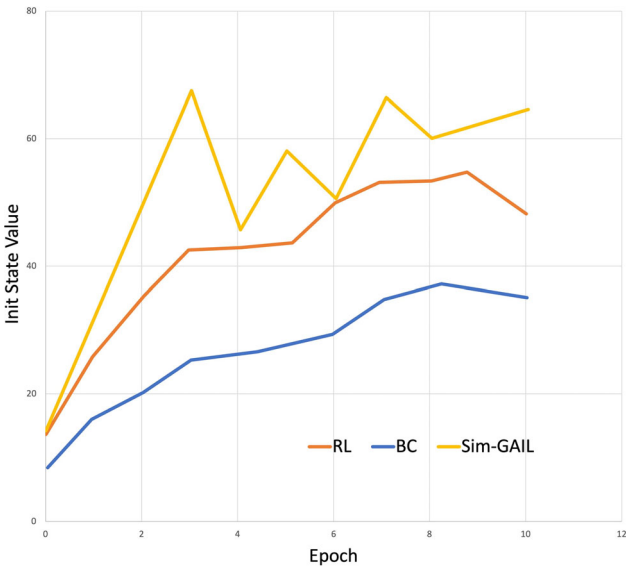

Fig. 11 Initial state value estimate of the FQE

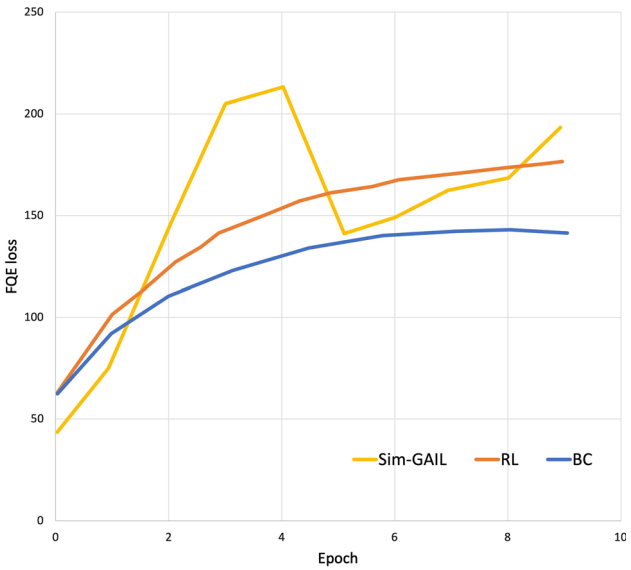

Fig. 12 The FQE-loss

5.3 Offline policy evaluation

As a robust policy evaluation method that does not require human participation, the offline policy evaluation (OPE) is often used to evaluate reinforcement learning (RL), which has shown great potential in decision-making tasks, such as robotics [56] and games [57]. In these tasks, RL optimal

