Deep Reinforcement Learning Approaches for Technology Enhanced Learning

ZHAOXING LI

Deep Reinforcement Learning Approaches for Technology Enhanced Learning

Zhaoxing Li

A Thesis presented for the degree of
Doctor of Philosophy

Department of Computer Science
Durham University
United Kingdom
June 2023

Abstract

Artificial Intelligence (AI) has advanced significantly in recent years, transforming various industries and domains. Its ability to extract patterns and insights from large volumes of data has revolutionised areas such as image recognition, natural language processing, and autonomous systems. As AI systems become increasingly integrated into daily human life, there is a growing need for meaningful collaboration and mutual engagement between humans and AI, known as Human-AI Collaboration. This collaboration involves combining AI with human workflows to achieve shared objectives.

In the current educational landscape, the integration of AI methods in Technology Enhanced Learning (TEL) has become crucial for providing high-quality education and facilitating lifelong learning. Human-AI Collaboration also plays a vital role in the field of Technology Enhanced Learning (TEL), particularly in Intelligent Tutoring Systems (ITS). The COVID-19 pandemic has further emphasised the need for effective educational technologies to support remote learning and bridge the gap between traditional classrooms and online platforms. To maximise the performance of ITS while minimising the input and interaction required from students, it is essential to design collaborative systems that effectively leverage the capabilities of AI and foster effective collaboration between students and ITS.

However, there are several challenges that need to be addressed in this context. One challenge is the lack of clear guidance on designing and building user-friendly systems that facilitate collaboration between humans and AI. This challenge is relevant not only to education researchers but also to Human-Computer Interaction (HCI) researchers and developers. Another challenge is the scarcity of interaction data in the early stages of ITS development, which hampers the accurate modelling of students' knowledge states and learning trajectories, known as the cold start problem. Moreover, the effectiveness of Intelligent Tutoring Systems (ITS) in delivering personalised instruction is hindered by the limitations of existing Knowledge Tracing (KT) models, which often struggle to provide accurate predictions. Therefore, addressing these challenges is crucial for enhancing the collaborative process between

humans and AI in the development of ITS.

This thesis aims to address these challenges and improve the collaborative process between students and ITS in TEL. It proposes innovative approaches to generate simulated student behavioural data and enhance the performance of KT models. The thesis starts with a comprehensive survey of human-AI collaborative systems, identifying key challenges and opportunities. It then presents a structured framework for the student-ITS collaborative process, providing insights into designing user-friendly and efficient systems.

To overcome the challenge of data scarcity in ITS development, the thesis proposes two student modelling approaches: Sim-GAIL and SimStu. SimStu leverages a deep learning method, the Decision Transformer, to simulate student interactions and enhance ITS training. Sim-GAIL utilises a reinforcement learning method, Generative Adversarial Imitation Learning (GAIL), to generate high-fidelity and diverse simulated student behavioural data, addressing the cold start problem in ITS training.

Furthermore, the thesis focuses on improving the performance of KT models. It introduces the MLFBKT model, which integrates multiple features and mines latent relations in student interaction data, aiming to improve the accuracy and efficiency of KT models. Additionally, the thesis proposes the LBKT model, which combines the strengths of the BERT model and LSTM to process long sequence data in KT models effectively.

Overall, this thesis contributes to the field of Human-AI collaboration in TEL by addressing key challenges and proposing innovative approaches to enhance ITS training and KT model performance. The findings have the potential to improve the learning experiences and outcomes of students in educational settings.

Keywords: Human-AI Collaboration, Technology Enhanced Learning, Intelligent Tutoring Systems, Knowledge Tracing, Decision Transformer, Generative Adversarial Imitation Learning, MLFBKT, LBKT.

Declaration

The work in this thesis is based on research carried out at the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Note on Publications Included in This Thesis: At the time of submission, five chapters of this thesis are heavily based on papers submitted for publication or published in conferences and journals:

Chapter 3 Li, Z., Shi, L., Cristea, A. I., Zhou, Y. (2021, June). ***A survey of collaborative reinforcement learning: interactive methods and design patterns.*** In Designing Interactive Systems Conference 2021 (pp. 1579-1590).

Li, Z., Shi, L., Cristea, A. I., Zhou, Y. (2021, June). ***A Design Trajectory Map of Human-AI Collaborative Reinforcement Learning Systems: Survey and Taxonomy*** ACM Transactions on Computer-Human Interaction. ***Under Review.***

Chapter 4 Li, Z., Shi, L., Cristea, A., Zhou, Y., Xiao, C., Pan, Z. (2022,

July). ***SimStu-Transformer: A Transformer-Based Approach to Simulating Student Behaviour.*** In Artificial Intelligence in Education. 23rd International Conference, AIED 2022, Durham, UK, July 27–31, 2022, Proceedings, Part II (pp. 348-351). Cham: Springer International Publishing.

Li, Z., Shi, L., Zhou, Y., Wang, J. (2023, May). ***Towards Student Behaviour Simulation: A Decision Transformer Based Approach.*** In Augmented Intelligence and Intelligent Tutoring Systems: 19th International Conference, ITS 2023, Corfu, Greece, June 2–5, 2023, Proceedings (pp. 553-562). Cham: Springer Nature Switzerland.

Chapter 5 Li, Z., Shi, L., Zhou, Y., Wang, J., Cristea, A. I., Zhou, Y. (2023, August) ***Sim-GAIL: A Generative Adversarial Imitation Learning Approach of Student Modelling for Intelligent Tutoring Systems.*** Neural Computing and Applications, 1-20.

Chapter 6 Li, Z., Shi, L., Zhou, Y., Wang, J. (2023, September) ***Broader and Deeper: A Multi-Features with Latent Relations BERT Knowledge Tracing Model.*** 18th European Conference on Technology Enhanced Learning, EC-TEL 2023, Aveiro, Portugal, September 4–8, 2023, Proceedings. Cham: Springer International Publishing, 2023.

Chapter 7 Li, Z., Shi, L., Zhou, Y., Wang, J. (2023, November) ***LBKT: A LSTM BERT-based Knowledge Tracing for Long-Sequence Data***. The 30th International Conference on Neural Information Processing (ICONIP2023). ***Under Review.***

Note on Publications Not Included in This Thesis: As well as the above papers, the following works have been published during the period of research for

this thesis; they have helped my deeper understanding of the work towards this thesis; however, these publications do not fit into the narrative of this thesis and have not been included in the text.

- Wang, J., Ivrissimtzis, I., Li, Z., Zhou, Y., Shi, L. (2023, May). ***User-Defined Hand Gesture Interface to Improve User Experience of Learning American Sign Language.*** In Augmented Intelligence and Intelligent Tutoring Systems: 19th International Conference, ITS 2023, Corfu, Greece, June 2–5, 2023, Proceedings (pp. 479-490). Cham: Springer Nature Switzerland.

- Wang, J., Ivrissimtzis, I., Li, Z., Zhou, Y., Shi, L. ***Exploring the Potential of Immersive Virtual Environments for Learning American Sign Language.*** 18th European Conference on Technology Enhanced Learning, EC-TEL 2023, Aveiro, Portugal, September 4–8, 2023, Proceedings. Cham: Springer International Publishing, 2023.

- Wang, J., Ivrissimtzis, I., Li, Z., Zhou, Y., Shi, L. ***Developing and Evaluating a Novel Gamified Virtual Learning Environment for ASL.*** 19th International Conference of Technical Committee 13 (Human- Computer Interaction) of IFIP (International Federation for Information Processing).

- Xiao, C., Shi, L., Cristea, A., Li, Z., Pan, Z. (2022, July). ***Fine-grained Main Ideas Extraction and Clustering of Online Course Reviews.*** In Artificial Intelligence in Education: 23rd International Conference, AIED 2022, Durham, UK, July 27–31, 2022, Proceedings, Part I (pp. 294-306). Cham: Springer International Publishing.

- Zhou, Y., Shi, L., Li, Z., Wang, J. ***Design Paradigms of 3D User Interfaces for VR Exhibitions.*** 19th International Conference of Technical Committee 13 (Human- Computer Interaction) of IFIP (International Federation for Information Processing).

published without the author's prior written consent and information derived from it should be acknowledged".

Acknowledgements

This thesis would not be possible without the support of many people. First and foremost, I would like to express my sincere gratitude to my supervisor, Dr Lei Shi, for his guidance, support, and invaluable insights throughout my PhD journey. His encouragement and expertise have been instrumental in shaping the direction of my research.

I would like to extend my thanks to my second supervisor, Prof. Alexandra I. Cristea, for her helpful feedback and constructive criticism. Her input has been instrumental in shaping the quality and scope of my research. I am deeply indebted to my review team, Dr Frederick Woon-Bor Li and Dr Ioannis Ivrissimtzis, for their insightful comments and suggestions that helped me refine my work and achieve greater clarity in my research. I would also like to acknowledge the support and camaraderie of my labmates, Yunzhan Zhou and Jindi Wang. Their presence has been a constant source of motivation and inspiration. In addition, I want to express my appreciation to my friends Nawal Wakil, Jiaxin Lou, Lei Wang, and my girlfriend Yuge Dong for their encouragement and support throughout the ups and downs of my PhD journey. I would also like to express my gratitude to my PhD thesis examiners: Dr Ahmed Kharrufa from Newcastle University and Dr Deng Jingjing from Durham University, for their kind suggestions and insightful discussions during my viva, which helped me to improve the presentation of this thesis.

Last but not least, I am deeply grateful to my parents, Mr Shen Li and Mrs Yuxiu Han, for their unwavering love, support, and encouragement throughout my life. Their guidance and sacrifices have been instrumental in enabling me to pursue my academic dreams.

Contents

List of Figures

List of Tables

List of Symbols

Acronyms / Abbreviations

AI Artificial Intelligence

AL Apprenticeship Learning

BC Behavioral Cloning

BERT Bidirectional Encoder Representations from the Transformers

GAIL Generative Adversarial Imitation Learning

GAN Generative Adversarial Networks

IL Imitation Learning

ITS Intelligent Tutoring System

KT Knowledge Tracing

LBKT LSTM BERT knowledge tracing model

LSTM Long Short-Term Memory

MDP Markov Decision Process

ML Machine Learning

MLFBK Multi-Features with Latent Relations BERT Knowledge Tracing model

NLP Natural Language Processing

RL Reinforcement Learning

TEL Technology Enhanced Learning

BP Back Propagation

BDL Bayesian Deep Learning

XAI Explainable Artificial Intelligence

SVM Support Vector Machine

GPT Generative Pre-Training

CNNs Convolutional Neural Networks

RNNs Recurrent Neural Networks

MLPs Multi-Layer Perceptrons

MAP Maximum a Posteriori

KL Kullback Leibler

DL Deep Learning

NLI Natural Language Inference

EDM Educational Data Mining

CHAPTER 1

Introduction

1.1 Background and Motivation

1.1.1 Background

Artificial Intelligence (AI) has made remarkable advancements in recent years, revolutionising various industries and domains. Its ability to extract patterns and insights from vast amounts of data has propelled advancements in areas such as image recognition, natural language processing, and autonomous systems [1]. As AI systems continue to permeate various aspects of human daily life, there is an increasing interaction between Humans and AI. Researchers are actively seeking to foster collaboration between humans and AI, moving beyond simple interaction to develop advanced systems that enable meaningful collaboration and mutual engagement, which involves Human-AI Collaboration [2]. Human-AI Collaboration entails combining Artificial Intelligence (AI) with human workflows to achieve shared objectives. This collaborative approach involves understanding mutual goals, coordinating tasks proactively, and tracking progress together [3].

In the current educational landscape, an increasing number of AI methods are being applied in Technology Enhanced Learning (TEL) [4]. TEL has become in-

creasingly important in providing high-quality education and facilitating lifelong learning [5]. The recent COVID-19 pandemic has further highlighted the need for effective educational technologies, including Intelligent Tutoring Systems (ITS) [6] to support remote learning and bridge the gap between traditional classroom settings and online platforms. Presently, designing TEL systems that foster effective collaboration between humans and AI is of utmost importance. The integration of AI technologies into educational systems has the potential to enhance learning experiences, promote individualised instruction, and provide personalised and timely feedback to learners. A key aspect of this system is the collaboration between the student (human) and the Intelligent Tutoring System (ITS) (AI) to achieve a common educational aim — **Maximise the performance of Intelligent Tutoring Systems (ITS) whilst minimising the input and interaction required from students** [7]. However, despite the potential benefits of integrating AI into educational systems, there are several challenges that need to be addressed. The first challenge, not only for education researchers but also for HCI researchers and developers, is the lack of clear guidance on designing and building user-friendly systems that effectively facilitate collaboration between humans and AI.

The collaboration between humans and AI is a partnership to accomplish shared objectives [8]. AI systems are designed with specific functions, and their role is to assist humans in achieving these goals. This collaboration between humans and AI involves two perspectives: the first is humans-to-AI, humans enhancing the AI system's training through specific interactive behaviours, addressing challenges such as cold starts or limited training data [9].

AI-to-human collaboration, on the other hand, refers to AI systems aiding humans in accomplishing specific tasks. Consider ITS as an example: ITS is designed to assist students in achieving better academic outcomes. Students' objectives include obtaining improved grades through the use of AI systems. In this context, collaboration involves students actively engaging with AI systems, enabling the AI to understand the students' challenges better and enhance the efficiency of their learning process.

In order to develop a successful human-AI collaborative system, it is essential to

understand the dynamics of this interaction and optimise it for improved learning outcomes. To address this problem, Chapter 3 provides a comprehensive survey of human-AI collaborative systems, examining the current state of the field and identifying key challenges and opportunities. In this Chapter, different human-AI collaborative frameworks were proposed. We use one of the frameworks (shown in 3.1) to structure subsequent research, providing insights into designing and developing user-friendly and efficient Student-ITS collaborative systems.

We conceptualise the interaction between the student and ITS as a "human-AI collaborative process," where the student represents the "human" component and the ITS represents the "AI" component (shown in figure 1.1). The aim of this collaborative process is to leverage the capabilities of AI to optimise the learning experience for students while also considering enhancing the ITS training process by adapting student modelling methods.

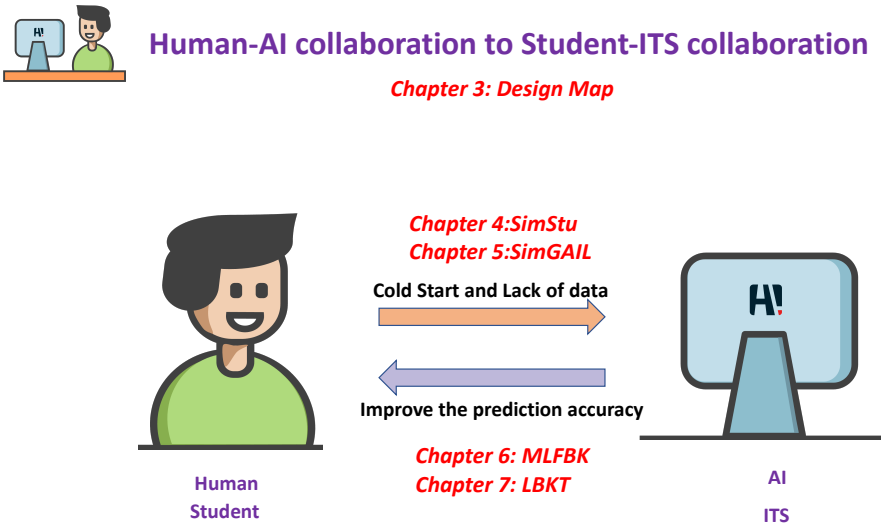


Figure 1.1: Human-AI collaboration to Student-ITS collaboration.

Within this collaborative process, there were two key challenges. The first challenge faced by the Student-to-ITS process (Human-to-AI process) is the lack of interaction data in the early stages of ITS development [10]. As new ITS is designed and deployed, there is limited availability of labelled data to train the algorithms

that drive algorithms in ITS. This scarcity of interaction data hampers the system's ability to model students' knowledge states accurately and predict their learning trajectories at the early stage of ITS training, which is also called the cold start problem [11]. Moreover, ITS needs a substantial amount of student interaction data to train the algorithms effectively. ITS relies on a large volume of data to capture the complexity of student learning behaviours and make accurate predictions. However, collecting and labelling such a vast amount of data can be time-consuming, expensive, and logistically challenging [12]. Therefore, it is crucial to develop methods that could address this challenge of data scarcity and enable efficient training of ITS models.

To address this challenge, we proposed two innovative student modelling approaches in Chapters 4 and 5 to generate simulation data to enhance the human-to-AI process in the development of ITS. These approaches focus on simulating student behaviour data to augment the limited real interaction data available during the ITS development. The proposed student modelling methods, namely SimStu and Sim-GAIL, leverage advanced deep learning and reinforcement learning techniques, including the Decision Transformer [13] and Generative Adversarial Imitation Learning (GAIL) [14], to generate high-fidelity and diverse simulated student behavioural data for ITS training. By incorporating these simulated data into the training process, the ITS could overcome the scarcity of labelled data and enhance its performance, leading to more accurate predictions and personalised instruction.

Another challenge is from the ITS-to-Students process (AI-to-Human process). The aim of the ITS is to help the student to achieve higher learning performance, which requires a deep understanding of the learning process and knowledge mastery level. Therefore, efficient and accurate modelling of students' knowledge states and learning trajectories is essential. Knowledge Tracing (KT) plays a crucial role in ITS by modelling students' learning progress and predicting their future actions based on their past behaviour data [15]. KT enables ITS to estimate students' knowledge states, identify areas of weakness, and provide tailored feedback, hints, and additional learning resources [16]. However, current KT models often struggle to provide accurate predictions, hindering the ability of ITS to deliver effective

personalised instruction [17].

One main problem is that previous KT models have often been limited by their reliance on a single or few features, which could fail to capture the complexity of student learning behaviour data [18, 19]. To address this gap, Chapter 5 proposed MLFBK, which introduces a novel approach that incorporates multiple features and mines latent relations. By considering a broader range of features and exploring the latent relationships between them, MLFBK aims to capture the complexity of student learning behaviour data more effectively. This approach has the potential to improve the accuracy and performance of KT models.

Another problem is that traditional probabilistic and logistic KT models struggle with processing efficiency and memory usage when dealing with growing amounts of longer sequence data [20–22]. To address this challenge, Chapter 7 proposed LBKT, which is a knowledge tracing approach that combines the strengths of the Bidirectional Encoder Representations from the Transformers (BERT) model and the Long Short-Term Memory (LSTM) model. BERT captures complex data relations, while LSTM handles long sequential data. The model leverages a Rasch model-based embedding method to incorporate difficulty-level information from students' historical behaviour data. This embedding improves the model's performance and interpretability.

By integrating the insights and methodologies from Chapters 4 to 7, this thesis aims to develop innovative approaches to enhance the collaborative process between students and Intelligent Tutoring Systems (ITS). The collaborative process consists of two main parts: the Student-to-ITS (human-to-AI) process, which focuses on generating simulated student behaviour data, and the ITS-to-Student (AI-to-human) process, which emphasises the development of advanced KT models. The integration of these components enables the design of more effective and personalised ITS, enhancing the learning experiences and outcomes of students.

1.1.2 Personal Motivation

My journey into the field of Artificial Intelligence and Technology Enhanced Learning has been driven by a deep personal motivation. From an early age, I was

fascinated by the potential of technology to transform education, making it more accessible and personalised.

Growing up, I experienced the strengths and limitations of traditional education systems firsthand. This fuelled my passion to bridge the gap between human potential and educational technology, particularly in the context of Intelligent Tutoring Systems (ITS).

The challenges addressed in this thesis are not just academic pursuits for me; they are part of my commitment to revolutionise education. My personal motivation is to make learning more adaptive and effective for learners of all backgrounds.

As I embarked on this research journey, my personal motivation has guided me to explore innovative solutions and contribute to the advancement of Technology Enhanced Learning. I am excited to share the outcomes of this research, hoping to inspire further progress in human-AI collaboration and educational technology.

1.2 Research Questions & Objectives

The main research questions in this thesis can be summarised as follows:

Research Theme 1: Exploring the design map and trajectories of the Human-AI Collaborative Systems.

The first research theme of this thesis pertains to exploring how to design a user-friendly and high-efficiency human-AI collaborative system. With the increasing integration of AI technologies into various domains, it is essential to develop collaborative systems that effectively leverage the strengths of both humans and AI to achieve optimal performance and user satisfaction. This research theme addresses the following research question:

Research Question 1: ***What trajectory could we follow to develop a user-friendly Student-ITS collaboration system?***

The following objectives address this research question:

RO 1.1: Investigate existing human-AI collaboration theories and frameworks to identify key principles and best practices for designing a user-friendly and high-efficiency human-AI collaboration system.

RO 1.2: To create a new Human-AI System design guidance Map.

RO 1.3: To conceptualise the existing frameworks from the perspectives of different perspectives.

RO 1.4: To elaborate generic Human-AI CRL challenges, providing the research community with a guide towards novel research directions.

Research Theme 2: Proposing efficient approaches to solve the cold-start problems and scarcity of data for ITS training

The second research theme focuses on addressing the challenges of the Student-to-ITS (Human-to-AI) process, the cold-start problems and data scarcity problems in the development of Intelligent Tutoring Systems (ITS). These challenges hinder the effective training of ITS models, which heavily rely on large amounts of labelled data. This research theme aims to propose efficient approaches to generate high-fidelity and diverse simulated student behaviour data for ITS training. The following research question is investigated:

Research Question 2: ***How can we generate high-fidelity and diverse simulated student behaviour data to train Intelligent Tutoring Systems (ITS) using deep learning methods?***

This research question is addressed by the following objectives:

RO 2.1: Investigate designing a deep learning-based model to generate simulated student behavioural data.

RO 2.2: To evaluate the performance of the proposed model and compare the proposed model with other existing methods.

RO 2.3: Applying the proposed model's generated data to the emerging ITS technology to test the performance.

Research Question 3: ***How can we generate realistic and diverse simulated student behaviour data for training Intelligent Tutoring Systems (ITS) through reinforcement learning techniques?***

The following objectives address this research question:

RO 3.1: To propose a student modelling method based on a reinforcement learning approach.

RO 3.2: To compare the performance of the proposed method with other student

modelling methods.

RO 3.3: To evaluate the effectiveness of the proposed method in improving the KT model's prediction accuracy.

Research Theme 3: Improving the performance of Knowledge Tracing models.

The third research theme focuses on enhancing the performance of the ITS-to-Student (AI to Human) process. More specifically, improving the performance of Knowledge Tracing models is an essential task for Intelligent Tutoring Systems. KT models aim to estimate students' knowledge states accurately and predict their future learning behaviours based on their interaction data. This research theme investigates methods to improve the accuracy and efficiency of KT models. The following research questions are addressed:

Research Question 4: ***How can multiple features and latent relations in student interaction data be integrated to improve the accuracy and efficiency of Knowledge Tracing (KT) models for Intelligent Tutoring Systems (ITS)?***

This research question is addressed by the following objectives:

*RO 4.1:*Identify relevant features and latent relations in student interaction data that can contribute to improving the accuracy and efficiency of Knowledge Tracing (KT) models.

RO 4.2: Develop novel methods to integrate multiple features and latent relations into KT models, leveraging techniques such as feature engineering, latent variable modelling, and deep learning architectures.

RO 4.3: Evaluate the performance of the proposed KT models in terms of accuracy, efficiency, and interpretability, comparing them with baseline models using benchmark datasets and real-world ITS data.

Research Question 5: ***How to effectively deal with large-scale datasets, process long-sequence data, and improve the performance of KT models for ITS?***

This research question is addressed by the following objectives:

RO 5.1: To propose a novel Knowledge Tracing model for processing long se-

quence data in Intelligent Tutoring Systems.

RO 5.2: Evaluate the performance and scalability of the proposed approaches in processing large-scale datasets and long sequence data, comparing them with existing methods using benchmark datasets and real-world ITS data.

RO 5.3: To conduct an ablation study to analyse the impact of each component of the method's overall performance and use the visualisation tool to demonstrate the interpretability.

1.3 Main Contributions

This thesis contributes to the field of Human-AI collaborative systems, Intelligent Tutoring Systems, and Knowledge Tracing. The main contributions of this thesis can be summarised as follows:

1. The thesis provides a systematic framework for the design of efficient and 'natural' human-AI collaborative methods systems in Chapter 3, which offers an extensive survey of CRL methods, conceptualises existing frameworks, and creates a new Human-AI CRL Design Trajectory Map as a systematic modelling tool for the selection of existing CRL frameworks, as well as a method of designing new CRL systems. Chapter 3 also elaborates on generic Human-AI CRL challenges, providing the research community with a guide towards novel research directions.

2. Proposed a Transformer-based approach to generate simulated student behavioural data for training Intelligence Tutoring Systems (ITS). Chapter 3 addresses the challenge of the scarcity of data sets providing interactions between students and ITS, which is necessary for training personalised ITS. SimStu generates simulated interactions between sim students and ITS, which can be used to train ITS to provide customised learning strategies and trajectories to real students. Chapter 3 presents an upgraded version of SimStu, which improves the model's performance by modifying the input and hyperparameters. The experimental results showed that SimStu could model the real student

well in terms of action frequency Sim-Stu distribution and elapsed time distribution. Moreover, the evaluation of SimStu in an emerging ITS technology, Knowledge Tracing, indicated that SimStu could improve the efficiency of ITS training.

3. Proposed a student modelling method called Sim-GAIL, which is based on Generative Adversarial Imitation Learning (GAIL) approach. This method can be used to train Intelligent Tutoring Systems (ITS) by replacing human students with sim-students (simulated students via student modelling). Chapter 5 compares the performance of Sim-GAIL with two traditional Reinforcement Learning-based and Imitation Learning-based methods using action distribution evaluation, Sim-GAIL cumulative reward evaluation, and offline-policy evaluation. The experimental results suggest that Sim-GAIL outperforms traditional student modelling methods on most metrics. Moreover, Chapter 5 applies Sim-GAIL to a domain plagued by the cold start problem, Knowledge Tracing (KT), and the experimental results show that Sim-GAIL could effectively improve the KT model's prediction accuracy in a cold-start scenario.

4. This thesis proposed a Multi-Features with Latent Relations BERT Knowledge Tracing model (MLFBK) that utilises multiple features and mines latent relations between features to improve the performance of the Knowledge Tracing (KT) model. The model incorporates four data features (student id, skill id, item id, and response id) and three meaningful latent relations among features to improve the performance: individual skill mastery, ability profile of students (learning transfer across skills), and problem difficulty. Chapter 6 also presents experimental results that demonstrate that the proposed algorithm outperforms baseline methods and demonstrates good interpretability.