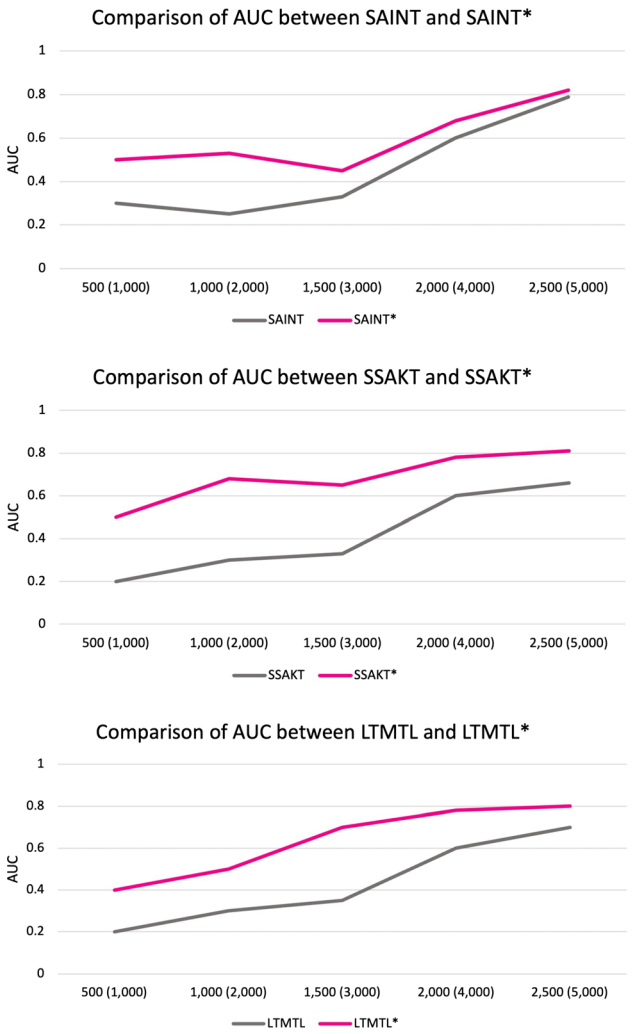

Fig. 13 Pairwise AUC comparisons of the three KT models trained on only original students' data (SAINT, SSAKT, LTMTL, in grey) and trained on the mixed dataset (SAINT*, SSAKT*, LTMTL*, in red). On the horizontal axis, 500, 1000,...,2500 indicate that the grey curve model uses the original dataset, and (1000),(2000),...,(5000) indicate that the red curve model uses the mixed dataset

strategies could be evaluated in either the environment or the simulator. There are various ways of evaluation, such as maximum cumulative reward, optimal policy, and evaluating the score in games, and the score could be high or low, and a high score indicates a better performance [58]. However, in human-participating tasks, evaluation becomes very difficult. First, human subjectivity may lead to bias in the results. Second, the simulator cannot consider every feature in a complex environment. Finally, experiments, where humans are involved, may make the evaluation process expensive, time-consuming, and resource-intensive. The OPE methods [59] were proposed to address these problems, where the evaluation of the policy is only based on the collected historical offline log data. They are mainly applied in scenarios where online interactions involve high-risk and expensive settings, such as stock trading, medical recommendation, and educational systems [60]. In this paper, we employed a combination of two OPE methods: the Importance Sampling (including three variants, OIS, WIS, and PIS) [61] and the Fitted Q Evaluation method [62], which allows for testing the policy performance of the three models.

5.3.1 Importance sampling

As one of the OPE methods, importance sampling (IS) is used in situations where it is difficult to sample directly from the original data distribution. It is a method that uses a simple and collectable distribution to calculate the expected value of the desired distribution [61]. There are many works using IS to evaluate the target policy (the policy derived from the RL algorithms) and the behaviour policy (the policy used to gather the data) when dealing with MDPs [63, 64]. However, the basic IS method may suffer from high variance, due to the huge difference between those two policies. In our experiment, we used three IS methods: the general IS (i.e., ordinary importance sampling (OIS)) and two variants of the general IS, including weighted importance sampling (WIS) and per-decision importance sampling (PDIS). WIS employs a weighted average to mitigate the variance [65]. The Per-Decision Importance Sampling modifies the sampling ratio and makes the reward dependent only upon the previous action in each timestamp [62]. The combination of the three methods can better observe the policy distribution of the generated data.

Table 4 shows the results of the Importance Sampling evaluation. On the OIS criteria, the BC-based method outperforms the RL-based method but is worsen than Sim-GAIL. On the PDIS criteria, the Sim-GAIL method outperforms both RL-based and BC-based methods and the BC-based method performs better than the RL-based method. Sim-GAIL outperforms the other two baseline models, and the RL-based method performs better than the BC-based method on the WIS criteria. In summary, Sim-GAIL outperforms the other baseline models on every criterion.

5.3.2 Fitted Q evaluation

The FQE algorithm regards the MDP as a supervised learning problem. This method uses a function approximator to fit the Q function under a specified policy, based on the observation of the dataset [62].

Figure 11 shows the Fitted Q Evaluation results on the initial state. Sim-GAIL outshines the other two methods, affirming its superior performance. This is likely due to the strengths of the GAIL approach, which efficiently captures

the complex dynamics of the environment and generates more robust policies. Sim-GAIL's ISV peaks in the third epoch, indicating rapid learning and optimisation. Despite subsequent oscillations, Sim-GAIL's performance consistently surpasses that of RL and BC methods, showcasing its robustness and stability. The RL method exhibits better ISV performance than the BC method. Both methods show a steady increase, with their maximum ISV reached in the 9th epoch. However, their peak performance still falls short of Sim-GAIL's average level, underscoring the superior efficiency and effectiveness of Sim-GAIL. In summary, Fig. 11 highlights the efficacy of Sim-GAIL in terms of policy quality and learning speed, as evidenced by its superior Fitted Q Evaluation results. This underscores the potential of Sim-GAIL as an efficient and robust approach for generating simulated student data for ITS training.

Figure 12 shows the FQE loss of the three methods. Sim-GAIL's FQE loss increases rapidly, peaking in the third and fourth epochs. It then swiftly declines but starts to ascend again after the fifth epoch. This rapid fluctuation reflects the model's active learning and adaptation process. In contrast, RL and BC methods exhibit relatively stable, slower FQE loss growth. In particular, RL shows moderate growth, while BC displays the slowest growth. This slower and more stable growth could be indicative of a more conservative learning process compared to Sim-GAIL.

Despite generating the highest $Q(s0, \pi(s0))$ values, Sim-GAIL also incurs higher and less stable validation loss compared to the RL and BC methods. This suggests that while Sim-GAIL is efficient in learning and optimising the policy, it may overfit the training data, leading to higher validation loss. While Sim-GAIL outperforms RL and BC methods overall, the results also indicate a need for parameter tuning to reduce the loss, highlighting an area for further improvement in Sim-GAIL's implementation.

In summary, Fig. 12 underscores the dynamic and efficient learning capability of Sim-GAIL, as well as the need for further tuning to optimise its performance. Despite the higher and less stable validation loss, Sim-GAIL's overall superiority in generating higher Q-values reaffirms its potential as a robust tool for generating simulated student data for Intelligent Tutoring System training.

5.4 Evaluation using knowledge tracing (KT) models

Knowledge tracing (KT) is an emerging research direction and has been widely applied in intelligent educational applications, where students' historical trajectories are used to model and predict their knowledge states [31]. However, the lack of student interaction data in the early stage of using a system, known as the cold-start problem, limits the performance of KT models. It has been one massive obstacle to the development and application of KT. In this experiment, we applied the original data and the data generated from the Sim-GAIL method to the state-of-the-art KT models to test whether our model could improve the performance of KT models in a cold-start scenario. This in turn proves the efficiency of our proposed Sim-GAIL method's ability to simulate and generate students' historical trajectory data.

In the KT research area, there is a Riiid Answer Correctness Prediction Competition on Kaggle,[3] which compares the state-of-the-art KT models using the EdNet dataset. The current top three models in this competition are SAINT, SSAKT, and LTMTI.[4] The prediction competition provides a dataset of 2500 students to train the KT model. Therefore, we assume that the volume of 2500 students is sufficient for KT models to get good prediction performances. Thus, in our experiments, we considered the case of a data size of no more than 2500. Therefore, we selected datasets of sizes 500, 1000, 1500, 2000, and 2500 student records. Each student record contains the student's sequence of discrete learning actions. In our experiment, we first used Sim-GAIL to generate simulated data whose size is equal to the original data size, and then we mixed it with the original real data to build a new dataset. After that, we fed this mixed dataset into the 3 KT models, respectively. For example, in the case of the original data size being equal to 500, we input the 500 student records to Sim-GAIL, which generated equally-sized (i.e., 500) simulated student records. Then, we mixed these 500 generated student records with the original 500 student records, to build a new dataset of size 1000. This new mixed dataset was finally used to train the KT models. We compared the performance of the KT models between using this mixed dataset and using only the original data. The metric we used here is AUC.

Figure 13 shows the pairwise AUC comparisons of the three KT models trained on only the original students' data (SAINT, SSAKT, and LTMTL; in grey) and trained on the mixed dataset (SAINT*, SSAKT*, and LTMTL*; in red). The curves of SSAKT* and LTMTL* are constantly higher than the curves of SSAKT and LTMTL in all the cases, i.e., 1000, 2000, 3000, 4000, and 5000 sizes of the mixed dataset. The curve of SAINT* is higher than the curve of SAINT in the cases of 1000, 2000, and 3000 sizes of data. Although the curve of SAINT* is very close to SAINT in the cases of 5000 sizes of data, the former still outperforms the latter. In all those three pairwise comparisons, especially in the cases of smaller data sizes (1000, 2000, and 3000), obviously, training on mixed data (a combination of

[3] https://www.kaggle.com/code/datakite/riiid-answer-correctness.