5. Propsed a novel LSTM BERT-based Knowledge Tracing model, LBKT, for processing long sequence data in Intelligent Tutoring Systems. LBKT uses a BERT-based architecture with a Rasch model-based embeddings block to deal with different difficulty levels information and an LSTM block to process

the sequential characteristic in students' actions. The model achieves better performance on most benchmark datasets on the metrics of ACC and AUC. Additionally, an ablation study is conducted to analyse the impact of each component of LBKT's overall performance.

By addressing these research challenges and making these contributions, this thesis aims to advance the field of Human-AI collaboration in educational settings, enhance the capabilities of Intelligent Tutoring Systems, and improve the accuracy and interpretability of Knowledge Tracing models. These contributions have the potential to significantly impact the field of Technology Enhanced Learning and pave the way for more personalised and effective educational experiences.

1.4 Thesis Structure

This thesis is organised into the following chapters:

- **Chapter 1** provides an overview of the research background, motivation, research questions, the corresponding objectives, and the main contributions of this thesis. It also outlines the structure of the thesis.

- **Chapter 2** presents a comprehensive review of relevant literature in the fields of Human-AI Collaborative Systems, Deep Learning Methods, Reinforcement Learning, and Knowledge Tracing Methods. This Chapter aligns the contributions of this thesis with the existing literature, highlighting the research gaps addressed by the proposed models.

- **Chapter 3** focuses on the design principles and methodologies for developing a user-friendly and high-efficiency human-AI collaborative system. Drawing on the insights from this Chapter, we present a structured framework for the student-ITS collaborative process **(RQ1)**.

- **Chapter 4** addresses the Student-to-AI collaborative process, which delves into another simulation method, the SimStu, which utilizes a decision transformer to simulate student learning trajectories. We investigate its impact on

training powerful ITS models using limited human actions. We compare the results with existing approaches and analyse the efficiency and effectiveness of the proposed method **(RQ2)**.

- **Chapter 5** presents the Sim-GAIL model, which utilises generative adversarial imitation learning to simulate student learning trajectories. This chapter covers the model's design, implementation, and evaluation, highlighting its effectiveness in addressing the cold start problem in Intelligent Tutoring Systems **(RQ3)**.

- **Chapter 6** addresses the challenge of improving the accuracy and efficiency of Knowledge Tracing models in intelligent tutoring systems. This Chapter proposed the MLFBKT, which integrates multiple features and latent relations in student interaction data to improve the performance of the KT model. We analyse the performance of the proposed model and compare it with existing state-of-the-art KT models in real word dataset**(RQ4)**.

- **Chapter 7** focuses on the improvement of knowledge tracing models by effectively dealing with large-scale datasets and processing long sequence data. We proposed the LSTM BERT knowledge tracing model (LBKT) and its effectiveness in processing long sequence data. We evaluate the performance of the model on benchmark datasets and analyze its interpretability **(RQ5)**.

- **Chapter 8** summarises the main findings, contributions, and implications of the thesis. It provides a concise recapitulation of the research objectives and addresses the research questions outlined in the introduction. This chapter concludes by highlighting the impact and potential future developments in the field.

CHAPTER 2

Background

This chapter starts by reviewing the general background of Human-AI collaborative systems. Subsequently, the chapter proceeds to introduce the technologies employed in this thesis, focusing on Deep Learning (DL) and Reinforcement Learning (RL) methods. Within the realm of DL, we have explored the LSTM, Transformer, and BERT models, renowned for their capabilities in handling sequential and contextual data. Furthermore, the chapter explores reinforcement learning methods, including Imitation learning and Generative Adversarial Imitation Learning (GAIL). Next, this chapter introduces the DL and RL methods applied in educational scenarios, with a specific focus on student modelling techniques and Knowledge Tracing methods.

2.1 Human-AI Collaborative Systems

In recent years, the rapid development of Artificial Intelligence (AI) has sparked both optimism and concern in various domains. While AI is often depicted in extremes as either a saviour or a threat to humanity, the reality lies somewhere in between [23, 24]. To navigate this complex landscape, exploring how humans and AI can collaborate effectively and complement each other's shortcomings is

crucial. This collaboration has the potential to leverage the strengths of AI in well-defined tasks and human intuition, creativity, and ethical considerations in complex decision-making scenarios.

The field of human-computer interaction (HCI) has a long history of studying the interaction between humans and computers, offering valuable insights into effective collaboration patterns. Various models and paradigms have been proposed to guide human-computer collaboration, such as Cognitive Systems Engineering (CES) [25], Schmidt collaboration patterns [26], and co-active design patterns [27]. These models provide frameworks for understanding the roles, responsibilities, and interaction dynamics between humans and computers.

As an emerging research direction in Human-Computer Interaction (HCI), Human-AI collaborative systems have gained significant attention within the machine learning field [3]. Despite the growing interest and research efforts in this field, there remains a lack of surveys or literature reviews specifically exploring the collaboration between humans and AI agents, particularly in the context of reinforcement learning. Only a limited number of surveys have been published, with a focus on topics such as reinforcement learning based on human advice [28], human-centred reinforcement learning [29], explainable reinforcement learning [30], and the design principles and open challenges of interactive reinforcement learning [9]. The collaboration between humans and AI agents in reinforcement learning remains an underexplored area, with a scarcity of literature addressing this specific research direction.

In this context, the aim of this thesis is to investigate and design a structured approach for constructing human-AI collaborative systems. By examining existing collaboration approaches, design patterns, and algorithmic models, we seek to provide insights into effective collaboration strategies and potential directions for future research in this field.

2.2 Deep Learning

Deep Learning (DL) [31]is a branch of machine learning which is primarily based on neural networks and representation learning. Deep Learning has emerged as a powerful approach in the field of artificial intelligence and has made significant contributions to various domains, including natural language processing [32], computer vision [33], and speech recognition [34]. This section provides an overview of deep learning techniques, focusing on two prominent models: Transformer and BERT.

2.2.1 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a class of deep learning models widely used for processing sequential data, making them well-suited for tasks involving time series or sequential information [35]. RNNs incorporate feedback connections, allowing them to retain and utilize information from previous time steps. This capability makes RNNs effective in language modelling, speech recognition, and sequence prediction tasks. In the context of education, RNNs have been applied to tasks such as automatic essay grading, student performance prediction, and language modelling for educational dialogue systems. The architecture of the RNN is shown in the figure 2.1. It consists of an input layer for receiving sequential data, a hidden layer with recurrent connections allowing information to flow across time steps, and an output layer for producing predictions or classifications. The recurrent connections allow RNNs to capture temporal dependencies in data, making them suitable for tasks like natural language processing and time series analysis.

An RNN consists of an input layer (x), a hidden layer (s), and an output layer (y). It can be seen as a set of short-term memory units that process sequential data. Figure 1(b) illustrates the unfolded diagram of an RNN for an input sequence. In this context, deep RNNs have been proposed to leverage the advantages of deeper networks and address the challenges of training deep architectures.

One major challenge of RNNs is the issue of vanishing and exploding gradients, where the gradients either decay or explode exponentially during training. This sensitivity to gradient scaling can result in the network forgetting initial inputs or

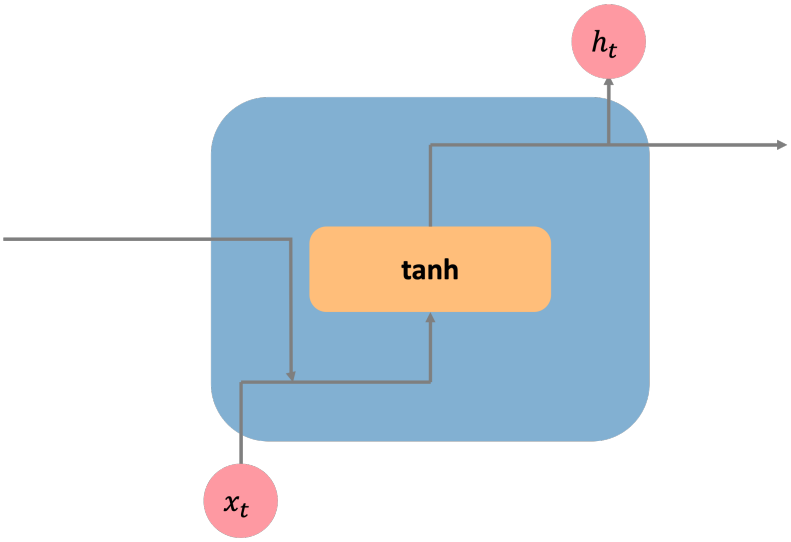


Figure 2.1: The architecture of the RNN.

experiencing unstable learning. To tackle this problem, Long Short-Term Memory (LSTM) networks have been introduced [36]. LSTM networks incorporate memory blocks with memory cells and gated units to control information flow, enabling them to capture long-term dependencies more effectively. Furthermore, the use of residual connections [37] in very deep networks can mitigate the vanishing gradient problem. Residual connections allow the gradients to propagate more directly through the network, facilitating the training of deeper architectures.

2.2.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of recurrent neural network architecture that overcomes the vanishing gradient problem and enables the modelling of long-term dependencies in sequential data [36]. LSTMs utilize memory cells with input, forget, and output gates to selectively store and retrieve information over multiple time steps. The equations governing the behaviour of an LSTM cell are as follows:

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
\tag{2.1}
$$

Where x_t is the input at time step t, h_t is the hidden state at time step t, c_t is the cell state at time step t, f_t, i_t, and o_t are the forget, input, and output gates respectively, and σ represents the sigmoid activation function. LSTMs have been effective in tasks that require capturing long-term dependencies, such as language translation, speech recognition, and sentiment analysis [38]. The architecture of the LSTM is shown in the figure 2.2. LSTMs have a unique structure with memory cells that can store and retrieve information over long sequences. They consist of three gates (input, forget, and output) and a cell state. The input gate controls the flow of new information into the cell state, the forget gate regulates what information should be discarded, and the output gate determines what information should be used to make predictions.

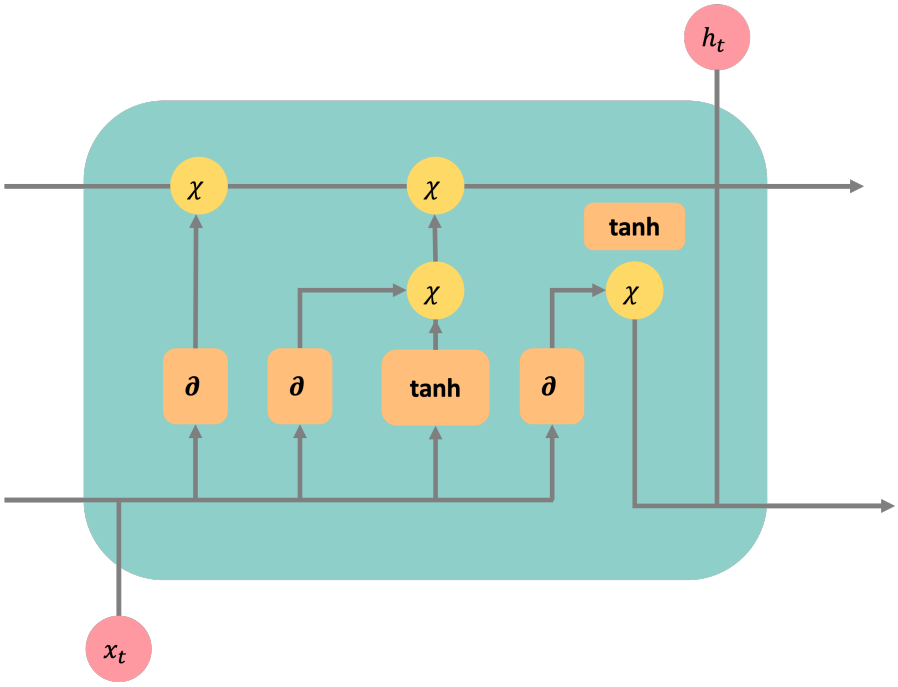


Figure 2.2: The architecture of the LSTM.

2.2.3 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are a class of deep learning models consisting of a generator network and a discriminator network, trained together in a competitive manner [39]. GANs have gained significant attention for their ability to generate realistic synthetic data that resembles the training data distribution. The objective of the generator is to produce data samples that fool the discriminator, while the discriminator aims to distinguish between real and generated samples. The training process can be formulated as a minimax game with the following objective function:

$$\min_G \max_D V(D,G) = \mathbb{E}x \sim p\text{data}[\log(D(x))] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))] \tag{2.2}$$

where D represents the discriminator, G represents the generator, x represents real data samples, z represents random noise vectors, p_{data} represents the real data distribution, and p_z represents the noise distribution. The architecture of the LSTM is shown in the figure 2.3. It consists of two neural networks, a generator and a discriminator, engaged in a competitive training process. The generator generates fake data samples from random noise while the discriminator tries to distinguish real data from fake ones. They are trained iteratively in a minimax game, where the generator aims to produce data that can't be distinguished from real data, and the discriminator aims to improve its ability to tell real from fake.

2.2.4 Transformer

The Transformer model, proposed by Vaswani *et al.*, is a neural network architecture that has gained widespread popularity in the field of deep learning, particularly in natural language processing (NLP) tasks [40]. The Transformer utilises a self-attention mechanism to extract inherent features and effectively model long-range dependencies in sequential data. It consists of stacked self-attention layers, as shown in Equation 2.3, allowing the model to attend to different positions in the input sequence and capture complex patterns and relationships.

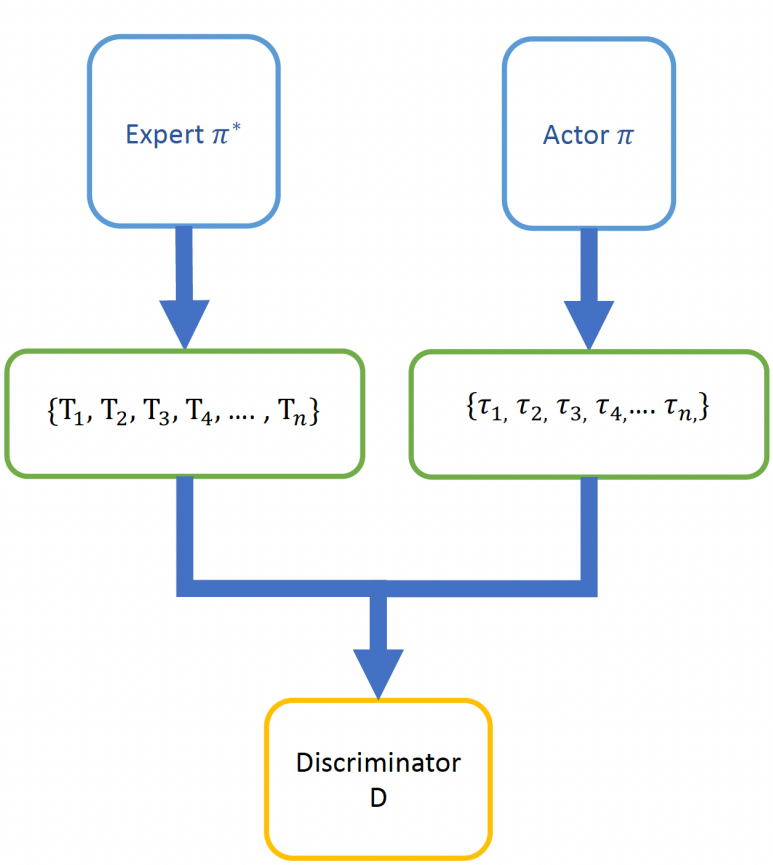


Figure 2.3: The architecture of the GAN.

$$z_i = \sum_{j=1}^{n} \text{softmax}\left(\langle q_i, k_{j'}\rangle \; j' = 1^n\right) j \cdot v_j \tag{2.3}$$

In the above equation, z_i represents the i-th output embedding, q_i denotes the query of the i-th token, $k_{j'}$ represents the key of the j'-th token, and v_j represents the value of the j-th token. The self-attention mechanism computes attention weights between query and key pairs, which are then used to weigh the corresponding values and generate the output embeddings.

The Transformer architecture revolutionized the field of NLP, allowing for more efficient and effective processing of natural language text. Its ability to capture bidirectional contextual information by considering dependencies between preceding and succeeding tokens in a sequence is one of its key strengths. Additionally, the self-attention mechanism enables the model to dynamically weigh the importance of different tokens, improving its ability to capture complex patterns and relation-

ships in the data. The architecture of the transformer is shown in the figure 2.4. It relies on a multi-head self-attention mechanism to efficiently capture relationships between elements in input sequences. Transformers enable effective modelling of complex data dependencies with layers of position-wise feed-forward networks, layer normalization, and residual connections. They are highly modular and scalable, with stacked layers for capturing hierarchical features. Transformers are used in encoder and decoder configurations, making them versatile for various tasks, including machine translation and language generation. Their innovation has led to state-of-the-art models like BERT and GPT, setting new language understanding and generation standards.

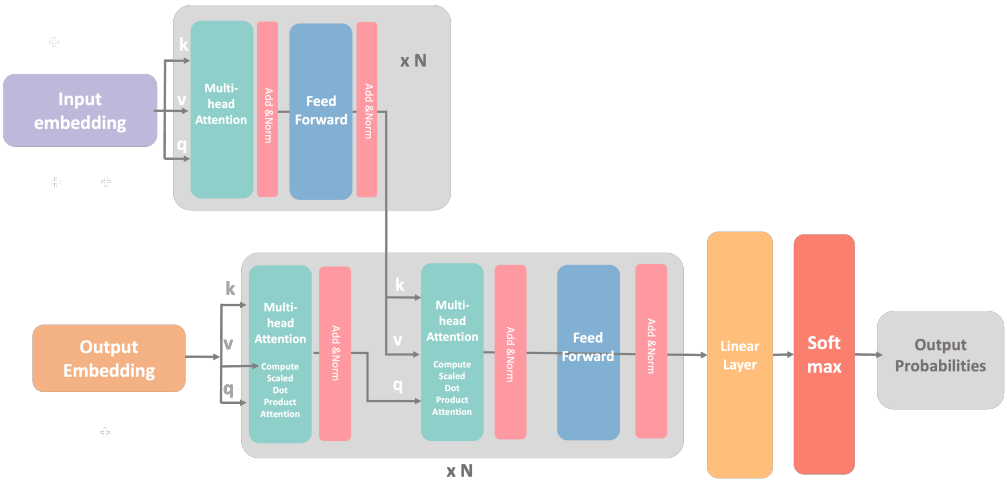


Figure 2.4: The architecture of the Transformer.

2.2.5 BERT

BERT (Bidirectional Encoder Representations from Transformers), introduced by Devlin *et al.*, is a pre-trained Transformer-based language model that has achieved remarkable performance across various NLP tasks [41]. At its core, BERT retains the key components of the Transformer architecture, including multi-head self-attention, positional encodings, and feed-forward neural networks. What sets BERT apart is its bidirectional pre-training approach, where it learns to predict missing words from both left and right contexts simultaneously, capturing rich semantic information and

context in natural language. This bidirectional context understanding is crucial for its remarkable performance in various NLP tasks.

One of the key features of BERT is its bidirectional context, which allows it to capture dependencies between preceding and succeeding tokens in a sequence. This bidirectional modelling significantly improves the understanding of context and semantics. BERT's training process involves masking some tokens in the input sequence and training the model to predict those masked tokens based on the surrounding context. This approach enables BERT to learn a robust language representation that can be fine-tuned for downstream tasks with relatively small amounts of labelled data.

The success of BERT can be attributed to several factors. The large-scale pretraining corpus used in its training allows it to capture a wide range of linguistic patterns and general knowledge. BERT's transformer architecture, with its self-attention mechanism, effectively captures global dependencies between tokens, enabling it to model complex relationships in the data. Additionally, BERT is known for generating high-quality embeddings, which are crucial for various natural language processing tasks.

BERT has achieved remarkable performance in NLP tasks and has been adapted and applied to other domains with excellent results. For example, ConvBERT applies the original BERT architecture in image processing tasks [42], BERT4Rec [43] enhances the performance of recommendation systems [43], and LakhNES [44] incorporates BERT to improve the quality of music generation. However, in the Knowledge Tracing field, although some BERT-based models, such as BEKT [45] and BiDKT [46], have been proposed to improve performance.

2.3 Reinforcement Learning

Sequential decision-making problems are commonly modelled using Markov Decision Process (MDP), which serves as the foundation for reinforcement learning (RL) [47]. RL is a machine learning paradigm that enables an agent to learn optimal decision-making policies through interactions with an environment, aiming to maximize cu-

mulative rewards [48]. In this section, we delve into MDP and RL, providing a comprehensive understanding of these concepts.

2.3.1 Markov Decision Process

Markov Decision Process (MDP) [49] is a mathematical framework used to model decision-making problems in a stochastic environment that exhibits Markov properties [50]. It comprises interacting components, namely agents and environments and encompasses states, actions, policies, and rewards. In an MDP, the agent observes the current state of the environment and selects actions based on a policy, which dictates its behaviour. These actions impact the state transition of the environment, and the agent receives rewards based on the chosen actions and resulting states. The primary objective of an agent in an MDP is to find a policy that maximizes the cumulative reward over time. The MDP can be represented as follows:

$$M = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}) \tag{2.4}$$

where $\mathcal{S}$ represents the set of possible states, $\mathcal{A}$ denotes the set of possible actions, $\mathcal{T}$ represents the state transition probabilities, and $\mathcal{R}$ denotes the reward function. The agent's goal is to find an optimal policy π^* that maximizes the expected cumulative reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right] \tag{2.5}$$

where γ is the discount factor, and R_t is the reward received at time step t.

2.3.2 Reinforcement Learning

Reinforcement Learning (RL) is a machine learning approach that tackles the problem of learning optimal decision-making policies through interactions with an environment [48]. RL can be defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ represents the set of possible states, $\mathcal{A}$ denotes the set of possible actions, $\mathcal{T}$ represents the state transition probabilities, $\mathcal{R}$ denotes the reward function, and γ is the discount factor.

The objective of an RL agent is to learn an optimal policy π that maximizes the expected cumulative reward. The agent's policy is typically represented by a value function or an action-value function, which estimates the expected return from a given state or state-action pair. The value function is defined as:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s, \pi\right] \tag{2.6}$$

where $V^{\pi}(s)$ represents the expected cumulative reward starting from state s and following policy π. The action-value function is defined as:

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s, a_0 = a, \pi\right] \tag{2.7}$$

where $Q^{\pi}(s, a)$ represents the expected cumulative reward starting from state s, taking action a, and following policy π. RL algorithms aim to estimate these value functions and utilize them to guide the agent's decision-making process.

2.3.3 Imitation Learning

While RL involves learning policies by interacting with the environment to maximize rewards, Imitation Learning (IL) focuses on emulating expert behaviour by learning from expert demonstrations [51]. Unlike RL, which relies on an explicit reward function, IL aims to learn a policy that imitates the behaviour demonstrated by experts.

Behavioral Cloning

Behavioral Cloning (BC) is a type of IL that approaches policy learning as a supervised learning problem, using expert state-action pairs [52, 53]. BC can be effective but heavily relies on a large volume of data. Without sufficient data, distributional mismatch, also known as covariate shift, can occur during test time due to errors and stochasticity in the environment.

BC directly maps the states/contexts to actions/trajectories by leveraging the demonstration provided by an expert/oracle. After generating the control input or

trajectories, the loss function L is designed according to the problem formulation and optimized in a supervised learning fashion. The state-of-the-art BC uses the negative log-likelihood loss to update the policy, i.e.,

$$\arg\min_{\pi} L(\pi) = -\frac{1}{N}\sum_{k=1}^{N} \log \pi(a_k|s_k) \tag{2.8}$$

The above equation outlines the state-of-the-art BC process. While traditional BC has less connection to MDP compared to other prevalent methods, its efficiency is guaranteed. However, it suffers when the agent visits an unseen state. The loss function L can be customized for a specific problem formulation, and various existing loss functions are available to measure differences, such as the L_1 loss, L_2 loss, KL divergence, and Hinge Loss. For example, when using KL divergence as the loss function, the objective policy can be obtained by minimizing the deviation between the expert distribution q_{π_E} and the induced distribution q_π, i.e.,

$$\pi^* = \arg\min_{\pi} \mathrm{DKL}(q_{\pi_E} \| q_\pi) \tag{2.9}$$

BC can be subdivided into model-free BC and model-based BC methods. The main difference lies in whether the method learns a forward model to estimate the system dynamics. Model-free BC methods perform well in industrial applications where accurate controllers are available, but they struggle to predict future states in "imperfect" environments. On the other hand, model-based BC methods leverage environment information to produce feasible outputs but have greater time complexity due to iterative learning involvement [54].

One significant BC method is DAgger, a model-free BC method proposed by Ross *et al.* [55]. The idea is to use dataset aggregation to improve generalization on unseen scenarios. DAgger adopts an iterative learning process and mixes a new policy $\hat{\pi}_{n+1}$ with probability β to construct the next policy. The mixing parameter β satisfies $\frac{1}{N}\sum_{i=1}^{N} \beta_i \to 0$. The startup policy is learned by BC and records the trajectory into the dataset. Unseen trajectories are recorded by combining expert corrections to alleviate the problem of traditional BC methods performing poorly in unseen scenarios. Subsequent research has proposed improvements on DAgger to enhance

data efficiency. However, DAgger involves frequent interaction with the expert, which may not be available or could be expensive. Later research has addressed this issue by actively learning to ask the demonstrator for help and minimizing context switches. Other recent methods have also alleviated this problem.

Apprenticeship Learning

Apprenticeship Learning (AL) seeks to identify generalizable features from expert demonstrations and find a policy that matches the expert's feature expectations [56]. AL aims to outperform the expert across a range of cost functions. However, AL may struggle to effectively imitate the expert trajectory when the true cost function lies outside the restricted class of cost functions, and there is no guarantee that the agent will surpass the expert's performance.

2.3.4 Generative Adversarial Imitation Learning

Generative Adversarial Imitation Learning (GAIL) addresses the limitations of RL and AL by incorporating the principles of Generative Adversarial Networks (GANs) into imitation learning [14]. GAIL is derived from a specific type of imitation learning called Maximum Causal Entropy Inverse Reinforcement Learning (MaxEntIRL) [57].

GAIL combines the benefits of constant regularizers and indicator regularizers in large environments. It introduces a novel cost regularizer, denoted as ψ_{GA}, which strikes a balance between the exact matching of occupancy measures and computational tractability. The formulation of ψ_{GA} is defined as follows:

$$\psi_{\mathrm{GA}}(c) = \begin{cases} \mathbb{E}_{\pi_E}[g(c(s,a))] & \text{if } c < 0 \\ +\infty & \text{otherwise} \end{cases} \tag{2.10}$$

where the function $g(x)$ is defined as:

$$g(x) = \begin{cases} -x - \log(1 - e^x) & \text{if } x < 0 \\ +\infty & \text{otherwise} \end{cases} \tag{2.11}$$

This cost regularizer penalizes cost functions that assign positive costs to expert state-action pairs while allowing any cost function that is negative everywhere. One key advantage of ψ_{GA} is its adaptability to arbitrary expert datasets, as it is an average over expert data. In contrast, the indicator regularizers δ_C, used in linear apprenticeship learning algorithms, are fixed and cannot adjust to data like ψ_{GA}.

The choice of ψ_{GA} is motivated by the fact that it approximates the optimal negative log loss of a binary classification problem. More specifically, it is proportional to the Jensen-Shannon divergence between the normalized occupancy distributions of the learner and the expert. By treating causal entropy H as a policy regularizer controlled by $\lambda \geq 0$, an imitation learning algorithm based on GAIL is formulated as:

$$\min_{\pi} \psi^*_{\mathrm{GA}}(\rho_\pi - \rho_{\pi_E}) - \lambda H(\pi) = \mathrm{DJS}(\rho_\pi, \rho_{\pi_E}) - \lambda H(\pi) \tag{2.12}$$

This algorithm aims to find a policy π that minimizes the Jensen-Shannon divergence between the learner's and expert's occupancy measures, while taking into account the policy regularizer. Unlike linear apprenticeship learning algorithms, GAIL can imitate expert policies exactly because it minimizes a true metric between occupancy measures.

To implement GAIL, a saddle point (π, D) of the expression is sought,

$$\mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi) \tag{2.13}$$

where both π and D are represented using function approximators. Specifically, GAIL employs a parameterized policy π_θ with weights θ and a discriminator network $D_w : S \times A \to (0, 1)$ with weights w. The algorithm alternates between a gradient step on w to increase the expression and a Trust Region Policy Optimization (TRPO) step on θ to decrease it. The discriminator network acts as a local cost function that provides a learning signal to the policy, enabling policy updates that move toward expert-like regions of state-action space.

In summary, GAIL leverages generative adversarial networks (GANs) to enable imitation learning by finding a policy that minimizes the Jensen-Shannon divergence

between the learner's and expert's occupancy measures. The algorithm employs a parameterized policy and a discriminator network, alternating between updating the discriminator and optimizing the policy using TRPO. This approach allows for the exact imitation of expert policies and is designed to work well in large environments.

2.4 Knowledge Tracing

Knowledge Tracing (KT) plays a crucial role in Intelligent Tutoring Systems (ITS) by modelling and predicting students' learning trajectories based on their historical interaction data with the system [58]. KT models could be broadly classified into three categories: probabilistic KT models, logistic KT models, and deep learning-based KT methods [17].

2.4.1 Probabilistic KT models

Probabilistic KT models use probabilistic graphical models to track students' changing learning states based on observed learning performance [59]. There are two kinds of Probabilistic KT models: Bayesian Knowledge Tracing (BKT) and Dynamic Bayesian Knowledge Tracing (DBKT).

Bayesian Knowledge Tracing (BKT)

To the best of our knowledge, Bayesian Knowledge Tracing (BKT) is the first proposed Knowledge Tracing (KT) model [59]. BKT utilizes a Hidden Markov Model (HMM) framework, where unshaded nodes represent unobservable latent knowledge states, and shaded nodes represent observable student answers. BKT assumes a two-state student modelling framework: knowledge is either learned or unlearned, with no forgetting once learned. The transition probabilities in BKT are determined by the learning parameter $P(T)$ (transition from unlearned to learned state) and the forgetting parameter $P(F)$ (probability of forgetting previously mastered knowledge). Emission probabilities are determined by the performance parameters $P(G)$ (probability of correct answer despite non-mastery) and $P(S)$ (probability of mistake despite mastery). The initial probability of mastery is represented by $P(L_0)$.

BKT estimates the knowledge state and probability of correct answers using the following equations:

$$P(L_n) = P(L_n|Answer) + (1 - P(L_n|Answer))P(T) \tag{2.14}$$

$$P(C_{n+1}) = P(L_n)(1 - P(S)) + (1 - P(L_n))P(G) \tag{2.15}$$

$P(L_n)$ represents the probability that a knowledge component (KC) is mastered at the n-th learning interaction, while $P(C_{n+1})$ represents the probability of correct answers at the next interaction. $P(L_n)$ is the sum of two probabilities: the probability that the KC is already mastered and the probability that the knowledge state will convert to the mastered state. The posterior probability $P(L_n|Answer)$ is estimated as follows:

$$P(L_n|\text{correct}) = \frac{P(L_{n-1})(1 - P(S))}{P(L_{n-1})(1 - P(S)) + (1 - P(L_{n-1}))P(G)} \tag{2.16}$$

$$P(L_n|\text{incorrect}) = \frac{P(L_{n-1})P(S)}{P(L_{n-1})P(S) + (1 - P(L_{n-1}))(1 - P(G))} \tag{2.17}$$

Dynamic Bayesian Knowledge Tracing (DBKT)

While BKT models each KC independently, it does not consider their dependencies [60]. To address this, Käser *et al.* [61] proposed Dynamic Bayesian Knowledge Tracing (DBKT), which incorporates prerequisite hierarchies and inter-KC relationships using dynamic Bayesian networks. DBKT jointly models multiple skills within one model, enhancing the representational power of BKT. DBKT represents a student's knowledge mastery with binary latent variables and estimates it based on learning interactions. It models the dependencies between KCs, such as prerequisites. For example, if KC1 and KC2 are prerequisites for mastering KC3, a student's mastery of KC3 depends on their mastery of KC1 and KC2. The objective of DBKT is to find the parameters θ that maximize the joint probability $p(a_m, h_m|\theta)$. The log-likelihood can be formulated using a log-linear model:

$$L(w) = \sum_{m} \ln \left(\sum_{h_m} \exp \left(w^T \Phi(a_m, h_m) - \ln(Z) \right) \right) \tag{2.18}$$

Here, $\Phi : A \times H \rightarrow \mathbb{R}^F$ maps the observed space A and the latent space H to an F-dimensional feature vector. Z is a normalizing constant, and w represents the weights.

2.4.2 Logistic Models

Logistic models encompass a wide range of models based on logistic functions, which represent the probability of answering exercises correctly as a function of student and knowledge component (KC) parameters. These models estimate the parameters based on various factors in students' learning interactions and utilize logistic functions to predict the probability of mastery [62]. In this section, we introduce three logistic models: Learning Factor Analysis (LFA), Performance Factor Analysis (PFA), and Knowledge Tracing Machines (KTM).

Learning Factor Analysis (LFA)

The LFA model [63] considers the following learning factors:

- Initial knowledge state: Parameter α estimates the initial knowledge state of each student.
- Easiness of KCs: Parameter β captures the easiness of different KCs.
- Learning rate of KCs: Parameter γ represents the learning rate of KCs.

The standard LFA model can be expressed as follows:

$$p(\theta) = \sigma \left(\sum_{i \in N} \alpha_i S_i + \sum_{j \in KCs} (\beta_j + \gamma_j T_j) K_j \right), \tag{2.19}$$

where σ is the sigmoid function, S_i is the covariate for student i, T_j represents the covariate for the number of interactions on KC j, K_j is the covariate for KC j, and $p(\theta)$ is the estimated probability of a correct answer.

Performance Factor Analysis (PFA)

The PFA model [64] extends the LFA model by considering additional factors related to student performance:

- Previous failures: Parameter f represents the prior failures for the KC of the student.
- Previous successes: Parameter s denotes the prior successes for the KC of the student.
- Easiness of KCs: Parameter β captures the easiness of different KCs, as in the LFA model.

The standard PFA model can be expressed as follows:

$$p(\theta) = \sigma\left(\sum_{j \in KCs} (\beta_j + \mu_j s_{ij} + \nu_j f_{ij})\right), \tag{2.20}$$

Where μ and ν are the coefficients for s and f, respectively, representing the learning rates for successes and failures.

Knowledge Tracing Machines (KTM)

The KTM model [65] employs factorisation machines (FMs) to extend previous logistic models to higher dimensions. FMs are general predictors that work with any real-valued feature vector and can model interactions between variables using factorised parameters. KTM incorporates side information about exercises, students, KCs, or other related factors into the model. The knowledge mastery of the student is modelled based on a sparse set of weights for all features involved in the learning process. Let L be the number of features related to students, exercises, KCs, or other factors.

2.4.3 Deep learning-based KT models

The complexity of the cognitive process poses challenges for probabilistic or logistic models to capture its intricacies accurately. Deep learning, with its ability

to achieve non-linearity and extract features, is well-suited for modelling complex learning processes, especially when abundant learning interaction data is available. In recent years, several deep learning-based models for knowledge tracing (KT) have been proposed and achieved impressive performance. However, these models often lack interpretability due to their end-to-end learning strategy, limiting their applicability. In this section, we introduce deep learning-based KT models from five perspectives: deep knowledge tracing, memory-aware knowledge tracing, exercise-aware knowledge tracing, attentive knowledge tracing, and graph-based knowledge tracing.

Deep Knowledge Tracing (DKT)

Deep knowledge tracing (DKT) [66] was the first approach to incorporate deep learning into KT. It utilises recurrent neural networks (RNNs) to model the learning process of students. DKT employs RNNs to process the input sequence of learning interactions over time, maintaining a hidden state that implicitly encodes the history of past elements. The hidden state evolves based on the previous knowledge state and the current input-learning interaction. DKT provides a high-dimensional and continuous representation of the knowledge state, enabling it to capture the complexity of the learning process better. Long short-term memory (LSTM) networks, a variant of RNNs, are commonly used in DKT implementations due to their ability to handle long-term dependencies. DKT has demonstrated superior performance compared to probabilistic and logistic models. However, it lacks interpretability and faces challenges in explicitly determining a student's level of knowledge mastery from the hidden state.

Memory-aware Knowledge Tracing

To enhance the interpretability of DKT, memory-aware knowledge tracing models [67]introduce an external memory module to store and update the knowledge and mastery of students. The most notable model in this category is Dynamic Key-Value Memory Networks (DKVMN) for knowledge tracing. DKVMN utilises a static key matrix to store latent knowledge components (KCs) and a dynamic value matrix

to store and update the mastery of corresponding KCs over time. By reading and writing to the memory module, DKVMN effectively captures the temporal dynamics of knowledge acquisition. It addresses the limitations of DKT by considering the long-term dependencies in the learning process. A modified LSTM called Hop-LSTM is also proposed to improve the modelling of long-term dependencies in DKVMN.

Exercise-aware Knowledge Tracing

The text content of exercises plays a crucial role in understanding and answering them. Exercise-aware knowledge tracing (EKT) [16] leverages exercise text contents to mine their potential value for KT. EKT automatically learns the semantic representation of each exercise from its text contents using pre-training models like Word2Vec. It utilises bidirectional LSTM to capture the semantic word representation and constructs exercise embeddings. EKT also considers the KCs associated with each exercise and uses a memory module to represent the knowledge impact. The student's knowledge state is updated based on both the exercise embeddings and the knowledge impact. EKT demonstrates the importance of exercise text contents in the KT task and achieves improved performance.

Attentive Knowledge Tracing

Attentive knowledge tracing models [68] utilise the self-attention mechanism, popularised by the Transformer model, to capture global dependencies within a sequence of learning interactions. Self-Attentive Knowledge Tracing (SAKT) [69] directly applies the Transformer to capture long-term dependencies between students' learning interactions. Adaptive sparse self-attention networks and attention-based KT models further improve the self-attentive computation for KT. Separated self-attentive neural knowledge tracing (SAINT) incorporates two temporal features, namely answering time and interval time, to enhance self-attention computation. Context-aware attentive knowledge tracing (AKT) [68] combines the self-attention mechanism with psychometric models to achieve better interpretability. These models leverage the attention mechanism to capture relationships and dependencies between learning interactions and improve prediction performance.

Graph-based Knowledge Tracing

Graph-based knowledge tracing (GKT) [70] utilises graph neural networks (GNNs) to incorporate the graph structure of KCs into KT. It conceptualizes the KCs as nodes in a graph, where edges represent relationships between KCs. GKT aggregates and updates the temporal knowledge state based on the aggregated features and the knowledge graph structure. Structure-based knowledge tracing (SKT) extends this approach by capturing multiple relations in the knowledge structure to model influence propagation among concepts. These models leverage the graph structure to capture spatial effects and improve the modelling of knowledge dependencies.

2.5 Theoretical Framework for Learning and Knowledge Construction

his thesis is grounded in key learning and knowledge construction theories that inform the design and development of Intelligent Tutoring Systems (ITS). Understanding these theories is crucial for developing effective TEL systems that align with the cognitive and social aspects of learning.

One foundational paper that informs this work is Sfard's (1998) discussion on the metaphors of learning: the acquisition metaphor and the participation metaphor [71]. The acquisition metaphor conceptualises learning as the process of acquiring knowledge, where knowledge is a commodity that learners obtain and accumulate [72]. This metaphor underpins traditional educational models and is reflected in the way ITS are designed to deliver content and assess learner's knowledge acquisition [73].

In contrast, the participation metaphor views learning as a process of becoming a member of a certain community, emphasising the social and communal aspects of learning [74]. This approach focuses on the learner's engagement and interaction within a learning community, rather than just the accumulation of knowledge [75].

Both metaphors offer valuable insights for the design of ITS and Human-AI collaborative systems. The acquisition metaphor guides the development of systems

that effectively deliver personalised content and assess learners' knowledge state, which is crucial for the intelligent adaptation of ITS [76]. On the other hand, the participation metaphor inspires the design of systems that foster active engagement, collaboration, and community-building among learners [77].

Incorporating both metaphors, this thesis aims to develop ITS that not only personalises learning experiences (acquisition) but also promotes active engagement and interaction between students and AI systems (participation) [78].This dual approach ensures that the ITS designed in this research are not only effective in delivering content and assessing learning but also in engaging learners in a meaningful and collaborative learning process.

The adoption of these learning theories has several implications for the design and development of ITS:

- *Personalisation and Adaptation*: Following the acquisition metaphor, ITS should be designed to adaptively present content based on individual learner's knowledge state, learning style, and preferences [79].

- *Engagement and Interaction*: In line with the participation metaphor, ITS should facilitate interactive and collaborative learning experiences, allowing learners to engage not only with the content but also with peers and the learning community [80].

- *Balanced Approach*: An effective ITS should balance both acquisition and participation aspects, ensuring that learners are not only gaining knowledge but are also actively participating in the learning process [81].

This theoretical grounding provides a comprehensive framework for the development of ITS and Human-AI collaborative systems, ensuring that they are not only technologically advanced but also pedagogically sound and aligned with contemporary learning theories.

CHAPTER 3

A Design Trajectory Map of Human-AI Collaborative Systems: Survey and Taxonomy

Prologue

In Chapter 2, we have provided a basic background of the development of the Human-AI collaborative system. Driven by the algorithmic advancements in reinforcement learning and the increasing number of implementations of human-AI collaboration, Collaborative Reinforcement Learning (CRL) has been receiving growing attention. Despite this recent upsurge, this area is still rarely systematically studied.

In this Chapter, we provide an extensive survey, investigating CRL methods based on both interactive reinforcement learning algorithms and human-AI collaborative frameworks that were proposed in the past decade. We elucidate and discuss via synergistic analysis methods both the growth of the field and the state-of-the-art; we conceptualise the existing frameworks from the perspectives of design patterns, collaborative levels, parties and capabilities, and review interactive methods and algorithmic models. Specifically, we create a new *Human-AI CRL Design Trajectory Map*, as a systematic modelling tool for the selection of existing CRL frameworks, as well as a method of designing new CRL systems, and finally of improving future

CRL designs. Furthermore, we elaborate *generic Human-AI CRL challenges*, providing the research community with a guide towards novel research directions. The aim of this Chapter is to empower researchers with a systematic framework for the design of efficient and 'natural' human-AI collaborative methods, making it possible to work on the maximised realisation of humans' and AI's potentials.

Declaration: This chapter is based on the following publication:

***Li, Z.**, Shi, L., Cristea, A. I., and Zhou, Y. (2021, June). **A survey of collaborative reinforcement learning: interactive methods and design patterns**. In Designing Interactive Systems Conference 2021 (pp. 1579-1590).*

Li, Z., Shi, L., and Zhou, Y. (2021, June). ***A Design Trajectory Map of Human-AI Collaborative Reinforcement Learning Systems: Survey and Taxonomy*** ACM Transactions on Computer-Human Interaction. ***Under Review.***

This chapter is presented largely as accepted, although referencing and notation have been altered and cross-referencing added for consistency across this thesis. Some stylistic changes have been made for consistency. The majority of the text is verbatim, with some minor wording and formatting changes.

3.1 Introduction

With the rapid development of Artificial Intelligence (AI) in recent years, the mainstream media holds two opposing views: AI will 'save the world' [23] or 'destroy' it [24]. AI is described as the 'saviour', to free humans from labour, while it is also described as the 'devil' who takes away workers' jobs [82]. Regardless of one's point of view, AI is playing an increasingly significant part in the future world. *Weak AI*, *strong AI*, and *super AI* are three stages of AI development, as proposed by John Searle [83]. Due to the limitations of current technology, Searle believes that we are supposed to have been and still be in the 'weak AI' stage for a long time. That is, at the current stage, AI often performs much worse than humans in highly

complex decision-making tasks that require consideration of morality and risk, but much better in tasks with well-specified feedback and large-scale data. Therefore, the two extreme situations described by the media are still far from the current stage that we've achieved [23, 24]. Exploring thus the way for humans and AI to better cooperate, with the goal of complementing each other's shortcomings, may provide the best way forward for the immediate future.

A common classification of Artificial Intelligence algorithms is supervised learning, unsupervised learning, and reinforcement learning [84]. Problems involving decision-making generally lie in the field of reinforcement learning [48], and how humans and AI agents can cooperate and complement each other's shortcomings are particularly important. While the interaction between humans and AI agents is an emerging research direction, research on the interaction between humans and computers has a long history. The community has proposed several patterns of human-computer interaction. For example, in 1983, Hollnagel and Woods proposed the Cognitive Systems Engineering (CES) model [25]. In 1991, Schmidt *et al.* created a conceptual paradigm classifying human-computer collaboration into three levels: augmentative, integrative, and debative [26]. In 2009, Johnson *et al.* proposed a co-active design pattern in human-AI joint activity [27].

Over the last few years, collaborative or interactive reinforcement learning has become a new field within the machine learning regime. Especially, at the end of 2022, the birth of ChatGPT [1] brought great shock to the field of AI research. ChatGPT is a large language model developed by OpenAI [2]. It can understand and respond to various types of natural language input, including text and voice, and achieve human-like text output. The reinforcement Learning method is used to fine-tune the ChatGPT model and adapt it to specific tasks, which achieves significant progress. The success of ChatGPT brings a lot of attention to collaborative reinforcement learning. Some excellent recent survey studies on Collaborative Reinforcement Learning (Collaborative RL, or CRL) have emerged, demonstrating this new field's importance. They cover a wide range of issues, including CRL in

[1] https://openai.com/blog/chatgpt/
[2] https://openai.com/

general, such as [85], and CRL applied in specific domains, such as safe RL [86], inverse RL [87], and explainable RL [30]. Other studies concentrate on specific design methodologies such as user feedback and testbeds of the environment [88]. When performing a brief search on Google Scholar with the keywords 'interactive AI', 'collaboration', 'reinforcement learning', and 'HCI' (Human-Computer Interaction) for the period from 2011 to 2022, we found that there are few surveys or literature reviews published. For example, between January 2020 and December 2022, only five surveys or literature reviews were published. Najar and Anis reviewed reinforcement learning based on human advice [28]. Gomez and Randy's work focused on human-centred reinforcement learning [29]. Puiutta and Erick presented a review of explainable reinforcement learning [30]. Arzate and Christian presented a survey on the design principles and open challenges of interactive reinforcement learning [9]. Suran and Shweta consecrated on collective intelligence [89]. It can be observed that this research direction is gaining increasing attention from the community. However, we posit that while these approaches do involve elements of collaboration, there is still room for deeper, more nuanced collaboration where the roles, contributions, and mutual adaptations of both humans and AI agents are more prominently featured. Our definition of collaboration encompasses not only the integration of human inputs but also the co-evolution and reciprocal shaping of human and AI capabilities within these systems. Surveying *collaboration between* humans and AI agents is being overlooked, let alone identifying the probable future direction of growth in this field. To bridge this gap, we thus aim to address the following research question:

How may designers approach the construction process of human-AI collaborative reinforcement learning systems in a structured manner?

To answer this research question, we summarise existing collaboration approaches and offer our own perspectives and proposals. We look at classic human-machine interaction strategies that have had a significant effect on the evolution of the Human-Computer Interaction (HCI) field. We intend to give academics and practitioners with a design toolkit that combines archetypes and specific tools in a micro-view [90] (see Fig. 3.2). Furthermore, our study introduces **Human-AI Collaborative Reinforcement Learning Design Trajectory Map** (see Fig. 3.5), a new cat-

egorisation approach and systematic modelling tool that seeks to suggest research objectives for **the next generation of Human-AI Collaborative Design**. Similar to how builders require the blueprint design as well as instructions on how to plan different functional parts and choose various types of materials for the house, the *Design Trajectory Map* provides readers with a comprehensive review regarding the *design patterns* for Human-AI Collaborative Reinforcement Learning systems (see Section 3.4) and guidance on how to customise the characteristics of different components to meet their specific requirements (see Sections 3.5 and 3.6), as well as how to customise the algorithmic models (see Sections 3.8) and the interactive methods (see Section 3.7).

This study builds on our previous work [91] published at DIS '21: Designing Interactive Systems Conference 2021. In that work, we proposed a Human-AI Design Model that designs a CRL model from three different perspectives: Human, AI agent, and Design pattern, which is a straightforward and effective method (see Fig. 3.1). In subsequent research, we found that in order to build a CRL system from macro to micro, we lacked the considerations of collaborative levels, parties, and capabilities, which are crucial for designing the functions and details of each functional party. Therefore, in this work, we add a new component: Capabilities (see Section 3.6). In addition, based on the original study, we improved the depth and topics covered. We created a more comprehensive framework and taxonomy, covering design patterns through algorithms, and expanded the review with 63 publications, accounting for 43 percent of the new literature. As a result, the primary contributions of this survey are as follows:

1. First, we summarise the most significant **Human-AI Collaborative Design Patterns**, which might help academics and practitioners in the HCI field.

2. Second, we present the **Collaborative Reinforcement Learning (CRL) Design Trajectory Map**, a *novel CRL Classification and Taxonomy*, as a systematic modelling tool to assist researchers in selecting and improving new CRL designs.

3. Third, we take stock and summarise the most recent *Collaborative Reinforce-*

ment Learning algorithms, analysing the state-of-the-art at the start of this new decade.

4. Fourth, as a *roadmap to good Human-AI Collaboration*, we identify several general CRL problems for future study in this field.

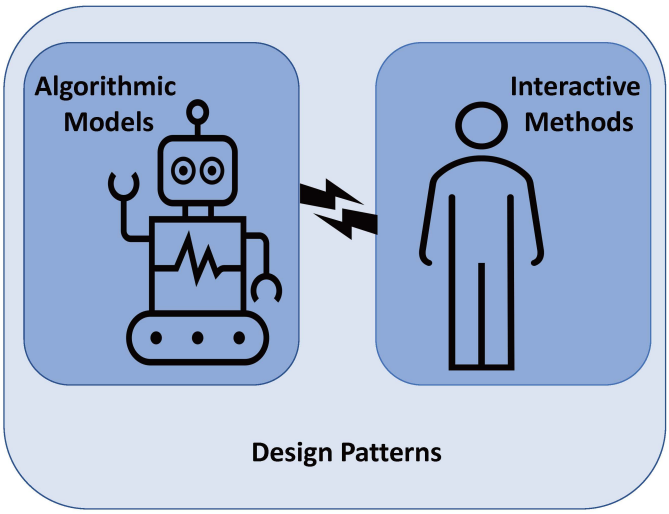


Figure 3.1: Human-AI collaboration Design Model: From a human perspective, we focus on how humans interact with AI agents; from an AI agent's perspective, we focus on how AI agents accept human instructions or suggestions in algorithm implementation; and from a collaboration pattern perspective, we focus on what kind of way that humans and AI collaborate.

3.2 Background

Reinforcement Learning (RL) is derived from theories of *animal learning* and *parameter disturbance adaptive control* [9]. The intuition is that if an agent's actions results in positive rewards (reinforcing signals), this type of behaviour will be reinforced, increasing the agent's inclination to repeat the behaviour in future acts. RL aims to train the agent to find the optimal strategy for each discrete state while maximising the expected discount rewards [48]. Mathematically, a reinforcement learning process can be described as a Markov Decision Process (MDP), defined by the tuple $\mathcal{M} =< \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma >$, which is a cyclical process, where an agent takes action $\mathcal{A}$ to change its state $\mathcal{S}$ to obtain a reward $\mathcal{R}$ from the process of interacting

with the environment; γ is the discount factor; $\mathcal{T} : S \times A \mapsto \Pr[S]$ is the transition function; the expected long-term reward follows policy π, represented as the Q-function $Q^{\pi}(s, a)$, which is computed as:

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s'} \mathcal{T}(s,a,s') \sum_{a'} \pi(s',a')\, Q^{\pi}(s',a') \tag{3.1}$$

$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a)$ is an optimal value function. Any optimal policy π^{*} that maximises the expected reward for each state is the solution to the MDP.

Reinforcement Learning differs from the other two types of machine learning (Supervised Learning and Unsupervised Learning). In **Supervised Learning**, a model learns the mapping relationship between input X and label y through a set of paired and labelled data to solve *Regression* and *Classification* problems. In **Unsupervised Learning**, a model learns unlabelled data without any guidance, to solve *Association* and *Clustering* problems, for discovering underlying patterns of the data. In **Reinforcement Learning**, a model learns the mapping relationship between *states* and *actions* (non-predefined data), to solve *Exploitation* or *Exploration* problems. The mapping directs the model, or the agent, to make optimal decisions based on these *states* towards *maximising cumulative rewards* [48]. The learning process emphasises the interactions between the *agent* and the *environment* that gives 'reward signals' during the agent's continual or exhaustive attempts of all the possible strategies to be adopted in a particular 'state', rather than directing the agent how to create the 'correct' action [92]. A 'reward signal' is usually a *scalar signal* and an assessment of the *quality* of the generated action by the agent. This way, the *agent* learns knowledge from the *environment* through *discrete feedback* of *actions*, which it uses to optimise the parameters that might lead to an optimal result. With minimal information given by the external environment, the *agent* must learn by its own interactions with the *environment*, frequently from the ground up. If the 'reward signal' r and the 'action' A were known, these corresponding representation-label data might be utilised to train a model using supervised learning. However, it is often impracticable to exhaust all conceivable *actions* in an *environment* and cre-

ate the corresponding '*reward signals*'. This is where Reinforcement Learning may help. Reinforcement Learning often beats *Supervised Learning* in scenarios where the discrete action space is small, such as the game of Go or Atari [93].