[4] http://ednet-leaderboard.s3-website-ap-northeast-1.amazonaws.com.

the original and generated data) could improve the KT models. The graphical representation of these results would likely show an upward trend for all KT models, demonstrating that the accuracy of the KT models can be improved with more data and iterations. The lines representing the training on mixed data would be above those of the original KT models, indicating our method's superior performance. This suggests that the data generated by our Sim-GAIL method can help improve the KT models, especially in cold-start scenarios, where the size of the available data is small.

6 Discussions and future work

6.1 Result analysis

From the results of the experiment, we observe that Sim-GAIL outperforms the baseline methods on the metrics of *Action Distribution Evaluation*, *Expected Cumulative Rewards Evaluation*, and *Offline Policy Evaluation*. The satisfying fit simulation results may come from the fact that there is no need to define a reward function for Sim-GAIL, compared with other baseline models. Defining reward functions manually may be too complex to fit the real student trajectories', thus a reward function built by algorithms instead of humans might result in a better policy [20]. The results of the evaluation using the KT models show that Sim-GAIL could be applied to real-world educational scenarios and improve the efficiency of current educational technologies. More specifically, our method could effectively alleviate the cold-start problem of KT models.

Our Sim-GAIL method outperforms the baseline models on every metric. The RL-based method outperforms the BC-based method in terms of offline policy evaluation. This indicates that a suitable setting of the reward function could generate better policies. This result is also reflected in the distribution of 'topic_fam' actions. The policy generated by the RL-based method places more emphasis on high-difficulty and high-reward actions. Such a policy works well for obtaining higher cumulative rewards, but it does not match the action distribution of real students' trajectories. Besides, the distribution of 'lecture' actions whose default reward value is 0, is very small and unstable. Thus, the action distribution generated by the RL-based method is inconsistent with the action distribution of real students' trajectories. The BC-based method outperforms the RL-based method in action distribution, but is worse in offline policy evaluation. This suggests that, although the BC-based method can render the action distribution more aligned with the real action distribution, it is difficult to obtain a better learning policy. Therefore, Sim-GAIL is a more advanced student simulation method than those two traditional ones. Besides, as Sim-GAIL does not require a dedicated reward function to fit different datasets, compared with traditional student simulation methods, our method could be easily transferred and applied to another ITS.

In the evaluation using KT models, we apply our method to three different state-of-the-art KT models. The results indicate that our method could improve training efficiency in cold-start scenarios. In Fig. 13, every KT model trained on the mixed data (a combination of the original data and the data generated by our Sim-GAIL method) performs better in each group. The results suggest that it could improve training efficiency in small-sized data scenarios, proving that it could alleviate the cold-start problem in the early stages of ITS development. For instance, in the above experiments, every KT* model performs better when the original data size is smaller than 2000. After the data size is larger than 2000, the performance of using the original dataset (KT) is close to that of using a mixed dataset (KT*), but the KT* still outperforms the KT.

6.2 Advantages

Our proposed model, Sim-GAIL, brings several significant advantages to the field of student modelling for intelligent tutoring systems (ITS). A fundamental strength of Sim-GAIL lies in its underlying mechanism, that of generative adversarial imitation learning (GAIL), which endows the model with the capacity to generate new data instances that closely resemble actual student behaviour data. This generative modelling capability of Sim-GAIL is crucial for creating a rich, diverse dataset needed for effective ITS training. Additionally, Sim-GAIL offers a solution to a common issue encountered in the early stages of ITS development - the cold-start problem. The ability to generate simulated student data allows Sim-GAIL to effectively tackle this problem, accelerating the training process of ITS.

In terms of performance, Sim-GAIL has demonstrated superiority over traditional reinforcement learning (RL) and behavioural cloning (BC) based methods across various metrics, including action distribution evaluation, cumulative reward evaluation, and offline-policy evaluation. This implies that Sim-GAIL can simulate student behaviours with higher accuracy and effectiveness. Furthermore, the efficiency of Sim-GAIL is evident from the rapid convergence to an optimal policy whilst simulating real student learning trajectories, providing a significant advantage in scenarios where computational resources or time are limited.