Bellman proposed the mathematical theory of dynamic programming in 1955 [94]. The 'Bellman condition' was considered to be the crucial theoretical foundation for reinforcement learning. Then, in 1957, Bellman proposed Markov Decision Processes (MDPs) [95], which are now used in most reinforcement learning algorithms. After the 1960s, the concepts of reinforcement and reinforcement learning gradually appeared in the literature. In 1963, a system called STeLLA was developed, which allows trial and error learning through interactions with the environment [96]. In 1975, Holland proposed an adaptive system based on the selection principle in his book 'Adaptation in Natural and Artificial Systems' [97]. Michie proposed an early reinforcement learning system called MENACE in 1995 [98]. This is regarded as one of the most significant events in the evolution of reinforcement learning. The book also included genetic algorithms, which aided in the development of optimisation algorithms.

Rummery and Niranjan suggested SARSA, a.k.a. state-action-reward-state-action, in 1994 on the basis of Q-learning. In terms of decision-making, SARSA is similar to Q-learning. However, it differs in terms of the updating method. SARSA employs an on-policy method, whereas Q-learning employs an off-policy one [99]. Thrun *et al.* introduced the Monte Carlo positioning method in 1999, which uses probability to solve the robot positioning problem [100]. Compared to traditional grid methods, it is more efficient and saves memory [99].

With the advancement of computing power and the advancement of deep learning, numerous approaches combining deep learning with reinforcement learning have lately been presented. In 2013, Mnih *et al.*, from the Deep Mind team, proposed Deep Q-Learning (DQL) [101]. This approach employs Q-learning to discover the appropriate control rules after transferring data from high-dimensional sensory input to a convolutional neural network to extract features. This team's AlphaGo defeated the world Go champion with a score of 3:0 in 2017. In December of the same year, the more advanced Alpha Zero achieved AlphaGo master by self-learning without

the assistance of human knowledge in only 21 days, and exceeded all versions after 40 days [102]. Since then, *Reinforcement Learning* has made great progress [84]. In 2021, Chen *et al.* proposed a method that transforms reinforcement learning into a sequence modelling problem called the decision transformer model [13]. This has gradually been applied in fields such as games [103], robotics [104], computer vision [105], natural language processing (NLP) [106], and recommender systems [107]. For example, the Open AI[3] team created an interactive reinforcement learning method that used human feedback, to learn summarisation [108]. Another recent project proposed by this team, GPT-3, has also made revolutionary achievements in the field of NLP [106]. At the end of 2022, ChaGPT, powered by the GPT-3 engine, drew lots of attention for its amazing performance in various natural language processing tasks, such as text completion, language translation, and question answering.

Due to its strong potential and firm theoretical foundation, Reinforcement Learning has recently been one of the most attractive research areas in AI technologies [48]. However, it faces many challenges. Currently, on the one hand, Reinforcement Learning only works well when the environment is definite, i.e., the state of the environment is fully observable. In particular, there are defined rules in games like Go, and the action space is discrete and constrained. In other words, the agent needs a great degree of prior knowledge, to understand its state in a complex environment [109]. On the other hand, even if the agent has been given well-specified feedback, the inexplicability and incomprehensibility caused by the agent's unconscious are still inadequate for the agent to decide on the precise next action [9]. Furthermore, most applications of Reinforcement Learning to date have only been for playing games, such as chess and Atari.

[3]OpenAI website: www.openai.com

3.3 Methodology and Scope

3.3.1 Literature Collection

This review focuses on hitherto undiscovered research areas on collaborations between humans and AI agents. We further refine our data pool, specifically narrowing the selection to the following selected target areas: *HCI*, *Human-AI Collaboration*, *Reinforcement Learning*, and *Explainable AI*, published in the recent decade, the time period between 2011 and 2022, from Google Scholar. In total, our search yielded 257 articles using keywords including *collaborative reinforcement learning*, *interactive reinforcement learning*, *human-computer interaction*, and *design patterns*. They were published in journals and conferences, including top venues such as TOCHI[4], IJHCS[5], AAAI[6], CHI[7], UbiComp[8], UIST[9], and IEEE[10]. Following a manual review of the title and abstract of each article, we eliminated 112 as irrelevant, leaving 145 as the source of this survey.

3.3.2 Human-AI Collaborative Reinforcement Learning Classification

We used an *inductive method* to organise the literature we collected and proposed a new classification method inspired by traditional human-machine interaction research. Previous work by Najar and Anis mainly targeted physical interaction between humans and machines from a human perspective [28]. In the early stages of computer science and engineering, no concept of AI was involved. So far, this is the only research on human-machine interaction. Cruz *et al.* proposed that the human-AI interaction is a kind of human-machine interaction, in general, [110]. In this Chapter, we have collected paradigms of human-machine interaction in the early

[4]The website of TOCHI: https://dl.acm.org/journal/tochi
[5]The website of IJHCS: http://dblp.uni-trier.de/db/journals/ijmms/
[6]The website of AAAI: www.aaai.org/
[7]The website of CHI: chi2021.acm.org/
[8]The website of UbiComp: www.ubicomp.org/
[9]The website of UIST: uist.acm.org/
[10]The website of IEEE:www.ieee.org/

Table 3.1: *CRL Classification* applied to the Pool of Papers about Collaborative RL, published between 2011-2022

	Collaborative RL	References
Design Patterns	Cognitive Systems Engineering Patterns	[25]
	Bosch and Bronkhorst's Patterns	[111]
	Coactive Design Patterns	[112]
	Schmidt's Patterns	[113]
Collaborative Levels and Parties	Augmentative Level collaboration	[114], [93], [115], [116], [117], [118], ChatGPT [11]
	Integrative Level collaboration	[112], [113], [119], [120], [121], [122], [26]
	Debative Level collaboration	[26], [123], [124], [86], [125], [9], [126]
	Collaborative types	[127], [128], [129], [130]
Collaborative Capabilities	Understanding	[131], [132], [57], [133], [134], [135], [136], [137]
	Communication	[128], [138], [139], [140], [141], [109], [142]
	Commitments	[143], [144] [145], [143], [146], [147], [148], [149], [150]
	Institutions	[151], [152], [153], [154], [155], [156], [157], [158], [159]
Interactive Methods	Explicit Methods	[160], [161], [162], [163], [164], [160], [165], [110], [166]
	Implicit Methods	[167], [114], [168], [169], [170], [171], [172], [173], [174], [175], [176], [177], [178]
	Multi-model Methods	[179], [180], [181], [182], [183], [184], [185], [186], [187]
Algorithmic Models	Reward-based methods	[188], [160], [189], [170], [171], [172], [173]
	Policy-based Methods	[190], [162], [126], [171], [172]
	Value Function based methods	[191], [192], [101], [102], [114]
	Exploration-process methods	[164], [193], [173], [174], [175], [112], [26]

stage, which also could apply to human-AI interaction. Therefore, in the following section, we will use the concept of human-AI interaction in a unified manner. In the work of Arzate and Chirstian, more attention was paid to the algorithmic model of the AI agent [9]. After reviewing the literature on human-machine interaction in traditional engineering, we found that the interaction between humans and AI corresponds to these design patterns. In particular, Schmidt's model [26] not only combines interactive methods and algorithmic models but also provides different design ideas, according to different human-AI collaborative levels. Based on the common characteristics of the classification in this literature, we derive the new *Human-AI Collaborative Reinforcement Learning (CRL) Classification* (see Table 3.1).

3.3.3 Human-AI Collaborative Reinforcement Learning Taxonomy

We incorporate past work in a novel way to create a new taxonomy. We draw on Schmidt's Machine Interaction Pattern [26], Dafoe's Collaboration Parties Classification Model [128], and Arzate's Algorithmic Classification Model Collaboration Method [9], to generate a novel taxonomy method from coarse to fine granularity. Based on this approach and populating them with representative works from the literature, for a structured approach (see Table 3.1), we define five axes: *Design*

Patterns, *Collaborative Levels and Parties*, *Collaborative Capabilities*, *Interactive Methods*, and *Algorithmic Models*. These five axes are then used to create a taxonomy, as shown in Fig. 3.2, which might be used as a systematic modelling tool for HCI researchers and practitioners to select and improve their new CRL designs.

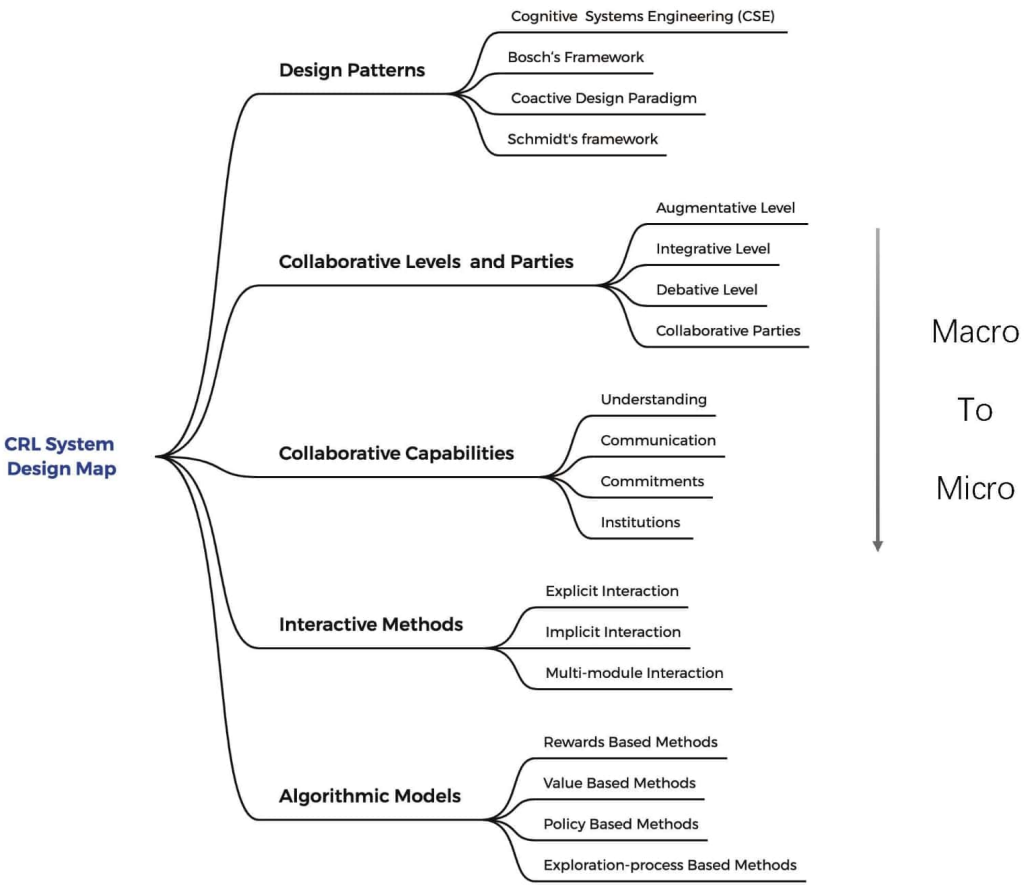


Figure 3.2: A new *CRL taxonomy* for interactive methods and design patterns.

3.4 Human-AI Collaborative Design Patterns

Human-AI collaborative design patterns may be used to provide an efficient and repeatable approach for building human-AI collaborative systems [194]. Reliable design patterns might increase these systems' quality, reusability, and maintainability. In this article, we collect the most recognised design patterns in the literature of

human-machine collaboration for academics and practitioners to populate the CRL Taxonomy as shown in Fig. 3.2.

In comparison to human-AI collaboration, human-machine collaboration has long been a source of concern for researchers. In the early stages of the development of human-machine interactions, domain experts believed that the collaboration between humans and machines was a physical, lower-level type of collaboration [26]. Specifically, machines were used by humans exclusively through physical contact, in the lack of feedback between machines and humans, which might be viewed as a kind of *unidirectional interaction*. After decades of development, several recognised human-machine collaboration design patterns have been created, which we summarise below.

3.4.1 CSE Pattern

Cognitive Systems Engineering (CSE), coined by Hollnagel and Woods, acts at the level of cognitive functions [25]. CSE is the first framework proposed to analyse the human-machine information exchange interaction. CSE is a framework for human-machine collaboration where machines 'plan and explore' using the knowledge or information provided by humans. This engineering method suggests that human-machine collaboration occurs at a conscious level of communication. It is a perceptual mode in which the machine is employed as a sensory extension to assist with human activities.

The major challenge at this level is identifying appropriate interactive methods to optimise human information processing. However, CSE has been constrained because it has only explored basic and low-level communications, leaving out more complicated problems and environments.

3.4.2 Bosch and Bronkhorst's Pattern

Bosch and Bronkhorst defined three levels of Human-AI collaboration: 1) *unidirectional interaction*, in which humans assist machines or machines explain themselves to humans; 2) *bi-directional interaction*, and 3) *collaboration* between humans and

machines [111]. The vast majority of the currently existing methods have only addressed the first level.

This framework's contribution provides a viewpoint on the directions of collaboration between humans and machines. Furthermore, it constructs the direction based on the roles that humans and machines play in a task, with one being the subject of a task and the other aiding the opposing side. It is believed to help in the development of more efficient communication methods. For example, if it is a human-centred framework, more consideration should be given to how to transform a 'machine language' into interpretable information such that humans can better understand, whereas if it is a machine-centred framework, more consideration should be given to how human knowledge could improve machine efficiency.

3.4.3 Coactive Design Pattern

Johnson *et al.* proposed a Coactive design pattern in human-AI joint activities. They experimented with a collaboration system from the perspective of observability, predictability, and directability [112]. *Observability* concerns the ability of both robots (or AI agents) and humans to observe each other's pertinent aspects of status and knowledge of the team, tasks, and environment. *Predictability* refers to the state that the actions of both robots (or AI agents) and humans can be predicted such that they may rely on each other's actions to perform their own actions. *Directability* refers to the ability of both robots (or AI agents) and humans to direct each other's behaviour in a complimentary manner.

This framework is similar to Bosch and Bronkhorst's framework in that it considers the direction of interactions and divides it into different levels. However, it is limited due to a lack of robustness and security considerations.

3.4.4 Schmidt's Pattern

Schmidt believes that collaboration should be tailored to diverse needs, fulfil different functions, and be carried out in various ways depending on the circumstances. Collaboration may be summarised as follows: 1) the augmentative level, in which

one role in the partnership (human or AI agent) assists the other in performing tasks; 2) the integrative level, in which both sides of the team share information and assist each other in completing tasks together; and 3) the debative level, in which tasks are completed through debate and negotiation between humans and AI agents, especially when dealing with complex issues [113].

This framework considers not only the information exchange direction but also different levels of collaboration, as well as robustness, security, and potential ethical considerations.

3.5 Collaborative Levels and Parties

The patterns described above frame the modes of human-AI collaboration from different perspectives. They are also very comparable in terms of compartmentalising collaboration modes or methods, i.e., into single-direction assistance, bi-directional collaboration, and higher-stage fused collaboration. In this survey, we utilise a fusion viewpoint that combines interactive methods and design patterns based on Schmidt's collaboration pattern to classify the current collaborative reinforcement learning techniques. Schmidt's model is divided into three levels: augmentative, integrative, and debative. We drew a pyramid model based on Schmidt's model (see Fig. 3.4). We highlight significant research that has emerged at each level and the issues that should be examined in the first three subsections that follow. We also discuss the characteristics, advantages, and disadvantages of these different methods and how to develop new ones in the future.

Apart from the classification of collaborative levels where humans and AI agents are both viewed as a whole, we also discuss collaboration from a *micro perspective* based on the framework proposed by Dafoe *et al.*, where diverse constellations of humans and AI agents are discussed, which we refer to as "collaborative parties" in the final sub-section.

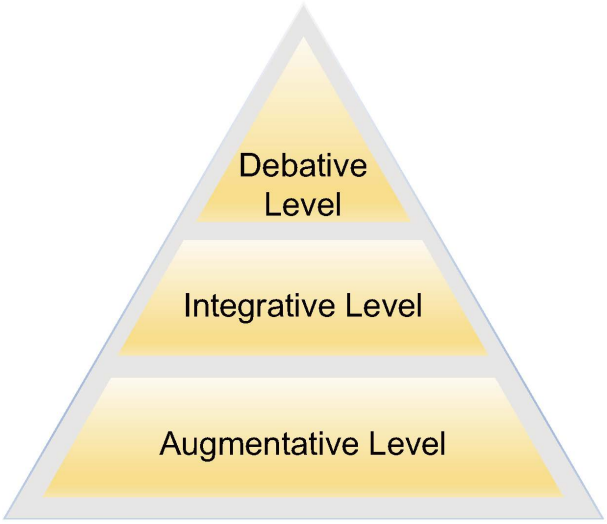


Figure 3.3: Triangle of different collaborative levels: the first level is Augmentative Level collaboration; the second level is Integrative Level collaboration; and the third level is Debative Level collaboration.

3.5.1 Augmentative Level collaboration

collaboration at the *Augmentative Level* entails one partner compensating for the shortcomings of the other [26]. AI has shown considerable promise in large-scale data processing with well-defined rules, as well as in natural-perception fields such as digital image recognition [105], natural language processing [195], among others. Nevertheless, in a complex, ambiguous environment, AI performance lags considerably below that of humans. The *Augmentative Level* approaches offered by the community are mostly divided into two types. First, AI takes the lead in decision-making, while humans assist AI in enhancing processing efficiency. In this case, humans use prior knowledge to help the agents specify the state space and efficiently obtain rewards from the complex environment. The RL methods used in ChatGPT belong to this kind of collaboration. In the case of ChatGPT, it uses human feedback to improve the model's performance over time. For example, a human could evaluate the responses generated by the model and provide feedback on their quality. This feedback could then be used to update the model's parameters, allowing it to learn from its mistakes and generate better responses in the future

[12]. Second, humans play the primary role in decision-making, with AI assisting in the process. In this case, the AI agents explain the tactics used to help humans make faster decisions in a simple environment. At the sub-level of *humans helping AI agents* improve efficiency, we will categorise how they communicate based on which parts of the algorithm humans' help can be injected into. At the sub-level of *AI agents helping humans*, we mainly focus on how AI agents may inform humans about why they make particular decisions.

Human-AI

The essential aspect of the role of humans supporting AI agents in decision-making has been how to efficiently deliver information to AI agents while reducing human fatigue to a minimum. Up to this point, many human-AI collaborative reinforcement learning algorithms have been proposed, which may be categorised into *explicit interaction modal*, *implicit interaction modal*, and *multi-modal methods* based on different forms of interactions (detailed in Section 3.4). Finding a better way for humans to interact directly with AI agents is still an essential research priority.

AI-Human

In the task of AI agents assisting humans in decision-making, the most challenging problem lies in interpretability. Interpretability refers to the degree to which humans can understand the rationale underpinning machines' decision-making [196]. The interpretability of AI models refers to the clarification of the internal mechanism and the understanding of the results. The more interpretable the model is, the easier it is for people to trust it [30]. Its significance may be seen in the following aspects. In the *modelling phase*, interpretability may assist developers in understanding the learning process, comparing alternative algorithms, optimising the procedure, and fine-tuning the models; in the *operation phase*, AI agents can explain the internal mechanisms and interpret the model outcomes to the decision-maker (i.e., humans). Consider a decision-making recommendation model: before the model runs, multiple

[12]https://openai.com/blog/chatgpt/

interpretable algorithms with their respective advantages can be provided to the decision-maker to choose from, and after the model is trained, the model must explain to the decision-maker why it recommended a specific solution given a specific context.

Patterns underlying the above problems lie under the umbrella of eXplainable AI (XAI), which is commonly regarded as critical for the practical deployment of AI models. DARPA launched XAI in 2016 [93]. The basic objective of XAI is to create machine learning models that, when combined with proper explanation techniques, will allow humans to comprehend better and eventually accept and trust the model's predictions. The literature generally proposes two types of explainability: 1) *transparent models*, which are embedded inside the operation of the AI algorithms, leading to explainability by design, and are applied to simpler AI algorithms with less accurate results; and 2) *post-hoc models*, which are performed after initial models have been trained. This type of method is usually more efficient, but it is less reliable than *transparent models* [115].

At present, there are a few intrinsic interpretability Reinforcement Learning methods. Verma *et al.* introduced a Programmatically Interpretable Reinforcement Learning method (PIRL) [116]. This method is an upgrade of traditional Deep Reinforcement Learning (DRL). In DRL, it is difficult to represent policies due to the nature of 'black-box' neural networks. To tackle this challenge, PIRL introduces an advanced human-readable programming language to define neural network policies. Shu *et al.* introduced a hierarchical and interpretable multi-task reinforcement learning framework where a complex task is broken into several sub-tasks, and then a hierarchical strategy is used to complete learning with 'weak supervision' from humans. By breaking a task into sub-tasks and thus making a learned strategy traceable to them and explaining the relationship between different hierarchies of the sub-tasks, this method builds intrinsic interpretability.

Compared with intrinsically interpretable Reinforcement Learning methods, post-hoc methods are simpler in algorithm structure and more efficient in the computing process. At present, many post-hoc methods have been proposed. For example, Liu *et al.* proposed an explainable DRL method based on linear model U-trees [117].

This is a stochastic gradient descent framework for explaining complex models by using linear model U-trees to fit Q-functions. There is also a Soft Decision Tree (SDT) method, which provides post-hoc explanations by extracting policies. Madumal *et al.* introduced an explainable method through a causal lens. In this framework, an AI agent learns to play StarCraft II, a large dynamic space strategy game [118]. To generate an explanation, they simplify the entire game to four basic actions and nine basic states and then use these basic causal factors to construct an explanation for why the AI agent chooses action A over action B.

3.5.2 Integrative Level collaboration

Integrative collaboration entails using the various advantages of both parties to complete a task. At this level, humans and AI agents are regarded as interdependent. The main task is broken into several sub-tasks, and humans and AI agents can perform just those they are skilled at [26]. At the integrative level, humans and AI agents play equal roles in the system. Information exchange at this level is generally referred to as 'communication' in the literature [26].

In the following sub-sections, first, we summarise the communication methods in this collaborative pattern. Then, we discuss how to make the communicating parties trust each other. On this basis, the system needs resilience to enhance its robustness in order to better deal with the complex conditions in the real world.

Communication

A grand challenge of collaborative reinforcement learning is how humans and AI agents communicate with each other. Only when communication is seamless can they decide on the next actions following each other's feedback. Liang *et al.* proposed an implicit human-AI collaboration framework based on Gricean conversational theory [197] to play the game Hanabi. The AI agent must cooperate with the human to win the game. In this framework, the AI agent tries to understand the implied meaning of human's natural language suggestions in a dialogue box [122].

Cordona-Rivera and Young proposed an AI Planning-based Gameplay Discourse Generation framework to achieve communication between human players and the

game [121]. Pablo and Markus proposed an approach to Human-AI collaboration by planning and recognising the plan [120]. Johnson *et al.* proposed a testbed for joint activities. The unique feature of this testbed is that it can be applied not only in interactive experiments for multiple agents but also in interactive experiments between humans and agents [27]. A series of studies were carried out on this testbed to investigate the collaboration of humans and agents in a team. For example, Matthew *et al.* introduced the relationship between interdependence and autonomy in a human-AI collaboration system [119].

Trust

Based on the established communications, figuring out how to make the partners trust each other to complete the task is also crucial. Although the community has not yet proposed a clear definition of trust between humans and AI agents, it is generally regarded as a psychological state [26]. Johnson *et al.* proposed a Coactive Design framework for human-AI joint activities. In their framework, the authors proposed a collaboration system following the perspectives of observability, predictability, and directability [112]. These components are critical for humans and AI agents to collaborate in a trustworthy manner.

Resilience

Resilience is another essential feature in human-AI collaboration. In complex problems, with possible delays and information noise, establishing a resilient mechanism to make the system more robust is crucial for communication and mutual trust. An effective human-machine collaboration mechanism should be able to diagnose a problem quickly and provide remedial explanations after the problem occurs so that the system can get back on course [112]. Zieba *et al.* proposed a mechanism to measure the resilience of human-machine systems, that is, the ability to anticipate, avoid, and recover from accidents to a normal state [113]. This is instructive for designing a collaborative system, as it is necessary to consider how it responds to emergencies and thus recovers quickly.

3.5.3 Debative Level collaboration

Debative models come into play when humans and AI agents hold different opinions on decision-making in a task and debate to find the optimum solution based on their differing knowledge and understandings. Models are often required to meet the following requirements. First, humans and AI agents share a unified goal, and achieving that goal is the primary task. A debate without a unified goal is meaningless for both parties. Second, both parties have strong justifications for their decisions and have insights into a problem based on their respective cognitive models. Third, both parties can effectively communicate and explain their decisions to each other. Communication and interpretability are the premises of the debate. Fourth, there are clear evaluation criteria to measure the outcome from a debate to ensure an optimal result. Fifth, both parties can learn and adjust their own knowledge after a debate to achieve better results in the future [26].

As knowledge-based decisions are fragile and controversial, it is necessary to debate the results [123]. In a complex and uncertain environment, a full debate will better demonstrate the advantages and disadvantages of different decisions. Collaboration at the level of debate requires that both humans and AI agents have sufficiently high levels of professionalism in a specific complex domain. Reinforcement learning algorithms based on this level are scarcely studied in the literature, but we expect that as the field progresses, this form of collaboration will attract more attention.

Geoffrey *et al.* introduced a framework that enables two agents to debate with each other, with a human judge deciding who to trust in the end [124]. Although it has not yet been applied to the debate between humans and agents, this framework meets the above requirements. In their experiment, the two agents attempted to persuade human judges to believe their judgments on the MNIST data [198]. First, the goals of the two agents were unified. Second, the two agents have different judgments based on their own algorithmic perceptions. Third, both agents are able to generate simple explanations to persuade human judges. Fourth, human judges have the intuitive knowledge to make accurate judgments. This experiment is enlightening for future research, especially in human-agent and multi-agent debating

collaboration.

3.5.4 Collaborative Parties

The different levels of human-AI collaboration take a macro view of humans and AI, looking at them as a whole. The collaboration of humans with AI, on the other hand, can be split down into different combinations of parties from a micro perspective or in consideration of practical scenarios. For example, future scenarios could include interactions between a human and several AI agents, interactions between human groups and AI-agent groups, or a more diversified fusion of the two. Therefore, in this section, we will discuss the types of interactions between humans and AI agents from the micro perspective (AI agents-agents, human-AI agents, human-human, and more complex constellations). Dafoe *et al.* categorises collaborative roles into three categories: AI-agent, Human and Organisations. Collaborative types based on the number of roles involved in the collaboration into six types [128] (see Fig. 3.4):

1. Human-Human collaboration: the classic human-to-human collaborative model;
2. Collaborative Tools: the AI agent is used to enhance collaboration, such as language translation.
3. Alignment and Safety: the AI agent acts like an assistant to help humans solve problems, such as the relationship between vehicles and humans in autonomous driving.
4. Human-AI-Human-AI collaboration: With the development of 5G technology and AI technology, large-scale human groups and AI groups may cooperate in the foreseeable future, such as in level 5 autonomous driving [127].
5. The Planner Perspective: This approach aims to strengthen the collaboration and infrastructure of the entire society from the planner perspective of social construction rather than the collaboration between an individual and a single AI, e.g., social media and network communications.

6. Organisations and Society: Collaboration could have a more complicated structure, with multiple types of hierarchical collaboration or a complex internal structure.

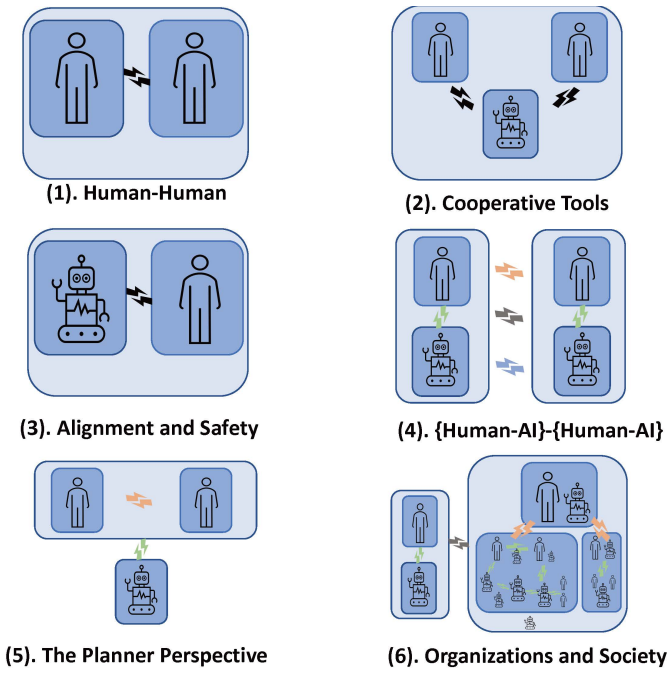


Figure 3.4: Different parties in the process of collaboration.

3.6 Collaborative Capabilities

In the previous section, we discussed the different levels of human-AI collaboration from a macro perspective, where both human and AI agents are viewed as integral parts. However, in real scenarios or from a micro perspective, there may be interactions between a human and several AI agents or interactions between human groups and AI groups. Therefore, in this section, we will discuss the types of human-AI interactions from the micro perspective (i.e., AI agents-AI agents, human-AI agents, human-human, and more complex constellations), as well as what kinds of collaboration capabilities the agent requires for group interactions.

Dafoe *et al.* divides the collaboration capabilities of agents into four types: Understanding, Communication, Commitments, and Institutions [128].

3.6.1 Understanding

In human-AI Collaboration, an AI agent's ability to understand the environment and predict the consequences of its actions is crucial for reaching mutually beneficial results. In game theory, there are many discussions about how important it is to understand multi-role collaboration. For example, in Nash equilibrium, each strategy is required to be the best response after fully understanding the other's strategies [131]. Moreover, under the constraint of partial information, Bayesian Nash Equilibrium and Perfect Bayesian Equilibrium provide a solution for how multi-role collaboration should enable other participants to understand better each other's strategies [199].

In collaborative reinforcement learning, the most important type of understanding is learning the preferences of the other AI agents, namely their values, goals, and reward functions. Humans understand how to better provide feedback or rewards to the AI agent in order to help it converge faster and function more efficiently. Some AI researchers attempted to learn the AI agent's behaviour directly. For example, Albrecht's study [132] summarised how humans observe the AI agent's behaviour in order to understand the agent. Inverse Reinforcement Learning, or IRL, is a type of research in which AI agents are oblivious to or indifferent to humans [57]. This type of method requires humans to inject their prior knowledge [133] or control the AI agent to make a few first steps [134, 135].

Besides, in a more complex decision-making environment, humans may consciously or unconsciously hide their opinions or ideas, causing significant challenges for AI agents. There have been some studies on the application of human recursive mind-reading methods in negotiations to overcome this challenge [136], and some studies have applied this method to the game Hanabi to improve the collaboration between humans and AI agents [137].

3.6.2 Communication

Understanding and collaboration could be difficult to achieve without effective communication. AI agents may often better understand others' behaviour, intentions, and preferences by communicating directly with them rather than just observing and

interacting with them regularly. The finding of Pareto-optimal equilibrium may be made easier as a result of information exchange [128].

Common Ground Common ground is necessary for collaboration. The message sender and receiver should use the same communication protocol so that each may understand the meaning of the other's message.

Many studies have been conducted on machine-to-machine communication problems, which are usually referred to as emergency communication [138–141]. However, there are few studies on establishing common ground and effective communication between humans and AI agents. The establishment of common ground is arguably the most difficult challenge [128].

Bandwidth and Latency The bandwidth of communication refers to the volume of data that may be transferred in a given duration of time [128]. Latency refers to the time it takes for a message to be transmitted and received [128]. How to enhance bandwidth and minimise latency in human-AI collaboration has long been studied, and some promising techniques have been proposed, including brain-computer interface technologies, which are designed to connect the human brain directly via hardware to achieve maximum bandwidth with the shortest possible latency [109].

3.6.3 Commitments

The aforementioned capabilities of Understanding and Communication strive to overcome the difficulties in collaboration caused by inaccurate or inadequate information. Collaboration, even with abundant information, may still fail. Social scientists have identified "commitment issues", or the inability to make credible threats or promises, as a primary cause of collaboration failure. Prominent research even claims that the problem of commitment is the most significant impediment to rational AI agent collaboration [143]. A substantial volume of research has explored the commitment issues that affect collaboration [143–147].

Many different studies have tried to build commitments between humans and AI. Some studies have attempted to develop a commitment contract method between

humans and AI agents based on semantics [148–150].

3.6.4 Institutions

Obtaining the requisite understanding, communication, and commitment for collaboration often necessitates the use of a social framework. In economics and politics, this social framework is typically referred to abstractly as *institutions* [128].

Decentralised Institutions In decentralised institutions, there is no single institutions centre, and each individual is connected to the other, continuously encouraging the construction of the structure through interaction with each other. Additionally, many multi-agent system construction methods have been proposed to aid in multi-agent systems communication, planning, and decision-making [151–155].

Centralised Institutions Centralised institutions involve a centralised authority that can define the rules and limit other participants [128]. The multi-agent systems research community attempts to build a collaborative mechanism amongst agents using approaches based on centralised institutions [200]. Several studies investigated the use of centralised multi-agent systems in automatic auction systems [156–158]. The method of centralised multi-agent path-finding technique could be utilised in autonomous vehicle obstacle detection in the future [159].

3.6.5 Distributed Cognition

In the process of collaboration between multiple parties, Distributed Cognition is a very important concept. Distributed Cognition, a theory developed by Edwin Hutchins, extends the concept of cognition beyond the individual to encompass the environment, artefacts, and social interactions. In the context of Human-AI Collaborative Reinforcement Learning (CRL), this theory sheds light on the shared cognitive landscape between humans and AI agents. It encourages exploring how tasks such as problem-solving and decision-making are collaboratively managed, focusing on the synergistic flow of information and the role of digital tools in facilitating cognitive processes. Particularly pertinent is the allocation of cognitive tasks,

where a balance is struck between human intuition and AI's computational prowess. This perspective not only augments our understanding of collaborative dynamics but also informs the design of CRL systems. It underscores the need for systems that optimise cognitive task distribution, ensuring efficiency and user satisfaction while maintaining transparency to foster trust and informed decision-making [201].

3.6.6 Activity Theories

Activity Theory, rooted in the works of Soviet psychologists such as Lev Vygotsky, provides a perspective through which to view human-AI interactions within their socio-cultural context. In Human-AI CRL, the theory emphasises the roles and responsibilities of both human and AI agents, their shared and individual objectives, and the influence of the broader social and organisational environment. It advocates reflecting on how AI systems are integrated into human activities, shaping and being shaped by social norms, regulations, and cultural aspects. This theory underlines the importance of user-centred design, advocating for AI systems that are intuitive and augment human capabilities while also considering the broader social and ethical ramifications. This approach is vital in addressing challenges like job displacement, privacy, and fairness, ensuring that AI systems contribute positively to the human social and work environment [201].

3.7 Interactive Methods

Traditional reinforcement learning methods require excessive training time in complex environments, and their applications are often confined to scenarios with clear rules. An effective way to mitigate these limits is by using the different strengths of humans and AI and complementing each other's inadequacies. This approach is known as Collaborative Reinforcement Learning (CRL). CRL employs human-in-the-loop training to improve the performance of algorithms or to help humans improve decision-making efficiency [9]. Recent CRL research has focused on developing AI that can communicate with humans in a more natural way [9]. There are two types of interactive methods: explicit and implicit. In the explicit method, hu-

mans explicitly provide the AI agents with clear numerical feedback. This method is preferable for AI agents since it allows them to process the feedback more easily, but it is likely to cause human fatigue due to the ambiguity of numerical representations, resulting in inefficiency in a long-term training process. In an implicit method, humans give feedback to AI agents through natural interactions such as posture and gaze, as opposed to explicit methods, which provide clear numerical feedback. This method places more demands on the AI agent, but it may improve the fatigue resistance of human trainers, allowing for long-term and stable collaboration [181]. Based on these unsolved problems, in this section, we present human-AI interactions from the perspective of interactive methods.

3.7.1 Explicit Interactive Methods

Currently, most AI agents learn from human feedback via explicit interactive methods. Humans provide feedback directly to the AI agent via keyboard, slider bar, or mouse to provide clear alpha-numerical feedback [160–163]. For example, Thomaz and Breazeal proposed a method of sending feedback to the AI agent by using the mouse to click on the sliding bar [164]. Knox and Stone proposed the TAMER framework, which allows an AI agent to learn from MDP and human advice by having a human trainer click the mouse to indicate the desired actions [160]. These methods are more efficient than traditional reinforcement learning and can achieve specific goals in complex environments with the assistance of humans.

However, the reaction time of human trainers may cause delayed feedback, leaving the AI agent unsure of which actions the human feedback was aimed at, especially for AI agents with frequent actions. A standard solution is to set a delay parameter to express past time steps. For example, Warnell *et al.* proposed a method to obtain the delay distributions of the human trainers to improve algorithm efficiency [165]. Knox and Stone provided another way for estimating the delay: using a probability density function [202]. Moreover, these methods may be unfavourable to non-professional human trainers, who need to spend a significant amount of time learning the user interface and the meaning of the feedback represented by each operation. Simultaneously, this kind of interaction can easily make human trainers

impatient.

Human trainers can also provide explicit feedback to AI agents using hardware delivery methods [110], where feedback is generally converted into a numeric value directly via the hardware devices, such as keyboards. However, a more user-friendly method allows the AI agent to learn implicit feedback from natural interactions with trainers.

3.7.2 Implicit Interactive Methods

Aside from receiving feedback directly from human trainers via explicit interactive methods, AI agents can also learn via implicit interactive methods.

Implicit interaction methods reduce the learning cost of human trainers, as they can directly participate in training the AI agents without specific learning. At the same time, a more natural way of interaction may reduce the fatigue of human trainers. Many implicit interactive methods have lately been proposed. For example, feedback can be based on natural language, facial expressions, emotions, gestures, and actions, as well as incorporating multiple natural interactive methods. In an ideal scenario, humans could train the AI agent in the same way they interact with humans in the real world. Below, we summarise some of the most prominent implicit interactive methods.

Gestural Feedback. Gestures are sometimes considered a form of unconscious human communication. It is also considered an effective way to complement other communication forms, and it is even more helpful than other communication methods for speech- or hearing-impaired users. For example, Voyles and Khosla proposed a framework which can train robots by imitating human gestures [167]. Moon *et al.* introduced a method of using gestures to command the AI agent to learn to control a wheelchair [114]. These methods are very friendly to human trainers and do not require any particular training on their part.

Facial Feedback. Li *et al.* trained a mapping model to map implicit emotions to various types of explicit feedback data. Facial expressions were marked with different

types of feedback in advance, such as 1 for "happy" and 0 or -1 for "sadness" [168]. Based on this work, Gadanho introduced a facial feedback reinforcement learning method based on an emotion recognition system. The system can learn to decide when to change or reinforce its behaviour with Q-learning by identifying human emotions [169]. Arakawa *et al.* introduced the DQN-TAMER model, where an AI agent may obtain facial expressions via a camera, and then use the facial expression data to map different emotions as implicit rewards to improve learning efficiency [170]. Veeriah *et al.* proposed a method where the agent may analyse human facial features from camera images to gain additional rewards. As a result, the AI agent can quickly adapt to the user's facial changes in order to complete the task [171]. One of the limitations of this method is that human emotions cannot be identified merely based on facial expressions, and there may be a delay in converting machine recognition expressions into feedback.

Natural Language Feedback. When compared to facial expression and gesture-tracking feedback methods, natural language feedback makes it easier to convert the token vector of the sentence into quantitative feedback. Natural language feedback can be transformed and applied to several aspects of reinforcement learning, such as rewards, values, and policies. Goyal *et al.* introduced the LEARN (LanguagE-Action Reward Network) method, which is a reward shaping method [172]. In the state-action space of the task, if most of the reward signals are 0s, we call it the sparsity of rewards. Sparse rewards may cause the algorithm to converge slowly. AI agents need to interact with the environment several times and learn from a large number of samples to reach an optimal solution. One solution to this problem is to provide the AI agent with a bonus reward in addition to the reward function whenever the AI agent takes the right step toward the goal. This process is called reward shaping. Maclin and Shavlik proposed RATLE (Reinforcement and Advice, Consulting Learning Environment) [173], where the AI agent can translate human natural language suggestions into feedback for the Q-value function to accelerate the learning process. Kuhlmann proposed a method that transforms natural language suggestions into an algorithm-understandable formal language to optimise the

learning policy [174]. In addition to the methods described above for transforming into different parts of the algorithm, natural language can be used to directly guide the AI agent's learning policy. For example, Williams *et al.* proposed an object-oriented Markov Decision Process (MDP) framework that can map natural language to reward feedback [175].

3.7.3 Multi-modal Feedback

The research above is focused on a single input interaction method. On the other hand, multi-modal interactions are more prevalent and efficient in day-to-day human-human interactions. Multi-mode communication has the following benefits. First, when a single-mode piece of information is disrupted by noise or occlusion, other modes can be used as information supplements. Second, when multi-modal interaction is available, it has the potential to improve the robustness and reliability of communication. Quek *et al.* introduced a framework for analysing language's mutual support and accompanying gestures [179]. Cruz *et al.* proposed a dynamic multi-modal audiovisual interaction framework that would allow humans to provide feedback using their voices and gestures [180]. Griffith *et al.* [190] introduced a multi-modal interaction method based on hand gestures and a speech recognition system, which was restricted to operating geometric objects on maps. Weber *et al.* [180] developed a dynamic audiovisual integration method that allows humans to input information via natural language and gestures. In the above experiments, multi-mode interactions generally outperformed single-mode interactions. Most of the current multi-mode interactions are merely a combination of two modes, such as any two of voice, gesture, sound, and vision. One of the problems with the above multi-modal methods is their inability to combine various forms of human feedback. The ability of humans to interact directly with AI agents using multiple methods at the same time remains unexplored. In the future, these multi-mode interactive methods can be combined in more forms to develop effective human-AI collaboration for a broader range of scenarios.

Some studies take into account the effect of human fatigue caused by increasing training time on the quantity and quality of feedback. As training duration increases,

human trainers become exhausted, reducing the amount of feedback while simultaneously lowering the quality of the feedback [181, 183]. Methods for encouraging human trainers to raise interaction excitement through gamification were proposed; such methods have been found to decrease weariness and effectively improve human trainers' efficiency [182].

3.8 Algorithmic Models

In the previous section, we analysed how humans provide feedback to AI agents. This section categorises algorithmic models based on how agents receive and process human feedback.

3.8.1 Reward-based Methods

Reward-based methods accelerate the learning process by adjusting the reward that the AI agent receives from the environment. Concretely, after the AI agent receives feedback from the environment, humans can scale up or down the rewards based on their knowledge, potentially accelerating the learning process [189]. Computationally, the reward from human $H(s, a, s')$, is added to the reward from the environmental reward $R(s, a, s')$ to get the new reward $\bar{R}(s, a, s')$.

$$\bar{R}(s, a, s') = R(s, a, s') + H(s, a, s'), \tag{3.2}$$

Thomaz and Breazeal proposed a method for non-expert human trainers to influence the AI agent's next action by providing a positive or negative numerical reward. If the agent received negative feedback, it would attempt to reverse the previous action in order to get a higher score [188].

Knox and Stone introduced the TAMER algorithm, which uses human demonstration as input to guide the AI to perform better [160]. Based on the TAMER method, Riku *et al.* introduced a framework that combines the deep learning method and TAMER, named DQN-TAMER, where rewards are shaped by the human's nu-

merical binary feedback and environment [189]. Additionally, Arakawa *et al.* investigated a facial expression function based on the reward-shaped method, which is applied in a maze-like environment game [203]. The human trainers' facial expressions could provide feedback to the AI agents. The major shortcoming is that the recognition of human facial expressions is imprecise and intermittent.

Rosenfeld *et al.* developed a heuristic function method where the AI agent receives feedback generated by hand-engineered data from the human trainer [204]. The experiment's findings [205] indicate that heuristic functions may be a natural method for AI agents to learn from human trainers. The primary disadvantage of this approach is that it requires human trainers with extensive professional backgrounds and programming skills. It will be extremely hostile to non-professional users.

Reward-based methods can efficiently expedite the learning process in an environment with sparse rewards, but certain drawbacks are listed below. The first problem is "credit allocation", which is especially problematic in a rapidly changing environment where humans may be too slow to provide timely feedback. Therefore, the method's limitation remains how to map human rewards to corresponding actions. The second problem is "reward hacking", where the AI agent may achieve the greatest rewards by using ways humans would not expect [9].

3.8.2 Inverse Reward Design Methods

The agent is constantly attempting to optimise the human-designed reward function. When designing the AI agent, human developers always set the reward function based on the experimental environment, but the AI agent always encounters a new environment. Using the original reward function designed by humans in a new environment may lead to poor convergence. Mindermann *et al.* presented an inverse reward function in response to this issue [206]. To obtain the true target, this method is based on the designed reward function and the trained MDP. This allows agents to adapt effectively to the new environment, eliminating the issue of reward hacking. More specifically, this method takes the designed reward function, the test environment model, and the MDP in the new environment as input. Then a

Bayesian function maps the proxy rewards to the real rewards. The experiment in [207] demonstrates that the inverse reward method could successfully boost the AI agent's learning efficiency.

3.8.3 Policy-based Methods

Policy-based methods modify the learning policy of the AI agent action process to encourage the action to fit what the human trainers expect [9]. Human trainers may be aware of a large number of potential optimal actions A in a given state S; the probability of humans providing feedback to the AI agent can be denoted as C, where $0 < C < 1$. The difference between positive and negative human feedback can be expressed as $\Delta_{s,a}$. The probability that humans give policy feedback $\Pr_c(a)$ in a given state S can be expressed as

$$\Pr_c(a) = \frac{C^{\Delta_{s,a}}}{C^{\Delta_{s,a}} + (1-C)^{\Delta_{s,a}}} \tag{3.3}$$

Currently, the method that uses human critique for state and action pairs as input to shape agent policy is widely accepted. Griffith *et al.* proposed an optimal policy method based on human feedback, a Bayesian method that takes as input critiques for each state and action pair [190]. The experiments in [161] suggest that this policy-based method outperforms other reward-based methods.

Krening and Feigh conducted an experiment in which they compared two different policy-based methods that could bring a better user experience [208]. The first one is the critique feedback method proposed by Griffith [190], and the second is their Newtonian action advice method [208]. The result is that the method of action advice is better and the time required is reduced.

MacGlashan *et al.* proposed a Convergent Actor-Critic method, COACH (Corrective Advice Communicated by Humans). This framework allows non-experts to use numerical binary feedback to formulate policies through corrective suggestions [162]. Dilip *et al.* proposed a deep COACH method based on the original COACH, which uses raw pixels as input to train the AI agent's policy. The authors

argued that the use of highly representative inputs facilitates the application of the algorithm in more complex environments [126].

When compared to reward-based methods, the advantage of policy-based methods is that they do not require specific feedback from humans to AI agents. Nevertheless, humans must determine which strategy is most effective in assisting the AI agent. This may have higher requirements for the prior knowledge of human trainers.

3.8.4 Value Function based Methods

Value function-based methods estimate future rewards to obtain the highest potential reward at the end of the task, by using human knowledge [9]. They combine the value representing human preference with the value obtained by the AI agent from the environment to promote the learning process. Matthew *et al.* proposed a method that combines human preference, and agent value called Human-Agent Transfer (HAT) [191]. The algorithm generates a strategy based on recorded human trainer preferences, which it then applies to shape the Q-value function. This shaping process provides a stable reward for the state-action pair, in the Q-learning process. Brys *et al.* proposed a method that uses human demonstrations as input for a value named RLfD. This method generates a Gaussian function by human demonstration to guide the exploration process of the Q(λ) algorithm [192].

Despite the fact that value function based methods are likely to be an effective way of minimising human feedback, there are now just a few studies based on it.

3.8.5 Exploration Process-based Methods

Reinforcement learning is a method in which an AI agent needs to interact continuously with the environment and complete tasks based on rewards. This means that the AI agent must perform actions it has never tried before. This process is referred to as the exploration process. In exploration process-based methods, humans can increase efficiency by reducing AI agent errors, and unnecessary attempts [189]. Exploration process-based methods aim to minimise the action space by injecting prior

human knowledge to guide the AI agent's exploration in order to increase learning efficiency.

Thomaz and Breazeal conducted an experiment in the game Sophie's Kitchen to evaluate human guidance that helps the AI agent minimise its action space in order to enhance learning efficiency [164]. The results suggest that employing human prior knowledge to limit low utility efforts is more efficient than using scalar reward functions [193]. Suay *et al.* developed an upgrading approach in which the user may help exploration by highlighting goal states in the environment [209]. Yu *et al.* proposed an action-biasing approach that leverages user feedback to stimulate the AI agent's exploration process. The sum of the agent and user value functions is employed as a value function, to incorporate human feedback into the AI agent's learning process [210]. These methods are considered effective, but they generally need to be trained by humans, and this training process requires a lot of professional knowledge and participation.

In general, collaborative reinforcement learning has shown great potential for improving the efficiency of decision-making tasks. However, further research is needed to determine how to build environment models in which humans interact with AI agents. These models should consider not only the effectiveness and efficiency of interactive methods but also their interpretability, accountability, and possible ethical issues in decision-making. Therefore, in the following sections, we refer to the literature on the pattern of human-machine relations in the engineering field and propose guidelines for the future development of collaborative reinforcement learning methods.

3.9 Design Trajectory Map

Based on the previous CRL taxonomy, we propose a novel *CRL Trajectory Design Map* to guide researchers in designing CRL systems. When researchers start designing a human-AI collaborative reinforcement learning system, they could follow our *CRL Trajectory Design Map* (Fig. 3.5) step by step. First, they start with selecting a collaborative pattern from a macro perspective in the *Design Patterns* (Section

3.4) category. Next, they choose different collaborative levels and a number of the participants in the *Collaborative Levels and Parties* (Section 3.5). After that, they choose the collaborative capabilities that every party should have in the *Collaboration Capabilities* (Section 3.6). Finally, they select suitable interactive methods and algorithmic models for the specific task requirement categories of *Interactive Methods* (Section 3.7) and *Algorithmic Models* (Section 3.8).

Fig. 3.2 presents our newly proposed CRL taxonomy, which contains the most commonly used and highly cited methods and design patterns in the CRL research area, and which can also be used as a Trajectory Map (see Fig. 3.5) of designing collaborative reinforcement learning systems, as follows. Researchers may utilise our Trajectory Map to develop their architecture as they go from the top Design Patterns to the next, until the most detailed Algorithmic Model is selected. In the Map, the first part suggests **Design Patterns**, which are the most popular structure of human-AI collaborative frameworks in the CRL domain. These include cognitive systems engineering (CSE) [25], Bosch's framework [111], the Coactive design pattern [112] and Schmidt's framework [113]. The second part is the **Collaborative Levels and Parties**, while the Third part covers **Collaborative Capabilities**, which include understanding, communication, commitments, and institutions. The fourth part is **Interactive Methods**, including explicit and implicit interaction methods as well as multi-module interaction modes [9]. The last part reflects **Algorithmic Models**, which contains reward-based methods [189], value-based methods [191], policy-based methods [190], and exploration-process-based methods [164]. This taxonomy could be used as a systematic modelling tool for researchers and practitioners to select and improve their new CRL designs. They could choose an archetype in *Design Patterns* for the overall architecture at the start. Then, they could select a *Collaborative Level and the numbers of the Parties* in the collaboration. After that, they could select the *Collaborative Capabilities* that the AI agents should have, and select suitable *Interactive Methods* and *Algorithmic Models* that can meet the requirements of specific tasks. If researchers wish to learn about the most advanced technology developed in the last decade, they could check the classification we provide in Table 3.1.

In the following sections, we first revisit the theories and techniques of Reinforcement Learning in Section 3.2. We then review the classic Human-AI Collaborative Design in Section 3.4. Section 3.5 summarises patterns of human-computer interaction that have had significant impacts. Sections 3.6 to 3.8 summarise collaborative reinforcement learning algorithms using Schmidt's human-computer collaboration view as a taxonomy.