Beyond these, the scalability and generality of Sim-GAIL further enhance its appeal. As a data-driven model, Sim-GAIL does not rely on expert knowledge for defining the reward function, which contrasts with some RL-based methods. This attribute allows Sim-GAIL to scale and generalise across different datasets and applications, seamlessly.

In essence, Sim-GAIL represents a novel, effective, and efficient approach to student modelling. By offering a promising tool for generating simulated student data, Sim-GAIL contributes to enhancing the efficacy of ITS training.

6.3 Limitations

The limitations of this work mainly lie in the following aspects. First, our work adopts a general state representation method from other studies [4, 51], where Sim-GAIL outperforms other baseline methods on most metrics. As discussed in Sect. 3.2, the selection of state representation may impact the models' performance. However, the experimental design of our work does not consider the potential impact of different state combinations on various methods. Second, in the experiments of evaluation using KT models, when a KT model moves beyond the cold-start stage and has sufficient data, the increase in the amount of simulated data may lead to a decrease in the prediction accuracy of the KT model, which may be the bias caused by Sim-GAIL not considering all the features of student actions.

6.4 Future work

While our proposed Sim-GAIL method shows promising results in student simulation for Intelligent Tutoring Systems (ITS), there are several avenues for future exploration and improvement.

Fine-grained simulations In our current implementation, Sim-GAIL focuses on generating simulated student behaviour data at a coarse level. Future work can explore methods to capture more fine-grained details, such as students' cognitive processes, metacognitive strategies, and affective states. Incorporating these aspects could lead to more accurate and comprehensive student modelling.

Adaptive simulation Currently, Sim-GAIL generates simulated student data based on predefined models. Future research can investigate methods to make the simulation adaptive, allowing sim students to learn and evolve based on feedback from the ITS. This adaptive simulation approach can provide more dynamic and personalised student trajectories.

Transfer learning and generalisation Sim-GAIL has been evaluated on the EdNet dataset, but its generalisability to other domains and datasets remains an open question. Future work can explore transfer learning techniques to enhance the model's ability to generalise across different educational contexts and datasets, enabling wider applicability of Sim-GAIL in various ITS settings.

Human-in-the-loop simulations Although Sim-GAIL offers an efficient alternative to collecting real student data, it is crucial to acknowledge the limitations of fully replacing human students with sim-students. Future research can investigate human-in-the-loop simulation methods, where sim students are combined with real student interactive data, allowing for iterative refinement and validation of the simulated trajectories.

By pursuing these future research directions, we can further enhance Sim-GAIL's capabilities and contribute to the advancement of student modelling techniques in the field of Intelligent Tutoring Systems.

7 Conclusion

In this study, we have introduced Sim-GAIL, a pioneering student simulation method founded on the generative adversarial imitation learning (GAIL) algorithm. It stands as the first of its kind that trains ITS using simulated student behaviour data, effectively addressing the challenges of high-cost, resource-intensive real student data collection, and the cold-start problem encountered during early-stage ITS training. Sim-GAIL demonstrates superior performance in comparison with traditional Reinforcement Learning-based and Imitation Learning-based methods, marking a significant advancement in state-of-the-art student modelling for Intelligent Tutoring Systems.

Our student simulation method, Sim-GAIL, leverages the EdNet dataset and outperforms the baseline methods: a Reinforcement Learning method based on Conservative Q-learning and an Imitation Learning method based on Behavioural Cloning. We have thoroughly evaluated our method from four aspects: action distribution discrepancy based on the Kullback–Leibler divergence, reward function using expected cumulative rewards (ECR), and two offline policy evaluation (OPE) methods—Importance Sampling and Fitted Q Evaluation. Our results convincingly demonstrate that Sim-GAIL outperforms the baseline models in all these aspects.

Further, we have applied Sim-GAIL to state-of-the-art knowledge tracing models and observed a noticeable improvement in their performance, especially in cold-start scenarios. This underlines Sim-GAIL's efficiency in simulating and generating students' historical trajectory data, further emphasising its novelty and potential to contribute to the field of student modelling for Intelligent Tutoring Systems.

Moving forward, research can explore fine-grained simulations, adaptive simulation techniques, transfer learning and generalisation, and human-in-the-loop simulations, to enhance Sim-GAIL's capabilities in student modelling even further, as discussed in Sect. 6. This study paves the way for these future endeavours by providing a robust, novel method for generating simulated student data for ITS training.

Data availability The datasets analysed during the current study are available in the EdNet repository http://doi.org/10.48550/arXiv.1912.03072 [12]. These datasets were derived from the following public domain resources: http://github.com/riiid/ednet#properties-of-ednet.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest in this work.

References

1. Zhu X (2015) Machine teaching: an inverse problem to machine learning and an approach toward optimal education. In: Proceedings of the AAAI conference on artificial intelligence, vol 29
2. Ritter FE, Nerb J, Lehtinen E, O'Shea TM (2007) In order to learn: how the sequence of topics influences learning. Oxford University Press, Oxford
3. Shi L, Cristea AI, Awan MSK, Hendrix M, Stewart C (2013) Towards understanding learning behavior patterns in social adaptive personalized e-learning systems. Assoc Inf Syst
4. Bassen J, Balaji B, Schaarschmidt M, Thille C, Painter J, Zimmaro D, Games A, Fast E, Mitchell JC (2020) Reinforcement learning for the adaptive scheduling of educational activities. In: Proceedings of the 2020 CHI conference on human factors in computing systems, pp 1–12
5. Stash NV, Cristea AI, De Bra PM (2004) Authoring of learning styles in adaptive hypermedia: problems and solutions. In: Proceedings of the 13th international world wide web conference on alternate track papers & posters. ACM, New York, pp 114–123. https://doi.org/10.1145/1013367.1013387
6. Li Z, Shi L, Cristea A, Zhou Y, Xiao C, Pan Z (2022) Simstu-transformer: a transformer-based approach to simulating student behaviour. In: International conference on artificial intelligence in education. Springer, Berlin, pp 348–351
7. Cristea AI, Okamoto T (2001) Considering automatic educational validation of computerized educational systems. In: Proceedings IEEE international conference on advanced learning technologies. IEEE, Madison, pp 415–417. https://doi.org/10.1109/ICALT.2001.943962
8. Jarboui F, Gruson-Daniel C, Durmus A, Rocchisani V, Goulet Ebongue S-H, Depoux A, Kirschenmann W, Perchet V (2019) Markov decision process for MOOC users behavioral inference. In: European MOOCs stakeholders summit. Springer, Berlin, pp 70–80
9. Zimmer M, Viappiani P, Weng P (2014) Teacher-student framework: a reinforcement learning approach. In: AAMAS Workshop autonomous robots and multirobot systems
10. Anderson CW, Draper BA, Peterson DA (2000) Behavioral cloning of student pilots with modular neural networks. In: ICML, pp 25–32
11. Schaal S (1999) Is imitation learning the route to humanoid robots? Trends Cogn Sci 3(6):233–242
12. Choi Y, Lee Y, Shin D, Cho J, Park S, Lee S, Baek J, Bae C, Kim B, Heo J (2020) Ednet: a large-scale hierarchical dataset in education. In: International conference on artificial intelligence in education. Springer, Berlin, pp 69–73
13. Shen S, Chi M (2016) Reinforcement learning: the sooner the better, or the later the better? In: Proceedings of the 2016 conference on user modeling adaptation and personalization, pp 37–44
14. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT press, Cambridge
15. Levin E, Pieraccini R, Eckert W (1998) Using Markov decision process for learning dialogue strategies. In: Proceedings of the 1998 IEEE international conference on acoustics, speech and signal processing, ICASSP'98 (Cat. No. 98CH36181), vol 1. IEEE, pp 201–204
16. Li Z, Shi L, Cristea AI, Zhou Y (2021) A survey of collaborative reinforcement learning: interactive methods and design patterns. In: Designing interactive systems conference 2021, pp 1579–1590
17. Hussein A, Gaber MM, Elyan E, Jayne C (2017) Imitation learning: a survey of learning methods. ACM Comput Surv (CSUR) 50(2):1–35
18. Pomerleau DA (1988) Alvinn: an autonomous land vehicle in a neural network. In: Advances in neural information processing systems, vol 1
19. Pomerleau DA (1991) Efficient training of artificial neural networks for autonomous navigation. Neural Comput 3(1):88–97
20. Ho J, Ermon S (2016) Generative adversarial imitation learning. In: Advances in neural information processing systems, vol 29
21. Bhattacharyya R, Wulfe B, Phillips D, Kuefler A, Morton J, Senanayake R, Kochenderfer M (2020) Modeling human driving behavior through generative adversarial imitation learning. arXiv preprint arXiv:2006.06412
22. Ross S, Bagnell D (2010) Efficient reductions for imitation learning. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR workshop and conference proceedings, pp 661–668
23. Ross S, Gordon G, Bagnell D (2011) A reduction of imitation learning and structured prediction to no-regret online learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR workshop and conference proceedings, pp 627–635
24. Abbeel P, Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the twenty-first international conference on machine learning, p 1
25. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial