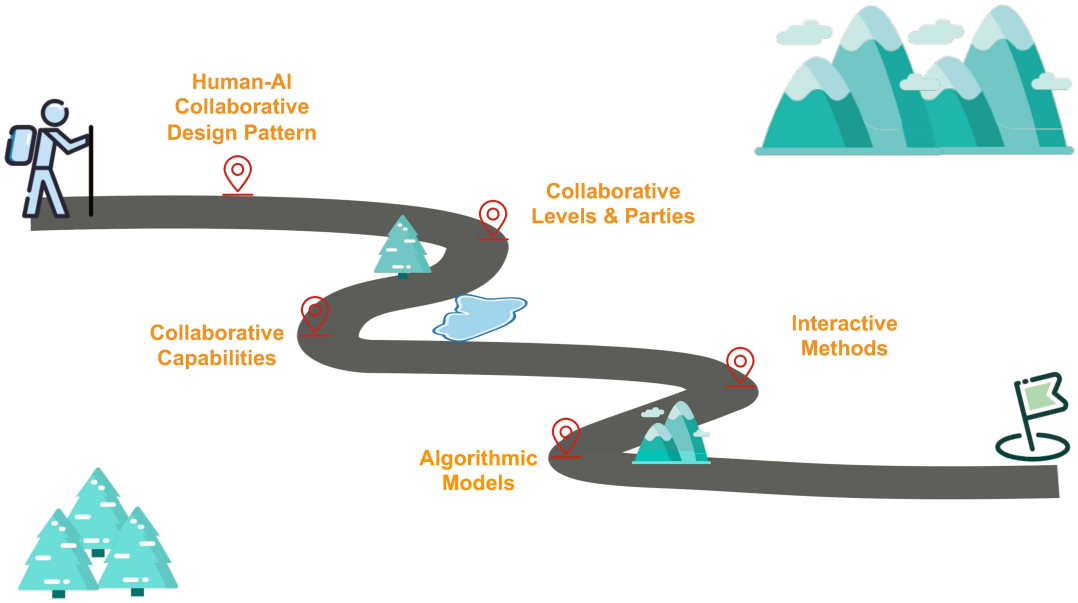


Figure 3.5: A Design Trajectory Map of Collaborative Reinforcement Learning Systems.

3.10 Summary

In this Chapter, we have presented a survey of Collaborative Reinforcement Learning (Collaborative RL, or CRL) to empower research into human-AI interactions and collaborative designs. This analysis resulted in us proposing a *new CRL classification method* (see Table 3.1), called *CRL Design Trajectory Map* (see Fig. 3.5) and a *new CRL taxonomy* (see Fig. 3.2) as a systematic modelling tool for selecting and improving new CRL designs. Researchers could use our Trajectory Map to design a CRL system from scratch or use parts of it according to their needs to refine their system. For example, they could select their desired system structure in the

Human-AI Collaborative Design Pattern, identify and satisfy the requirements of different components in Collaborative Levels & Parties and Collaborative Capabilities, and select different design components in Algorithmic Models and Interactive Methods. This comprehensive design approach is from top to bottom and macro to micro. In summary, through this survey, we provide researchers and practitioners with the tools to start improving and creating new designs for CRL methods.

Epilogue

In conclusion, the study presented in this chapter has addressed *RO 1.1* and *RO 1.2* to develop a comprehensive Human-AI CRL Design Trajectory Map that serves as a systematic modelling tool for selecting existing collaboration theories and CRL frameworks. We conceptualise existing frameworks by considering design patterns, collaborative levels, parties and capabilities, as well as reviewing interactive methods and algorithmic models to address the *RO 1.3*. Moreover, we have thoroughly examined the generic challenges associated with Human-AI Collaborative System to address the *RO 1.4*. By identifying these challenges, we aim to provide the research community with valuable insights and guidance for exploring new and innovative research directions in this field. This chapter addresses these research objectives to answer the **Research Question 1**: "What trajectory could we follow to develop a user-friendly Student-ITS collaboration system?"

CHAPTER 4

SimStu: A Transformer-based Approach to Generate Student Behavioural Data for Training ITS

Prologue

In Chapter 3, we have introduced a Human-AI collaborative system design trajectories map, which could be applied in guiding the design process of the Human-AI process. We utilise a Human-AI collaborative structure in the Student-ITS collaborative process. From the Student-to-AI process, we focus on developing student simulation methods to generate student behaviour data. This approach has the potential to improve the student-to-ITS process by minimizing the reliance on student interaction while maximizing the performance of the ITS.

ITS has made significant advancements using these technologies. Developing different teaching strategies automatically, according to mined student characteristics and learning styles, could significantly enhance students' learning efficiency and performance [7]. This requires the ITS to recommend different learning strategies and trajectories for different individual students. However, one of the greatest challenges is the scarcity of data sets providing interactions between students and ITS for the ITS training [211]. One promising solution to this challenge is to train "sim stu-

dents", which imitate real students' behaviour while using the ITS. The simulated interactions between these *sim students* and the ITS can then be generated and used to train the ITS to provide personalised learning strategies and trajectories to *real students*.

In this Chapter, we propose SimStu, built upon a Decision Transformer, to generate learning behavioural data to improve the training efficiency of ITS. The experimental results showed our Simtu could model real students well in terms of action frequency distribution and elapsed time distribution. Moreover, we evaluate SimStu in an emerging ITS technology, Knowledge Tracing. The results indicated that SimStu could improve the training efficiency of ITS.

Declaration: This chapter is based on the following publications:

***Li, Z.**, Shi, L., Cristea, A., Zhou, Y., Xiao, C., & Pan, Z. (2022, July). **SimStu-Transformer: A Transformer-Based Approach to Simulating Student Behaviour.** In Artificial Intelligence in Education. AIED 2022, Durham, UK, July 27–31, 2022, Proceedings, Part II (pp. 348-351). Cham: Springer International Publishing.*

***Li, Z.**, Shi, L., Zhou, Y., & Wang, J. **Towards Student Behaviour Simulation: A Decision Transformer based Approach.** In Augmented Intelligence and Intelligent Tutoring Systems: 19th International Conference, ITS 2023, Corfu, Greece, June 2–5, 2023, Proceedings (pp. 553-562). Cham: Springer Nature Switzerland.*

This chapter is presented largely as accepted, although referencing and notation have been altered, and cross-referencing has been added for consistency across this thesis. Some stylistic changes have been made for consistency. The majority of the text is verbatim, with some minor wording and formatting changes.

4.1 Introduction

In the previous chapter, we introduced the trajectory map for designing user-friendly and efficient human-AI collaborative systems. As we navigate the landscape of Human-AI Collaboration for Intelligent Tutoring Systems (ITS), this chapter embarks on a journey guided by the trajectory map's principles. Our destination: the development of innovative solutions to address a critical challenge in the realm of ITS - the scarcity of interaction data, often referred to as the 'cold start' problem.

The recent COVID-19 has significantly impacted people's educational activities, which promoted the Intelligent Tutoring System (ITS) to achieve significant development. Data-intensive approaches have been proposed for ITS to improve the quality of education service [212]. However, these need to be powered by data-hungry machine learning models, whose performance relies heavily on the size of training data available [213]. However, similar to the scarcity of labelled data in many AI fields, the shortage of student behavioural data has become one of the greatest challenges for ITS advancements [214]. Our work thus aims to tackle this challenge by answering the following research question:

How to create adequate high-fidelity and diverse simulated student behavioural data for training ITS?

The intuition of SimStu (shown in Figure 4.1) is that after ITS collects *a small amount* of real student behavioural data in the early stage development. It feeds the data into a generator, which produces *a large amount* of simulated student behavioural data. These simulated data can then be used, together with the real student behavioural data, to train the ITS. Therefore, the ITS could be enhanced to improve most users' experience. The generator, which we call "SimStu", is built upon the Decision Transformer [13].

In the subsequent research, to train and evaluate our SimStu model, we used the EdNet dataset[1], which is the largest student-ITS interaction benchmark dataset so far. Moreover, we improve the model's performance by modifying the input and hyperparameters. In this work, we propose an upgrade version of the SimStu, which

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

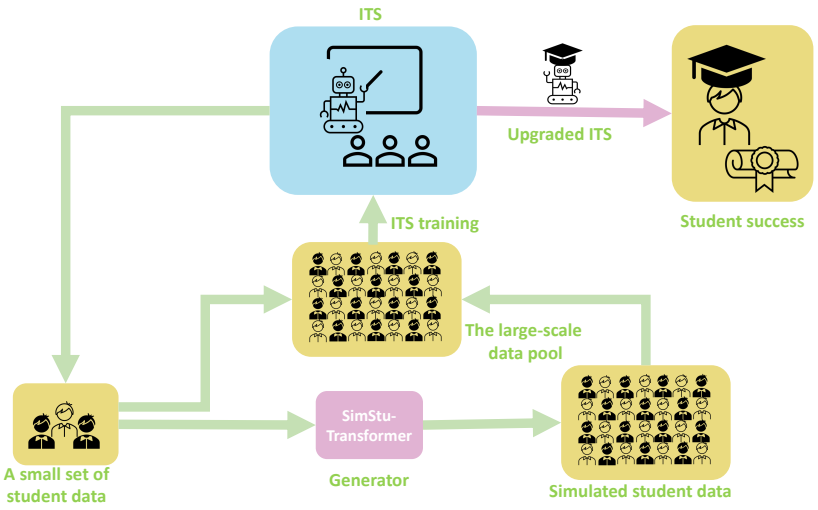


Figure 4.1: The intuition for the proposed SimStu pipeline in ITS.

get better performance. The results suggest that our method could simulate the real student well on the metrics of action distribution and elapsed time distribution. In addition, we apply our method in real educational scenarios, the Knowledge Tracing models. Knowledge Tracing (KT) is a kind of method that predicts the student's next action based on their historical behaviour data. There are many ITS utilising the KT models' prediction results to improve the student learning experience and be effective, i.e. giving recommendations and learning trajectories. Therefore, we apply our method's generated data in the state-of-the-art KT models, i.e., SAINT, SSAKT and LTMTL, to evaluate the performance of our model. The experimental results show that our method could improve the KT model's performance.

The main contributions of this Chapter lie in the following two aspects:

1. We propose a student learning behaviour simulation approach based on the Decision Transformer to provide adequate training data for ITS.

2. Our experiments demonstrate that a trained SimStu model can simulate real student behaviour well and outperform imitation learning-based models.

3. We apply the generated data by SimStu to real ITS technology, three Knowledge Tracing Model. The results show that our method could improve the efficiency of ITS training.

4.2 Related Work

4.2.1 Student Modelling

With increased attention to personalised learning, the traditional one-size-fits-all method can no longer satisfy user needs [215]. In offline scenarios, personalised learning can be supported by teachers in various ways. For example, a teacher can gain valuable information about their students, by observing their learning process and interactions, and then design the most suitable and beneficial learning strategy for them [216]. However, the lack of teacher-student interactions in online learning environments makes the personalisation process extremely difficult [217]. In such online scenarios, student modelling can and has been applied, as a powerful tool to combat this issue [218]. By doing so, individual student models can be built for students, thus enabling the system to adapt to their needs. According to [219], student modelling has been used to model a variety of characteristics, such as student abilities and knowledge [220], learning style [221], and reasoning [222]. Various techniques have been applied for student modelling, including data mining [223], machine learning [224], and overlay modelling [225].

Besides individual-level modelling discussed above, the other line of this research is group-level modelling [226], which is useful to model because 1) not all learning takes place in 'solo situations', and 2) especially in online learning settings, learning often occurs in a collaborative way, as students interact with not only the system, but also the teacher, and their peers. This additional information, on top of the individual student's own characteristics and behavioural data, can undoubtedly contribute to enhancing the efficiency of the system's capabilities of personalisation and adaptation.

Thereby, in the current study, we take advantage of the benefits of group-level student modelling and train our system using the learning data from a large number of individual students, in order to learn the patterns of student learning in the system. This can then enable the system to recognise "optimal" learning behavioural patterns, which lead to better student experience, performance, and learning results, as well as "poor" learning behavioural patterns, which may result in failure, thus

recommending not only personalised but also optimal learning trajectories to the students, or providing a reminder of progressing to potential failure. To achieve this objective, it is crucial to have a decent quality and quantity of training data to feed to 'data-hungry' machine learning models.

4.2.2 Knowledge Tracing

Knowledge Tracing (KT) is a common method of personalising learning strategies for individual students. It predicts whether a student has the capability to master a new piece of knowledge, by tracing the student's current knowledge state, which depends on past learning behaviour. The two major KT approaches are Bayesian Knowledge Tracing (BKT) [227] and Deep Knowledge Tracing (DKT) [66].

BKT is a probabilistic method for student model generalisation [228]. It uses the Hidden Markov Model (HMM), to model their knowledge state as a set of binary parameters, each of which indicates whether a single Knowledge Concept (KC) has been understood or not [59]. DKT considers knowledge tracing as a sequence prediction problem. It uses Recurrent Neural Network (RNN) to model a student's knowledge state in one summarised hidden vector [66]. DKT is powerful for capturing a complicated depiction of human learning. However, the parameters of the DKT model are non-interpretable [229], which may result in students distrusting the system and teachers being unable to understand student behaviour. Additionally, when dealing with sparse data, DKT may encounter the problem of not generalising well [230]. The main limitation of BKT and DKT is that they both rely on a huge amount of students' historical learning data [69]. Different from BKT and DKT, our approach generates *simulated* student learning data, thus not relying on a huge amount of *historical* data, and more importantly, the simulated data can be visualised in statistical charts, showing student's learning behavioural patterns and thus being able to mitigate the KT model's limitation of non-interpretability.

4.2.3 Transformers

Transformers have risen to prominence in the field of deep learning in recent years, particularly in natural language processing and image generation tasks [40, 231]. A Transformer is an encoder-decoder Sequence2Sequence architecture to model sequential data, which consists of stacked self-attention layers, as shown below.

$$z_i = \sum_{j=1}^{n} \text{softmax}\left(\{\langle q_i, k_{j'}\rangle\}_{j'=1}^{n}\right)_j \cdot v_j \tag{4.1}$$

where, $\{z_i\}_{i=1}^{n}$ denotes the n output embeddings. the i-th token is mapped via a transformer linear layer to a key k_j query q_i, v_i, and values v_j. The i-th output of the transformer is given by weighting the values v_j.

Before self-attention was introduced, the best-in-class architecture was the seq2seq model [232], with an attention component from the decoder to align weights to input positions in the encoder, deciding how much information to retrieve from each position of inputs. The purpose of the self-attention layer is to deprecate the traditional RNNs wrapped in the architecture completely, and instead to adjust each sequence of inputs based on its contexts, yielding contextualised embedding within each of the data instances, as opposed to fixed embedding, in previous model architectures.

Based on the Transformer architecture, Chen *et al.* [13] proposed the Decision Transformer, which abstracts the reinforcement learning problem, as a sequence modelling objective. The key in this algorithm is to generate actions based on *desired returns in the future*, rather than rewards in the past, and they proposed feeding a sequence of returns-to-go (sum of future rewards) $\widehat{R}_t = \sum_{t'=t}^{T} r_{t'}$ into the model. The equation below represents the trajectory denoted by τ, which consists of states s, actions a, and rewards $\widehat{R}$ at timestamp $t \in 1, ..., T$.

$$\tau = \left(\widehat{R}_1, s_1, a_1, \widehat{R}_2, s_2, a_2, \ldots, \widehat{R}_T, s_T, a_T\right) \tag{4.2}$$

This model first learns a linear layer for each in returns-to-go, state, and action, to project them to the embedding dimension, followed by a layer normalisation. A time-step embedding is also learned and added to the tokens, which are then fed into a GPT [233] architecture, with the goal of generating future actions. Our proposed

Sim is built upon this model. We feed the sequence of interactive data between students and the ITS into the Decision Transformer, to generate simulated student behaviour data.

4.2.4 Generative Pretrained Transformer (GPT)

The Generative Pretrained Transformer (GPT) model represents a class of state-of-the-art language models developed by OpenAI [233]. These models are founded on the transformer architecture, which has proven to be highly effective for sequence-to-sequence tasks due to its self-attention mechanism, allowing the model to weigh parts of the input data based on their relevance [40]. One of the distinguishing features of GPT is its unsupervised training approach. The model undergoes an initial pre-training phase on vast text corpora where it learns to predict the next word in a sequence. Following this, it can be fine-tuned on specific tasks, from machine translation to question answering, allowing it to achieve remarkable results even with smaller labelled datasets [234]. Moreover, subsequent versions of GPT, such as GPT-3, have demonstrated the capability to generalize across tasks without task-specific training, pushing the boundaries of few-shot and zero-shot learning [234].

4.3 Method

4.3.1 Architecture

The proposed SimStu is built upon the Decision Transformer [13] originally proposed by Chen *et al.* It consists of an encoder and a decoder and models the joint distribution of the sequence of student returns-to-go, states, and actions. Figure 4.2 illustrates the architecture. It separates student interactive trajectory sequences into two parts: one is used as the input embedding of the encoder, and the other is used as the output embedding of the decoder [40]. Then, the encoder takes the first part of the trajectory sequence embeddings as input and passes an output trajectory to the decoder. The decoder accepts a shifted embedding trajectory as input

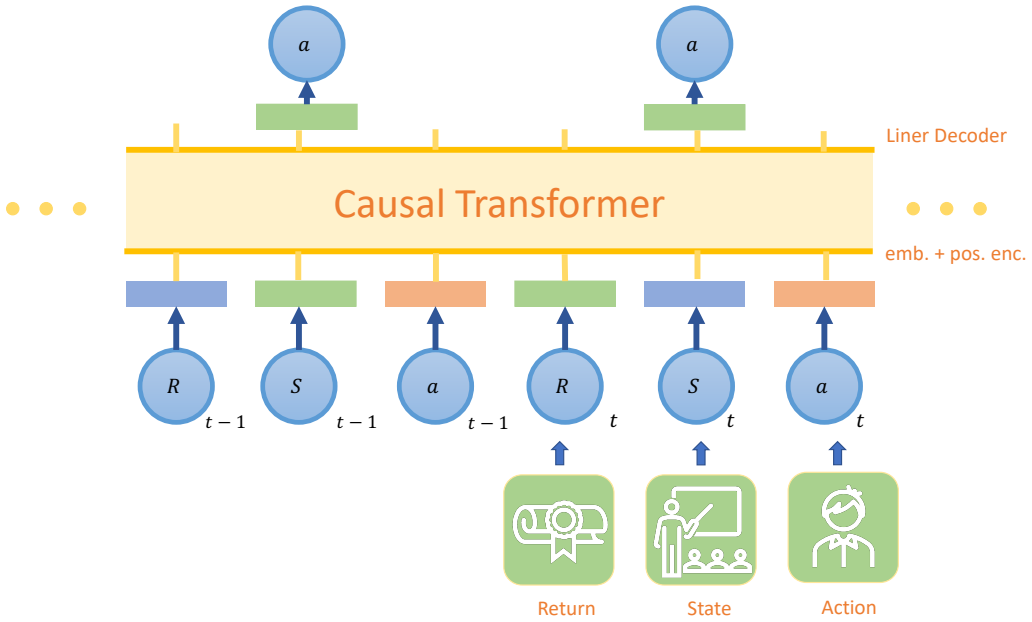


Figure 4.2: SimStu architecture.

Number of Interaction	131,441,538
Number of students	297,915
Number of exercises	13,169

Table 4.1: Statistics of the EdNet-KT4.

to produce the final output trajectory.

4.3.2 Dataset

The dataset used in our experiment is EdNet [235] - the largest student-ITS interaction benchmark dataset in the field of AIED/ITS. It contains more than 780K students' data extracted in South Korea over 2 years by a multi-platform ITS called SANTA[2]. EdNet consists of four hierarchical datasets, classified according to the number of interactions. We conduct our experiments based on EdNet-KT4, which includes problem-solving logs. Compared to KT-1 to KT-3, KT-4 provides the finest detailed interaction data, allowing access to specific features and tasks. Table 7.3 shows the statistical details of EdNet-KT4.

EdNet-KT4 contains 297,915 students' data. As this project aims to simulate

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

student data using a small amount of real student data, we randomly selected 1,000 students (a total of 861,247 action logs) for our experiments. We selected data from 200 students as the training data (a total of 139,835 action logs) to train the SimStu model, and another 200 students (with a total of 143,286 action logs) as the test data, to simulate student data. The remaining 600 are used as real data (a total of 578,626 action logs) to be compared with the simulated data.

Considering that learning behaviour may vary amongst students with different learning performances, we divided them into five groups, according to their scores. From Group 1 to Group 5, student performance ranged from "very good" to "very poor". Using this grouping strategy, we partitioned the training and test data using stratified sampling.

4.3.3 Trajectory Representation

The key desiderata of selecting the model features are to provide the algorithm with meaningful information to generate *the most likely trajectories*. We replace the timestamps with the difference between the individual timestamps, i.e., the time between switching actions. The single timestamp could contain little information, and the time values in the UNIX system that generated them are large. We thus reduce the large UNIX time integers to small values, which also are more suitable for training. Furthermore, we removed from the modelling data types with very sparse data, where it is difficult for the Decision Transformer model to learn anything from the small number of values actually presented in the data. For instance, as *cursor_time* is sparse, with a usual value of *NaN*, we removed *cursor_time* from the data. *action_type* is used to imitate students' behaviour, which is denoted by a in the Decision Transformer Trajectory τ. *user_answer*, denoted by R, is used for evaluating student performance, thus partitioning them into groups. We check whether the student's answers (options of a, b, c, and d) match with the correct answers: if yes, they get a positive reward of 1; and if no, they get a reward of 0. *item_id* is used for evaluating the feasibility of the learning paths, which takes as the state of the student and is denoted by s. Due to the fact that *user_id* does not affect or represent student behaviour, we choose to generate it randomly, after the

SimStu generation procedure ends.

4.3.4 Experiments

The SimStu was implemented using the Pytorch framework and trained on an Nvidia RTX 3090 GPU. We used the Adam optimiser with a batch size of 64. We set Adam betas as (0.9, 0.95). The initial learning rate was 0.0006, and the dropout rate was 0.1. To evaluate the proposed SimStu, we conducted two experiments.

In the first experiment, we compared the simulated data generated by the SimStu model with the real data. More specifically, we examined the average number of actions for the simulated and real data amongst the five student groups. Furthermore, we compared the similarity of the generated data and the real data using the Pearson product-moment correlation coefficient (PPMCC).

In the second experiment, we compared our SimStu with the Behaviour Cloning method proposed by Torabi [54], which is based on imitation learning. We used RELU as the nonlinearity function, with a standard batch size of 64. We set the initial learning rate as 0.0001 and the dropout rate as 0.1. More specifically, we investigated the distribution of 'elapsed time' between the Behaviour Cloning method and the SimStu with real student data by PPMCC. 'Elapsed time' is the amount of time a student works on a specific exercise. Students with different performances vary in the time spent practising, due to the differences in their knowledge and skills. Therefore, the 'elapsed time' can be used as a good metric to measure the performance of different groups of students [236].

In the third experiment, we evaluated SimStu using three top-performance KT models selected from the Riiid Answer Correctness Prediction Competition on Kaggle[3]. The top three models are SAINT, SSAKT, and LTMTI [4]. In the competition, Kaggle provides a dataset size is 2,500. We assume that 2,500 student record is sufficient for KT model training. Therefore, We selected five datasets that contained 500, 1,500, 2,000, and 2,500 student records, respectively. We fed these five datasets into the SimStu models, and then, generated another five simulated datasets (the

[3]https://www.kaggle.com/code/datakite/riiid-answer-correctness
[4]http://ednet-leaderboard.s3-website-ap-northeast-1.amazonaws.com

generated data size is equal to the original data size, for example, we input 500 student records data to SimStu, and generated another 500 student records). After that, we mixed the generated and original data to build a new dataset (in the case of the 500 data, we got a new dataset containing 1,000 student records). Lately, we fed these five mixed datasets into the three KT models respectively to compare whether our method could improve the training efficiency of KT models. The metric we used here is AUC (Area Under Curve).

4.4 Result and Discussions

Figure 4.3 shows the results from the first experiment: the average number of actions performed by the real students (on the left) and by the simulated students (on the right), across all those five groups. This suggests that the distributions between the real student data and the simulated student data share some similar statistical characteristics, i.e., in both real and simulated scenarios: 1) the "very good student" group (Group 1) is the largest group, whilst the "very poor student" group (Group 5) is the smallest group; 2) the "good student" group (Group 2) and the "average student" group (Group 3) have similar sizes; and 3) both the "good student" group and the "average student" group are much smaller than the largest "very good student" group (Group 1), and 4) both the "good student" group (Group 2) and the "average student" group (Group 3) are much larger than the smallest "very poor student" group (Group 5). However, the only difference is that in the Real Students scenario (on the left), the "poor student" group (Group 4) is the second smallest group and smaller than both the "good student" group (Group 2) and the "average student" group (Group 3), whilst in the Simulated Students scenario (on the right), the "poor student" group (Group 4) is the second largest group and larger than the "good student" group (Group 2) and the "average student" group (Group 3). Nevertheless, this result suggests that our SimStu model can generate student data which is similar to real student data.

As Figure 4.3 shows, the SimStu performed better in simulating the behaviour of students with higher grades (i.e. groups 1 ("very good") to 3 ("average")) than

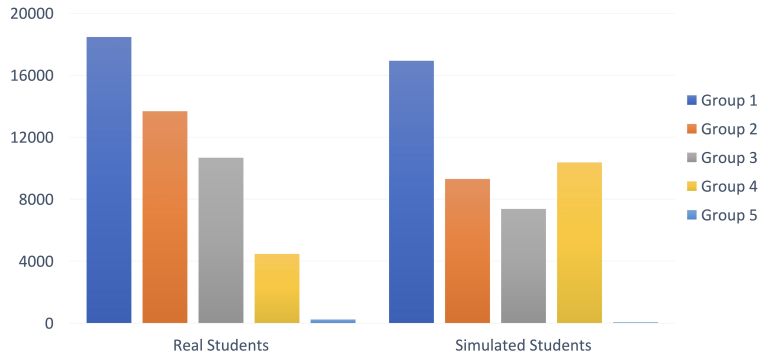


Figure 4.3: Action statistics of real student data (left), and simulated student data (right).

for lower grades students (i.e. groups 4 "poor" and 5 ("very poor")). This is in line with the difference in the amount and the frequency of actions. The abnormality of Group 4 may stem from the fact that students with lower learning performance tend to interact less frequently with the Intelligent Tutoring System (ITS). This sparsity in data results in weak causal relationships between actions, making it challenging for the model to understand and predict student behaviours accurately. Students who study better generally spend a longer time interacting with the ITS compared to students with relatively poor learning performance. This pattern makes many actions sparse and the causal relationship between actions weak, so the model cannot understand students' behaviours well. To paraphrase Tolstoy's words, "All good students may behave alike, but all poor performance students have their own reasons" [237].

Figure 4.4 and Table 4.2 show the action frequency distribution of the real student data (on the left) and the simulated student data (on the right). This result shows that the simulated data generated by our Sim-Transformer is similar to the real data in major action frequencies. For example, the main actions of the generated data, such as *respond*, *enter*, *play_audio*, and *submit*, have similar frequencies in each group. However, there are some differences in the actions that occur less frequently, such as *pay* and *undo_erase_choice*. The resulting PPMCC value of all actions is equal to 0.714, which suggests that the simulated student data and the real student data are 71.4% similar in the average distribution of actions. The result

suggests that simulated data is statistically similar to real data.