nets. In: Advances in neural information processing systems, vol 27
26. Ng AY, Russell SJ et al (2000) Algorithms for inverse reinforcement learning. In: ICML, vol 1, p 2
27. Brusilovsky P (2012) Adaptive hypermedia for education and training. In: Adaptive technologies for training and education, vol 46, pp 46–68
28. Shi L, Al Qudah D, Qaffas A, Cristea AI (2013) Topolor: A social personalized adaptive e-learning system. In: Carberry S, Weibelzahl S, Micarelli A, Semeraro G (eds) User modeling, adaptation, and personalization. Springer, Berlin, pp 338–340
29. Shi L, Cristea AI (2016) Learners thrive using multifaceted open social learner modeling. IEEE Multimed 23(1):36–47. https://doi.org/10.1109/MMUL.2015.93
30. Shi L, Cristea AI, Toda AM, Oliveira W (2020) Exploring navigation styles in a futurelearn MOOC. In: Kumar V, Troussas C (eds) Intelligent tutoring systems. Springer, Cham, pp 45–55
31. Liu Q, Shen S, Huang Z, Chen E, Zheng Y (2021) A survey of knowledge tracing. arXiv preprint arXiv:2105.15106
32. Alharbi K, Cristea AI, Okamoto T (2021) Agent-based classroom environment simulation: the effect of disruptive schoolchildren's behaviour versus teacher control over neighbours. In: Artificial intelligence in education. AIED 2021. Lecture notes in computer science. Springer, Cham. https://doi.org/10.1007/978-3-030-78270-2_8
33. Li Z, Shi L, Zhou Y, Wang J (2023) Towards student behaviour simulation: a decision transformer based approach. In: International conference on intelligent tutoring systems. Springer, Berlin, pp 553–562
34. Doroudi S, Aleven V, Brunskill E (2019) Where's the reward? Int J Artif Intell Educ 29(4):568–620
35. Iglesias A, Martínez P, Aler R, Fernández F (2009) Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems. Knowl Based Syst 22(4):266–270
36. Yudelson MV, Koedinger KR, Gordon GJ (2013) Individualized Bayesian knowledge tracing models. In: International conference on artificial intelligence in education. Springer, Berlin, pp 171–180
37. Hambleton RK, Swaminathan H, Rogers HJ (1991) Fundamentals of item response theory, vol 2. Sage, Newbury Park, London, New Delhi
38. Segal A, David YB, Williams JJ, Gal K, Shalom Y (2018) Combining difficulty ranking with multi-armed bandits to sequence educational content. In: International conference on artificial intelligence in education. Springer, Berlin, pp 317–321
39. Azhar AZ, Segal A, Gal K (2022) Optimizing representations and policies for question sequencing using reinforcement learning. Int Educ Data Min Soc
40. Tetreault JR, Litman DJ (2008) A reinforcement learning approach to evaluating state representations in spoken dialogue systems. Speech Commun 50(8–9):683–696
41. Rowe J, Pokorny B, Goldberg B, Mott B, Lester J (2017) Toward simulated students for reinforcement learning-driven tutorial planning in gift. In: Proceedings of R. Sottilare (Ed.) 5th annual GIFT users symposium. Orlando, FL
42. Chi M, VanLehn K, Litman D (2010) Do micro-level tutorial decisions matter: applying reinforcement learning to induce pedagogical tutorial tactics. In: International conference on intelligent tutoring systems. Springer, Berlin, pp 224–234
43. Beck J, Woolf BP, Beal CR (2000) Advisor: a machine learning architecture for intelligent tutor construction. AAAI/IAAI 2000(552–557):1–2
44. Emond B, Smith J, Musharraf M, Torbati RZ, Billard R, Barnes J, Veitch B (2022) Development of AIS using simulated learners, bayesian networks and knowledge elicitation methods. In: International conference on human-computer interaction. Springer, Berlin, pp 143–158
45. Shen S, Chi M (2016) Aim low: correlation-based feature selection for model-based reinforcement learning. Int Educ Data Min Soc
46. Ho J, Gupta J, Ermon S (2016) Model-free imitation learning with policy optimization. In: International conference on machine learning. PMLR, pp 2760–2769
47. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
48. Torabi F, Warnell G, Stone P (2018) Behavioral cloning from observation. arXiv preprint arXiv:1805.01954
49. Kumar A, Zhou A, Tucker G, Levine S (2020) Conservative q-learning for offline reinforcement learning. Adv Neural Inf Process Syst 33:1179–1191
50. Lefèvre S, Sun C, Bajcsy R, Laugier C (2014) Comparison of parametric and non-parametric approaches for vehicle speed prediction. In: 2014 American control conference. IEEE, pp 3494–3499
51. Azhar ZAZ (2021) Designing an offline reinforcement learning based pedagogical agent with a large scale educational dataset. Master of Science Thesis, Data Science. University of Edinburgh
52. Busoniu L, Babuska R, De Schutter B, Ernst D (2010) Reinforcement learning and dynamic programming using function approximators. CRC press, Subs. of Times Mirror 2000 Corporate Blvd. NW Boca Raton, FLUnited States
53. Bellemare MG, Dabney W, Munos R (2017) A distributional perspective on reinforcement learning. In: International conference on machine learning. PMLR, pp 449–458
54. Hershey JR, Olsen PA (2007) Approximating the Kullback Leibler divergence between gaussian mixture models. In: 2007 IEEE International conference on acoustics, speech and signal processing-ICASSP'07, vol 4. IEEE, p 317
55. Voloshin C, Le HM, Jiang N, Yue Y (2019) Empirical study of off-policy policy evaluation for reinforcement learning. arXiv preprint arXiv:1911.06854
56. Johannink T, Bahl S, Nair A, Luo J, Kumar A, Loskyll M, Ojea JA, Solowjow E, Levine S (2019) Residual reinforcement learning for robot control. In: 2019 International conference on robotics and automation (ICRA). IEEE, pp 6023–6029
57. Lapan M (2018) Deep reinforcement learning hands-on: apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo zero and more. Packt Publishing, Ltd. https://doi.org/10.5555/3279266
58. Weaver L, Tao N (2013) The optimal reward baseline for gradient-based reinforcement learning. arXiv preprint arXiv:1301.2315
59. Mandel T, Liu Y-E, Levine S, Brunskill E, Popovic Z (2014) Offline policy evaluation across representations with applications to educational games. In: AAMAS, vol 1077
60. Saito Y, Udagawa T, Kiyohara H, Mogi K, Narita Y, Tateno K (2021) Evaluating the robustness of off-policy evaluation. In: Fifteenth ACM conference on recommender systems, pp 114–123
61. Tokdar ST, Kass RE (2010) Importance sampling: a review. Wiley Interdiscip Rev Comput Stat 2(1):54–60
62. Tirinzoni A, Salvini M, Restelli M (2019) Transfer of samples in policy search via multiple importance sampling. In: International conference on machine learning. PMLR, pp 6264–6274
63. Shelton CR (2001) Importance sampling for reinforcement learning with multiple objectives
64. Ju S, Shen S, Azizsoltani H, Barnes T, Chi M (2019) Importance sampling to identify empirically valid policies and their critical decisions. In: EDM (Workshops), pp 69–78
65. Mahmood AR, Van Hasselt HP, Sutton RS (2014) Weighted importance sampling for off-policy learning with linear function approximation. In: Advances in neural information processing systems, vol 27