	Group 1		Group 2		Group 3		Group 4		Group 5	
Action	R	S	R	S	R	S	R	S	R	S
E	0.203	0.128	0.245	0.099	0.313	0.222	0.314	0.260	0.306	0.338
RE	0.068	0.095	0.190	0.074	0.185	0.230	0.231	0.166	0.224	0.261
S	0.065	0.066	0.114	0.030	0.155	0.108	0.181	0.129	0.224	0.132
Q	0.138	0.062	0.131	0.007	0.158	0.114	0.064	0.131	0.082	0.205
EC	0.000	0.000	0.024	0.000	0.007	0.059	0.004	0.001	0.000	0.000
UEC	0.000	0.000	0.001	0.000	0.000	0.006	0.000	0.000	0.000	0.000
PLA	0.254	0.298	0.102	0.041	0.069	0.119	0.099	0.150	0.061	0.030
PAA	0.256	0.302	0.102	0.041	0.071	0.119	0.040	0.153	0.061	0.034
PLV	0.008	0.011	0.045	0.002	0.024	0.014	0.021	0.005	0.020	0.051
PAV	0.008	0.011	0.044	0.002	0.024	0.010	0.044	0.005	0.020	0.047
P	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
REF	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
EC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 4.2: Comparison of the frequency of actions between simulated student data and real student data (R: real student, S: simulated student, E: enter, RE: respond, S: submit, Q: quit, EC: erase_choice, UEC: undo_erase_choice, PLA: play_audio, PAA: pause_audio, PLV: play_video, PAV: pause_audio, P: pay, REF: refund, EC: enroll_coupon).

In the second experiment, we fed the same training data and test data to the Behaviour Cloning model, which generated 600 student trajectories data (a total of 4,413,561 actions). Figure 4.5 shows the distributions of elapsed time of the real student, Figure 4.6 shows the distributions of elapsed time of the SimStu simulated student, and Figure 4.7 shows the distributions of elapsed time of the Behaviour Cloning model simulated student. The PPMCC value of the SimStu simulated data versus the real data is 0.762; while the PPMCC value of the Behaviour Cloning model simulated data versus the real data is 0.683. This shows that the SimStu simulated data is more similar to the real data, which suggests that our SimStu model outperforms the Behaviour Cloning model. This result may be due to the fact that when processing sequential student behavioural data, the student actions sequence context allows the SimStu to identify which policy can result in an action that promotes better learning states and improves training dynamics.

In the third experiment, We evaluate the SimStu using the three state-of-the-art KT models. Figure 5.12 shows the pairwise AUC comparisons of the three KT

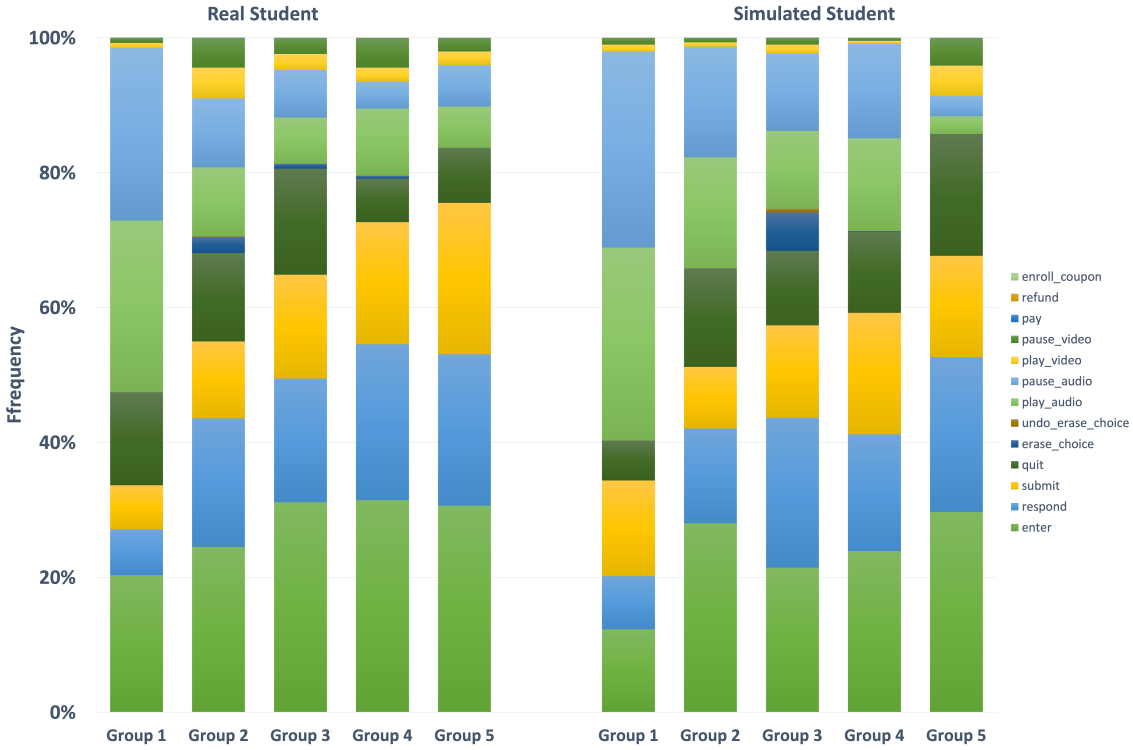


Figure 4.4: Action frequency distribution of real student data (left), and simulated student data (right).

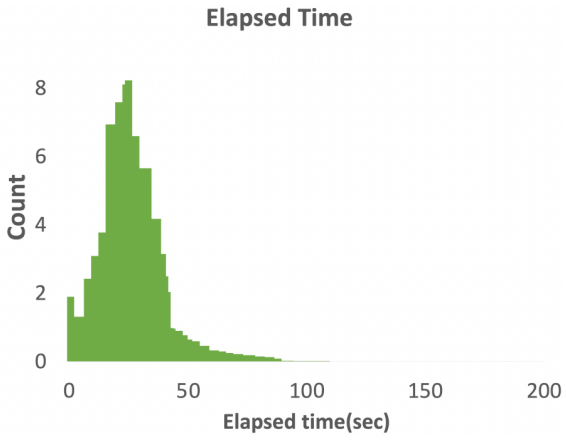


Figure 4.5: Elapsed time of Real Student Data

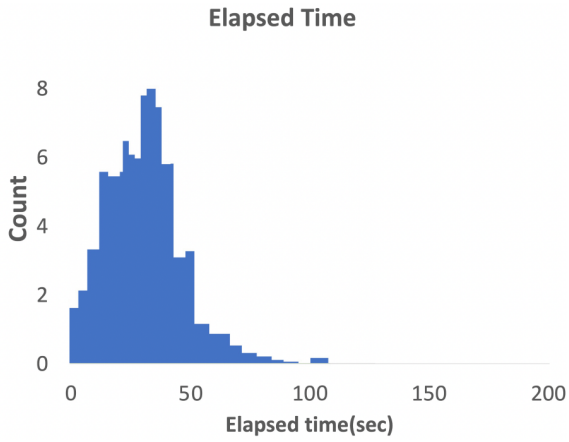


Figure 4.6: Elapsed time of SimStu

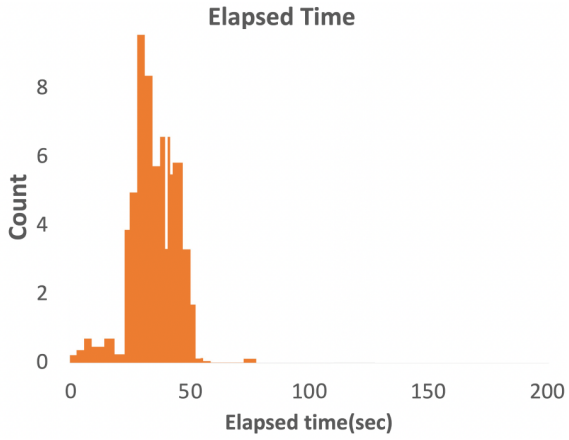


Figure 4.7: Elapsed time of Behaviour Cloning method Data

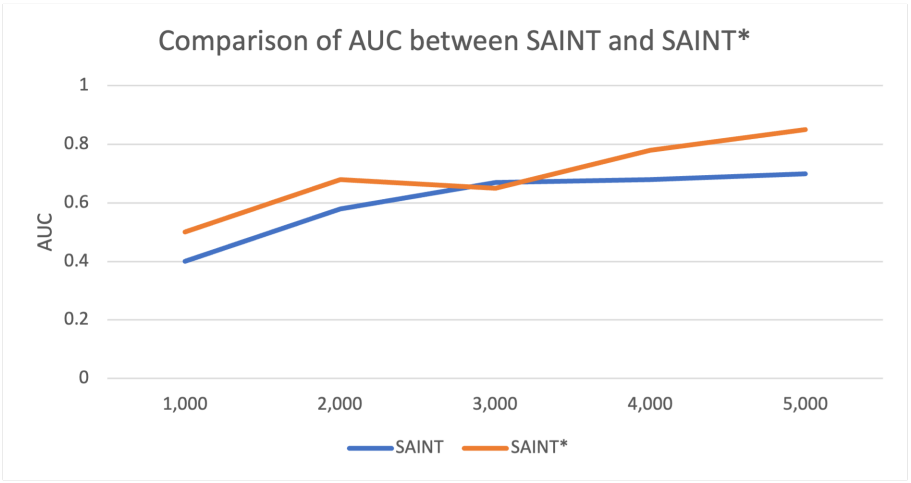
models trained on original datasets (SAINT, SSAKT and LTMTL, in blue) and the trained on the mixed dataset (SAINT*, SSAKT* and LTMT*L, in orange). In particular, The curves of SSAKT* and LTMTL* are constantly higher than the SSAKT and LTMTL. The curve of SAINT* is higher than the SAINT in every dataset, except in the case of the dataset size of 3,000. The results suggested that our method could improve the training efficiency of KT models, which have the potential to apply in real ITS.

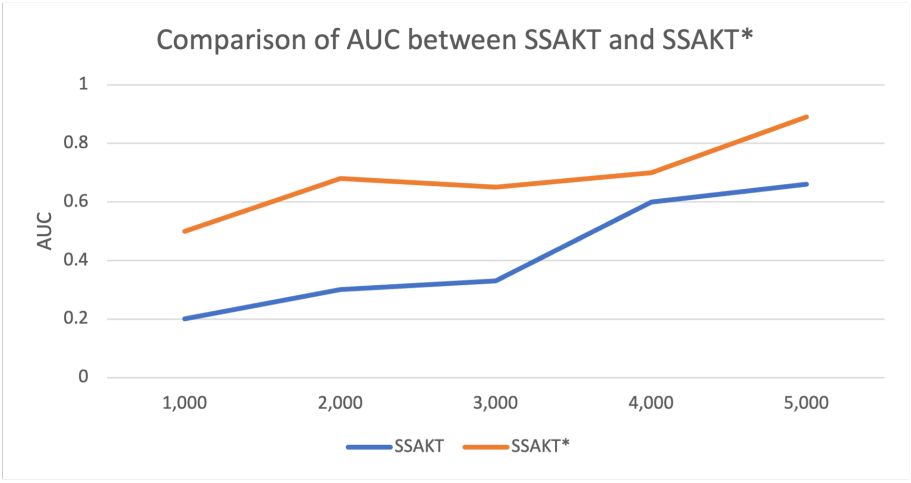
4.5 Summary

In this Chapter, we have proposed SimStu, a Transformer-based approach to simulating student behaviour, aiming to tackle the challenge of the scarcity of datasets for training ITS. We used the EdNet data to train the SimStu model, which generated learning behaviour data that could simulate the learning trajectories of different students. This method could be implemented in an ITS, such that it starts with collecting a small amount of student data, then uses our method to generate a large amount of simulated student data, and finally uses these data to train the ITS and improve its performance. The experimental results showed that SimStu could simulate the students' behaviour data well in terms of action distribution and elapsed time. Moreover, we evaluated SimStu by using three KT models. The results indicated that our method could improve the efficiency of ITS training.

Epilogue

This chapter has accomplished the following objectives: we designed the SimStu, which is a Transformer-based model to generate simulated student behavioural data to address *RO 2.1*. The aim was to develop a model capable of producing realistic and diverse student behaviours to train the ITS. Moreover, we evaluated the performance of the proposed model and compared it with existing methods to address *RO 2.2*. We explored the application of the generated data from the proposed model to an emerging ITS technology known as Knowledge Tracing to address *RO 2.2*. This





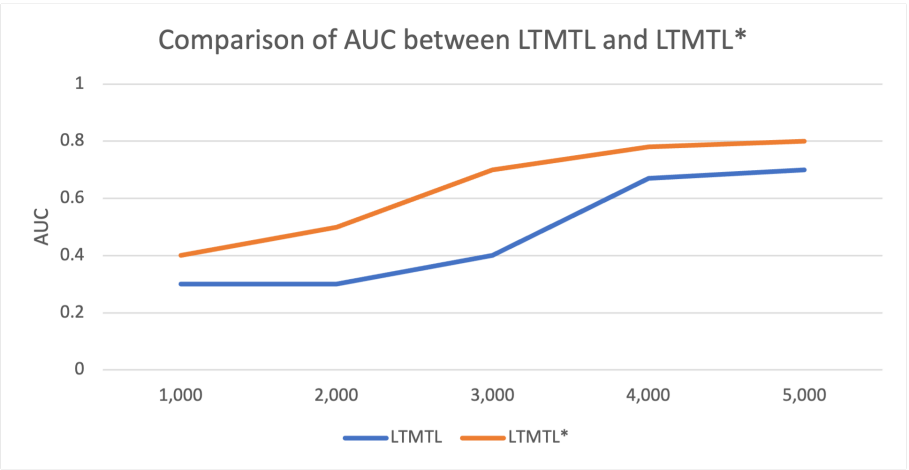


Figure 4.8: Pairwise AUC comparisons of the three KT models trained on only original students' data (SAINT, SSAKT, LTMTL, in blue) and trained on the mixed dataset (SAINT*, SSAKT*, LTMTL*, in orange).

application aimed to assess the performance of the proposed model in improving the accuracy of Knowledge Tracing. By addressing these objectives, this chapter addressed the **Research Question 2**: "How to generate high-fidelity and diverse simulated student behavioural data for training Intelligent Tutoring Systems (ITS) by using deep learning methods?"

CHAPTER 5

Sim-GAIL: A Generative Adversarial Imitation Learning Approach of Student Modelling for Intelligent Tutoring Systems

Prologue

In Chapter4, we have presented SimStu, a student behaviour data simulation method based on a deep learning method, the decision transformer, which could generate high-fidelity and diverse simulated student behavioural data to train the ITS.

In this chapter, we still focus on the Student-to-ITS process and further explore another student modelling method based on Generative Adversarial Imitation Learning (GAIL), which could be used to train the ITS by replacing human students with sim students (simulated students via student modelling). We analyse and compare the performance of our student modelling method with two traditional Reinforcement Learning-based and Imitation Learning-based methods using action distribution evaluation, cumulative reward evaluation, and offline-policy evaluation. The experimental results suggest that our method outperforms traditional student modelling methods on most metrics. Moreover, we apply our method to a domain

plagued by the cold start problem, Knowledge Tracing (KT). The experimental results show that our method could effectively improve the KT model's prediction accuracy in a cold-start scenario.

Declaration: This chapter is based on the following publications:

Li, Z., Shi, L., Zhou, Y., & Wang, J. (2023, August) ***Sim-GAIL: A Generative Adversarial Imitation Learning Approach of Student Modelling for Intelligent Tutoring Systems.*** Journal Neural Computing and Applications.

This chapter is presented largely as the manuscript submitted, although referencing and notation have been altered, and cross-referencing has been added for consistency across this thesis. Some stylistic changes have been made for consistency. The majority of the text is verbatim, with some minor wording and formatting changes.

5.1 Introduction

Research in cognitive science has shown that there is a strong relationship between the sequence of learning materials and learning outcomes [238]. 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 [239]. Research on customised learning trajectories for students has been 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 [10]. Although many mature ITSs have sufficient data to train algorithms, a large number of emerging ITSs are still suffering from the lack of training data in the early stages of development, also known as the cold start problem [240].

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. For example, Jarboui *et al.* attempted to model student trajectory sequences into a Markov Decision Process [241],

but in real educational scenarios, only a few ITS can provide all the feature data consistent with Markov Decision Process (e.g., the reward function of the ITS agent). Zimmer *et al.* defined reward functions to build reinforcement learning agents [242] 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.* proposed a student simulation method based on Behavioural Cloning (BC) [243], the simplest form of Imitation Learning which aims to solve the abovementioned problems where the reward is sparse and hard to define [244]. 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 discussions above, the research question of this chapter is: ***How to build an efficient student simulation method that can generate massive student learning data, which can be used for ITS training?***

In this chapter, we propose Sim-GAIL, a Generative Adversarial Imitation Learning (GAIL) approach to student modelling. This Sim-GAIL method can be used to generate simulated student data to solve the data lacking and cold start problems in ITS training. In the ITS field, there are two types of student modelling methods. The first is based on expert knowledge [245], e.g., building a reward function based on domain criteria [241, 242]. However, the reward function of the Reinforcement Learning based method appears challenging to define. The second is data-driven [47], e.g., Behavioural Cloning (BC) [246] based methods. However, BC-based methods suffer from low simulation accuracy [14]. We compare our Sim-GAIL with RL-based and BC-based student modelling approaches using data from EdNet [235]. 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 chapter 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 well. Our method outperforms traditional RL-based and BC-based methods on most metrics and could improve the training efficiency in cold start scenarios.

This chapter is structured as follows. Section 5.2 introduces the background of reinforcement learning, imitation learning (including behavioural cloning), and student modelling. Section 5.3 demonstrates the dataset, data pre-processing, and model architecture. Section 5.4 outlines the experiments and baseline models. Section 5.5 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 5.6 discusses our findings and future works. Section 5.7 draws conclusions.

5.2 Background

5.2.1 Markov Decision Process & Reinforcement Learning

Markov Decision Process (MDP) is the standard method for sequential decision-making (SDM) [47]. The models in sequential decision-making could generally be seen as an instance of Markov decision process. Reinforcement learning is also

typically regarded as an MDP [48]. Therefore, in this section, we introduce MDP and then reinforcement learning.

Markov Decision Process

Markov Decision Process (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 [50]. 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 [91]. Figure 5.1 shows the structure of the MDP.

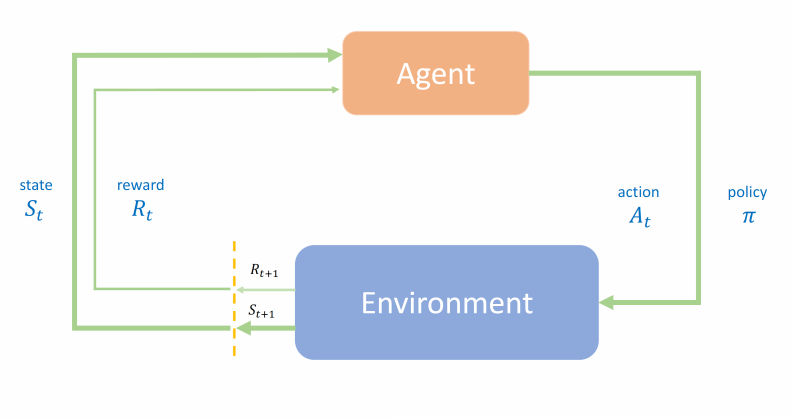


Figure 5.1: Framework of the Markov Decision Process.

Reinforcement Learning

Reinforcement Learning (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 action from the current state to the next state and $R : S \times \mathcal{A} \times S \to \mathbb{R}$ 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.

5.2.2 Imitation Learning

Different from RL, where the agent learns by interacting with the environment to obtain the maximum rewards, Imitation Learning (IL) is a method of learning policy that involves emulating the behaviour of experts' trajectories [51], instead of leveraging an explicit reward function as in RL.

Behavioural Cloning

Behavioural Cloning (BC) considers the learning of policy under supervised learning settings, leveraging state-action pairs [52,53]. Albeit simple and effective, BC suffers from the heavy reliance on extremely large amounts of data [14,247], without which a distributional mismatch, often referred to as covariate shift [55,248], would occur due to compounding errors and stochasticity in the environment during test time.

Apprenticeship Learning

Different from BC, Apprenticeship Learning (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 [56]. 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.

5.2.3 Generative Adversarial Imitation Learning

Generative Adversarial Imitation Learning (GAIL) addresses the drawbacks of RL and AL effectively [14] by borrowing the idea of Generative Adversarial Networks (GANs) [249]. It is derived from a type of Imitation Learning, called Maximum Causal Entropy Inverse Reinforcement Learning (MaxEntIRL) [57].

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.

5.2.4 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 [215, 250]. Various student modelling methods have been proposed, which are generally classified as integrating expert knowledge-based or data-driven methods [251, 252]. Knowledge-based methods refer to utilising human knowledge to address issues that would normally require human intelligence [17]. Data-driven methods simulate students' learning trajectories through massive student learning records data.

The majority of the studies in this field involve building different forms of student models to train a reinforcement learning (RL) agent [253]. Glesias *et al.* proposed a Markov Decision Process based on expert knowledge to train student models [245]. Doroud *et al.* suggested an RL-based agent method rooted in cognitive theory to optimise the sequencing of the knowledge components (KCs) [253]. 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 from human participants, which is excessively human resource-intensive. Yudelson *et al.* 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 [227]. Segal *et al.* suggested a student simulation method based on Item Response Theory (IRT) [254], which could respond to different reactions to courses at different difficulty levels [255].

Compared with integrating expert knowledge-based methods, data-driven methods could better simulate real students' learning trajectories and more effectively reduce biases [47]. There have been some studies [256–258] 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 [259]. 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. With the further development of ITS research, more and more high-quality datasets, such as EdNet [235], have been published in recent years, which makes it possible to achieve a high-quality data-driven student simulation method. 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.

5.3 Method

In this section, we introduce the methodology of our experiments. First, we describe the EdNet dataset we use in Section 5.3.1. In Section 5.3.2, we describe how we preprocess the data in EdNet to obtain the features we need. We then articulate the framework of our method in Section 5.3.3.

5.3.1 Dataset

We adopt EdNet [235], the largest dataset in the ITS field, for our experiments. This dataset comprises of 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

Number of Interactions	131,417,236
Number of Students	784,309
Number of Exercises	13,169

Table 5.1: Statistics of the EdNet

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 7.3. The interaction logs for each student are recorded in an independent CSV (Comma-Separated Values) file. EdNet is a four-layer hierarchical dataset divided from KT1 to KT4, according to the granularity of interactive actions. KT1 only contains simple information, such as question and answer pairs information 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.

5.3.2 Data Preprocess

The problems involving decision-making processes are transformed into MDPs in general [241]. In this experiment, we take 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. In this section, we explore the data and extract the *action space* and *state space*.

[1]https://www.aitutorsanta.com
[2]https://www.riiid.co

Action Space

There are 13,169 questions, 1,021 lectures, and 293 kinds of skills in EdNet [235]. However, there are no criteria for separating these courses into different parts. Bassen *et al.* [10] proposed a method to group knowledge concepts based on the assumption that each part was grouped by domain expert's 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 too granular for the action space, we also use the method proposed in [255], 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 have no difficulty ranking and thus being allocated a default difficulty of 0. Therefore, all action spaces are divided into 5 difficulty levels with 7 groups and 35 action types in total. Figure 5.3 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, action 5 corresponds to lectures where the difficulty level cannot be defined, which is set as 0. As shown in Figure 5.3, there are 7 groups in total that follow this pattern.

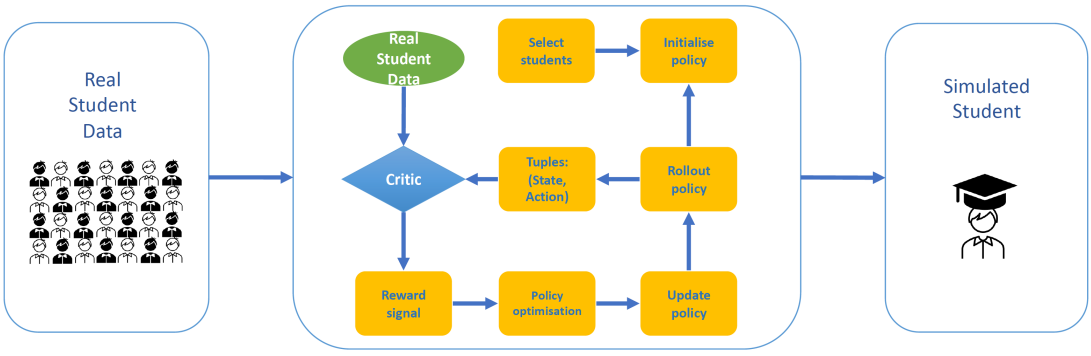


Figure 5.2: The Sim-GAIL Pipeline

State Space

EdNet records the interaction data of each student with the system by UNIX timestamps in separate CSV files. Therefore, most of the state features obtained from

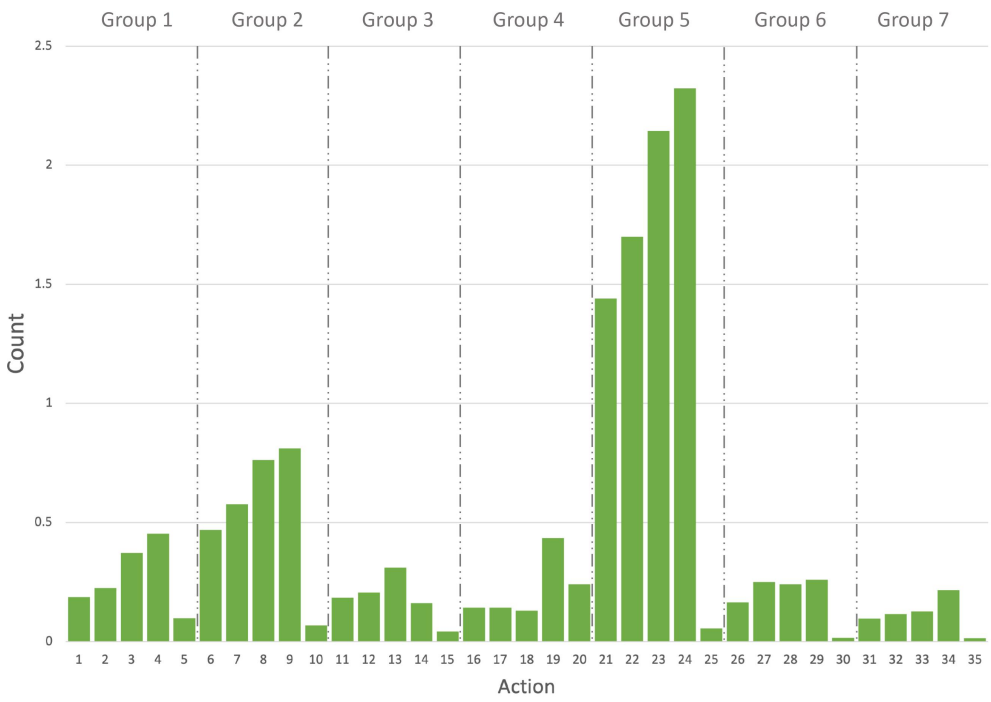


Figure 5.3: Action distribution of EdNet dataset.

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 [256, 260]. We select the state features which are widely chosen in similar simulated student works [10, 245, 256, 258]. These selected states could well represent a student's learning trajectories. Table 5.3.2 shows the *features* we select from EdNet: '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' indicates the student's accuracy rate before answering a specific question, while 'correct_so_far' represents the cumulative accuracy rate up to and including the current question. These metrics effectively capture the student's overall accuracy rate fluctuations as they progress through the questions. And 'steps_in_part' counts student learning steps in the current group. 'lecture_consumed' refers to the lectures that students have studied. Table 5.3.2 shows the *state features* constructed from EdNet. Compared to previous works [10, 256], we select more *state features*, which could potentially simulate the students' trajectories in real situations more effectively.

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

Table 5.2: State feature representation

5.3.3 Model Architecture

Our Sim-GAIL is built upon Generative Adversarial Imitation Learning (GAIL) [14], which aims to solve the problem where Imitation Learning has difficulty in dealing with constant regularisation and cannot match occupancy measures in large environments. Equation 5.1 demonstrates the optimal negative log loss distinguishing between the state π and action π_E pairs.

$$\psi^*_{\mathrm{GA}}\left(\rho_\pi - \rho_{\pi_E}\right) = \max_{D \in (0,1)^{\mathcal{S}\times\mathcal{A}}} \mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))], \tag{5.1}$$

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 could be derived:

$$\underset{\pi}{\mathrm{minimize}} \psi^*_{\mathrm{GA}}\left(\rho_\pi - \rho_{\pi_E}\right) - \lambda H(\pi) = D_{\mathrm{JS}}\left(\rho_\pi, \rho_{\pi_E}\right) - \lambda H(\pi). \tag{5.2}$$

This equation combines Imitation Learning (IL) and Generative Adversarial Networks (GAN) [249]. 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 equation 5.1 is the occupancy measure of the real trajectories.

$$\mathbb{E}_{\pi}[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi) \tag{5.3}$$

There is a function approximation of π and D. TRPO [261] is used to find a saddle point (π, D), which decreases the value of Expression 5.3. 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 5.2 shows the pipeline of Sim-GAIL. Real student data from EdNet is processed by the methods introduced in Section 5.3.2 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 Adam gradient [262]. Then, we take a policy update step using the TRPO rule to decrease the expected cost [261]. At last, we take a KL-constrained natural gradient step to train the Discriminator.

Algorithm 1 Algorithm of Sim-GAIL.

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

1: **for** each $i = 0, 1, 2, ...$ **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**

5.4 Experiments

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

5.4.1 Sim-GAIL

In order to simulate the real student learning behaviour policy in a real platform, we build a simulator to playback 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 '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. At last, we finished the simulation and obtained the trained policy.

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 1,000 students' interaction logs.

5.4.2 Baseline Models

Among the few studies that could be selected as baseline methods, the top performers so far are the Behavioural Cloning based method proposed by Torabi [54] and the Reinforcement Learning-based method proposed by Kumar [263]. Therefore, we use these two methods as the baselines for the experiments.

Behavioural Cloning (BC)

The first baseline is the Behavioural Cloning (BC) based method proposed by Torabi [54]. This model has shown good performance in the task of simulating users' behaviour from observations. Similarly, we employ a Mixture Regression (MR) approach [264], which is a Gaussian mixture of the actions and states, to process the data features. We use the same action-state pair data to train the Sim-GAIL and BC-methods, and the data is extracted from EdNet. The supervised learning method is applied to train the policy and Adam optimisation [262] with a batch size of 128.

Reinforcement Learning (RL)

The second baseline is the Reinforcement Learning (RL) based method proposed by Kumar [263], which uses the Conservative Q-learning (CQL) approach. EdNet does not contain any students' prior- or post-test scores. We use the method proposed by Azhar*et al.* [265] to build a reward function based on the historical logs of students' scores. More specifically, we use the correctness of the student's responses as the reward function. If the 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. Check whether the student's responses match the correct answers; if yes, they get a positive reward of 1 to 4, and if no, they get a negative reward of -1 to -4. The Dynamic Programming (DP) [266] 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 [267] to make the policy iteration based on the current policy.

5.5 Evaluation

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

In the first part evaluation, as shown in Figure 5.1, since the most critical elements for a Markov Decision Process are *action*, *reward*, and *policy*, we evaluate the modelling effect 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 actions and the real actions from the historical 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 generally is used to measure the difference between two distributions [268]. Second, we compare the Expected Cumulative Rewards (ECR) 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.

In the second part evaluation, we use three state-of-the-art Knowledge Tracing models to evaluate the 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 students' next actions through their historical behavioural trajectories [240]. 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.

5.5.1 Action Distribution Evaluation

As mentioned in Section 5.3.2, we obtain the action distribution of EdNet by allocating 35 actions into seven groups, resulting in five actions per group, as shown in Figure 5.3. 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. 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 5.5 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 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 5.6 shows the action distribution of the simulated students generated by the Behavioural Cloning (BC) based method. Within this distribution,

Table 5.3: Kullback–Leibler divergence of action distribution.

Model	Sim-GAIL	RL	BC
KL value	0.297	0.408	0.391

actions in group 6 illustrate the highest frequencies, indicating that actions in group 6 are the most frequent ones. Figure 5.7 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 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. KL is a measure from information theory that quantifies the difference between two probability distributions. It provides an asymmetric measure of the information lost when one distribution is used to approximate another [268]. The KL divergence, denoted $D_{KL}(P \parallel Q)$, between two probability distributions P and Q over the same discrete event space is defined as:

$$D_{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{5.4}$$

where $\mathcal{X}$ is the set of possible events, and $P(x)$ and $Q(x)$ are the probabilities of event x according to distributions P and Q, respectively. Note that the KL divergence is non-negative and is zero if and only if P and Q are the same distribution.

Table 5.3 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 Sim-GAIL generating data's action distribution and real data's action distribution 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

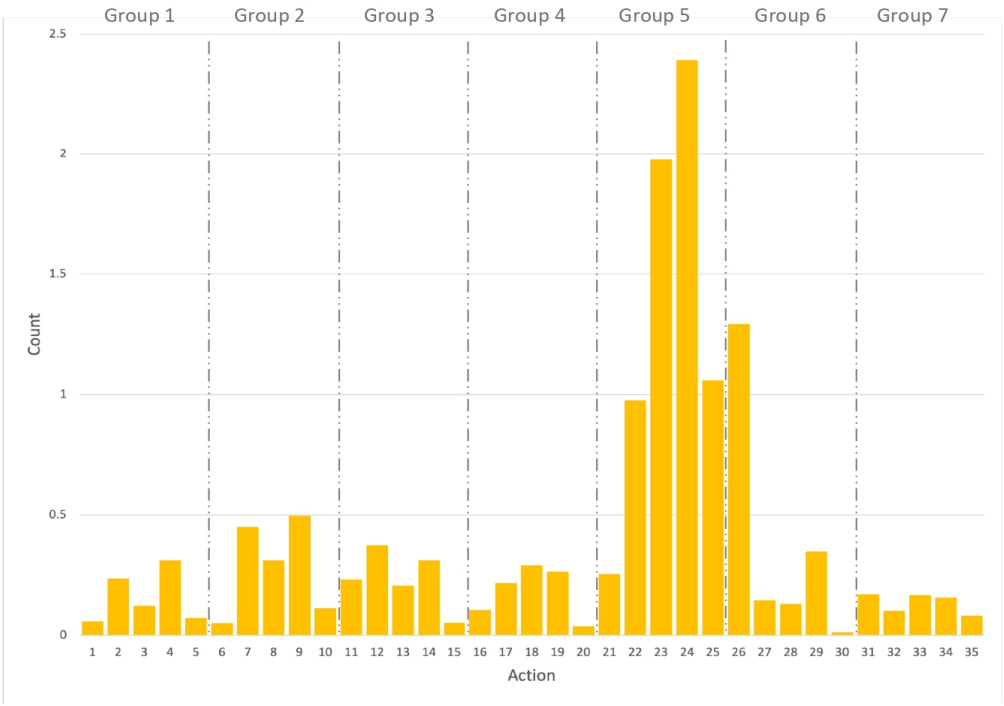


Figure 5.4: Action distribution of the Sim-GAIL model.

compare the action distribution of the state value 'topic_fam' from simulated students generated by three different methods, which is shown in Figure 5.8. 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

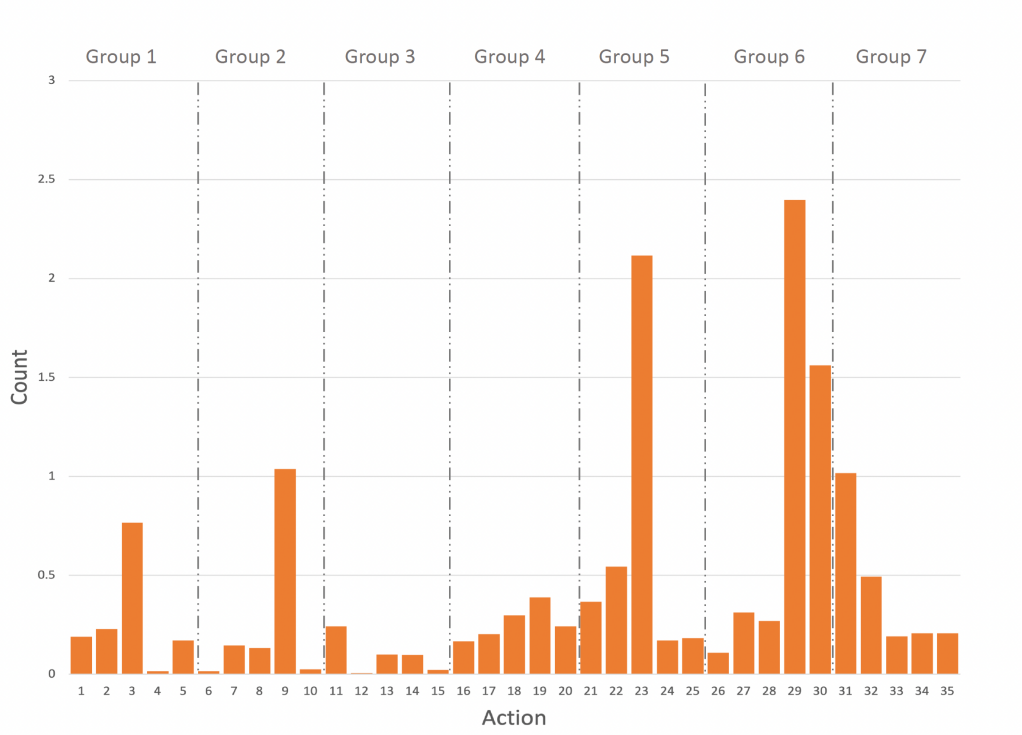


Figure 5.5: Action distribution of the Reinforcement Learning-based model.

by the Sim-GAIL method can balance the data distribution and optimal policy to achieve a better simulation effect.

The assignment of a 0 reward to lectures is based on the specific learning scenario we are modelling. In many educational contexts, attending lectures is often considered a passive activity, and students do not directly take action during lectures. Instead, they are expected to acquire knowledge passively by listening and observing. In the context of reinforcement learning, assigning a reward of 0 to lectures encourages the simulated students to focus on active learning actions, such as answering questions or engaging in interactive activities, rather than simply attending lectures. This is consistent with the idea that students' active participation and engagement tend to be more indicative of their learning progress.

5.5.2 Expected Cumulative Rewards Evaluation

Expected Cumulative Rewards (ECR) represent the average of the expected cumulative rewards under a given policy [269]. ECR could effectively reflect the cumulative

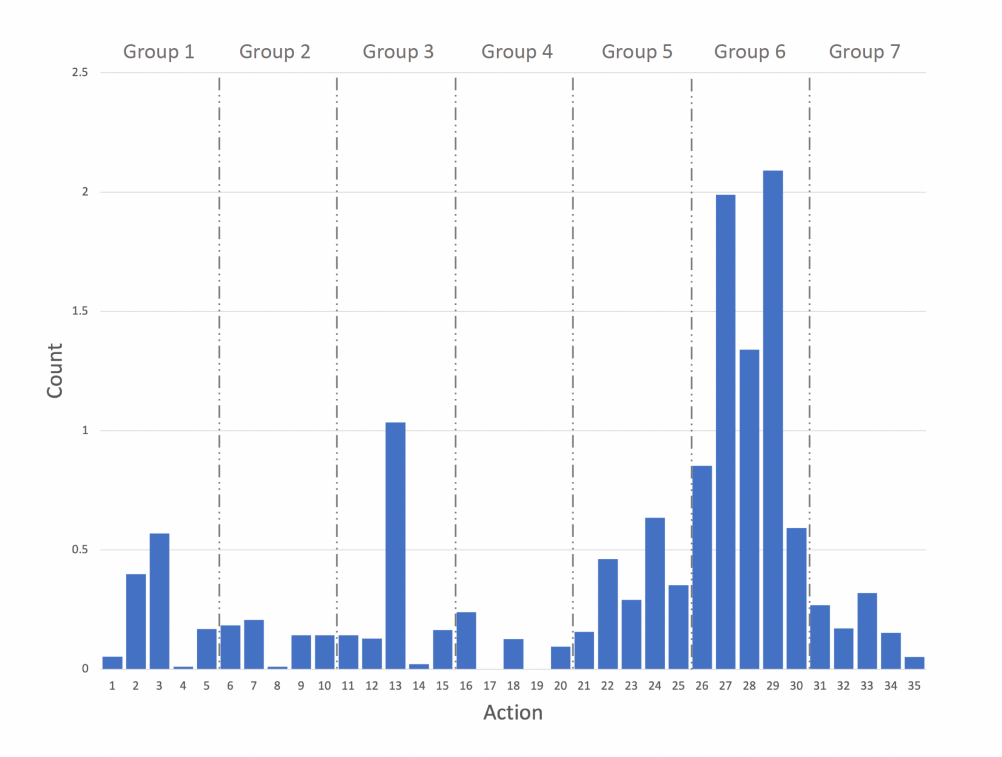


Figure 5.6: Action distribution of the Behavioural Cloning-based model.

reward obtained by the method, which is a crucial indicator of the effect of the method. The equation for computing ECR is:

$$ECR = \mathbb{E}_{s_0 \sim \mathcal{D}, \pi^*} Q\left(s_0, \pi^*\left(s_0\right)\right), \tag{5.5}$$

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 $ECR = V_{\pi^*}\left(s_0\right)$. We calculate the cumulative rewards for 100 rounds over 1,000 steps starting from the initial state. The results of the expected cumulative rewards evaluation are shown in Figure 5.9. It is clear that Sim-GAIL outperforms other baseline methods. The RL-based method performs better than the BC-based method.

5.5.3 Offline Policy Evaluation

As a robust policy evaluation method that does not require human participation, Offline Policy Evaluation (OPE) is often used to evaluate Reinforcement Learning

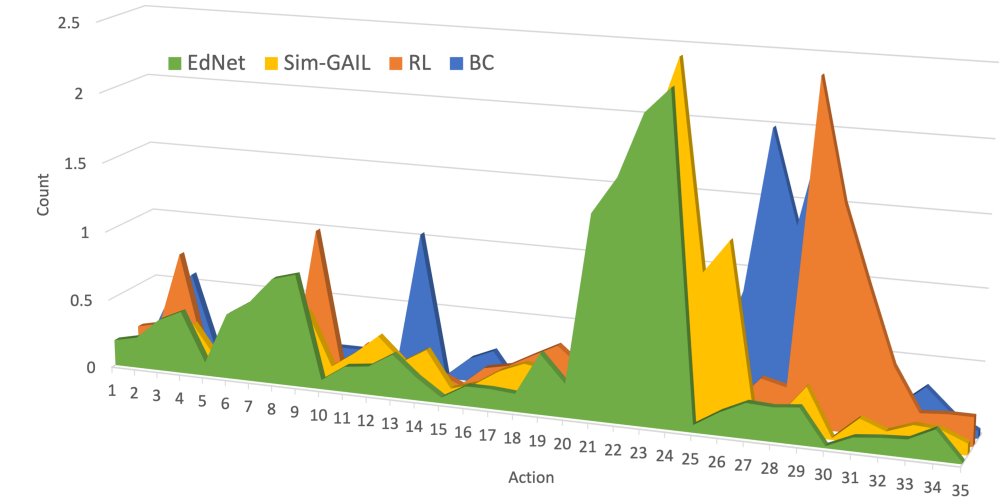


Figure 5.7: Comparison of different models' actions distribution.

(RL), which has shown great potential in decision-making tasks, such as robotics [270] and games [271]. In these tasks, RL optimal 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. [272]. 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 [273] were proposed to address these problems, where the evaluation of the policy is only based on the collected historical offline log data. It is mainly applied in scenarios where online interactions involve high-risk and expensive settings, such as stock trading, medical recommendation, and educational systems [274]. In this chapter, we employed a combination of two OPE methods: the Importance Sampling (including three variants, OIS, WIS, and

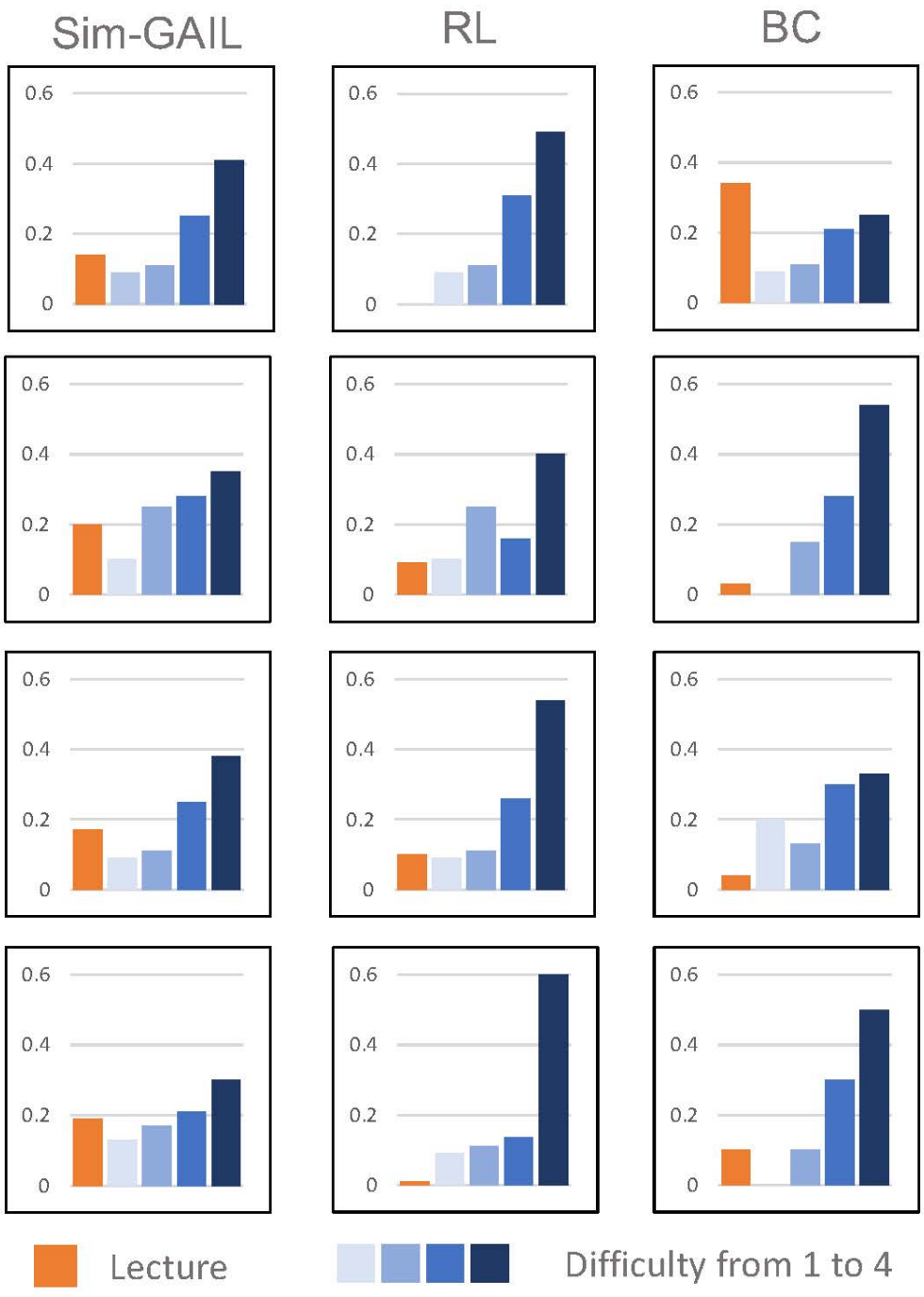


Figure 5.8: Action distribution of the state feature 'topic_fam' from simulated students generated by three different methods. The horizontal axis is the value of 'topic_fam' 1 to 4, the vertical axis is the normalised counts of the actions, the orange bar represents the lecture consumption, and the blue bar represents questions, from easy to difficult. The difficulty is represented by hue strength.

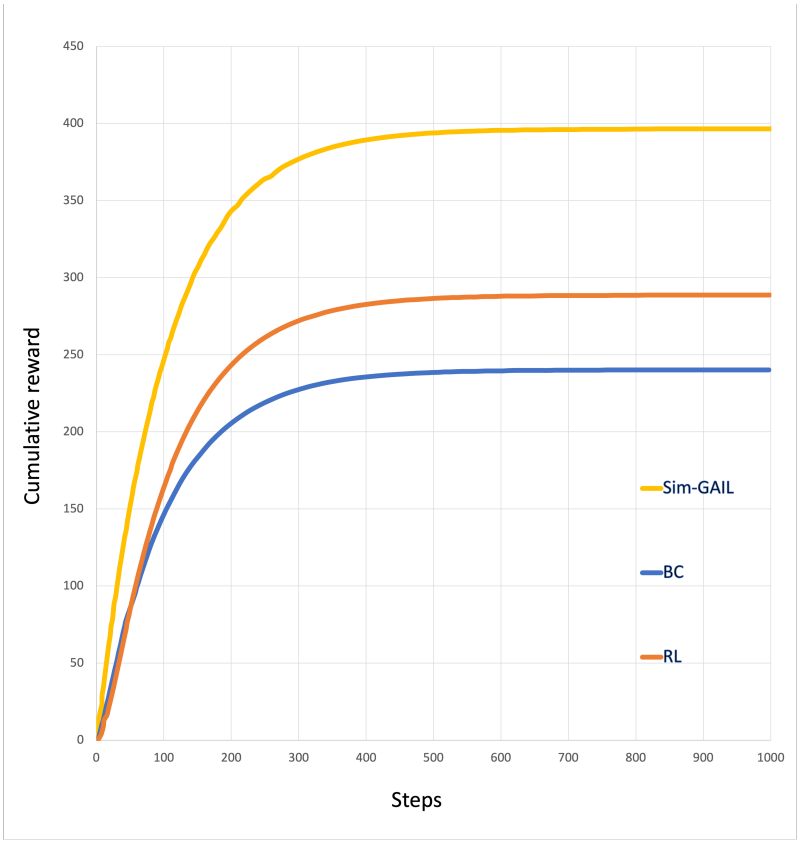


Figure 5.9: Expected Cumulative Rewards evaluation.

PIS) [275] and the Fitted Q Evaluation method [276], which could effectively test the policy performance of three models.

Importance Sampling

As one of the OPE methods, Importance Sampling (IS) is used in situations where the original distribution data is difficult to sample directly. It is a method that uses simple and collectable distribution to calculate the expected value of the desired distribution [275]. 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 [277, 278]. However, the basic IS method may result in 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

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

Table 5.4: Importance Sampling Evaluation results.

employs a weighted average to mitigate the variance [279]. The Per-Decision Importance Sampling modifies the sampling ratio and makes the reward dependent only upon the previous action in each timestamp [276]. The combination of the three methods can better observe the policy distribution of the generated data.

Table 5.5.3 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.

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 [276].

Figure 5.10 shows the Fitted Q Evaluation results on the initial state. The Sim-GAIL policy is superior to that of the other baseline models. Figure 5.11 shows the FQE loss of the three methods. Although Sim-GAIL generates the highest $Q(s0, \pi(s0))$, the validation loss is also higher and more unstable than the RL and BC-based methods. Although Sim-GAIL outperforms the two baseline methods, the parameters still need to be tuned to reduce the loss.

5.5.4 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

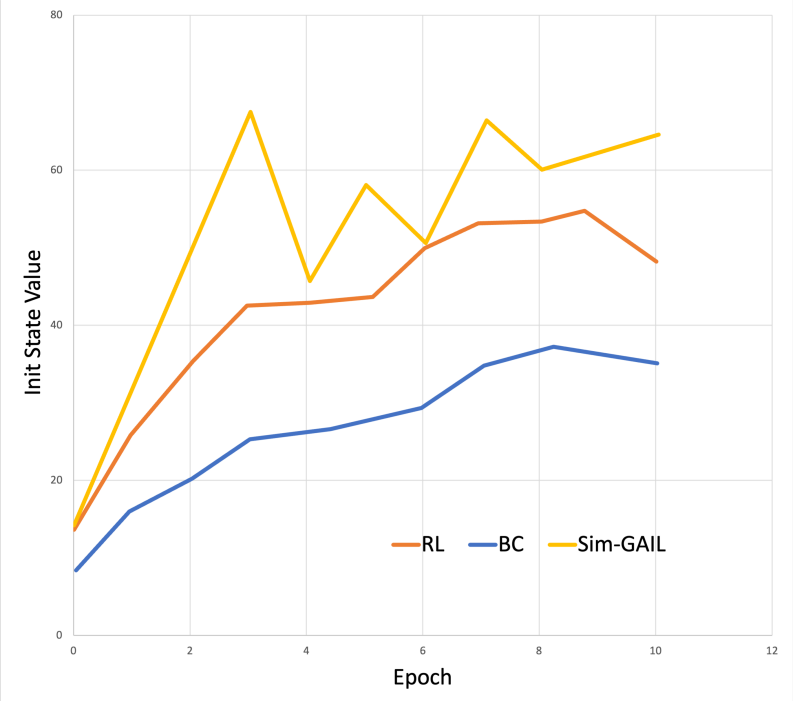


Figure 5.10: Initial State Value Estimate of the FQE.

are used to model and predict their knowledge states [17]. 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 2,500 students to train the KT model. Therefore, we assume that the volume of 2,500 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 2,500. Therefore, we selected

[3]https://www.kaggle.com/code/datakite/riiid-answer-correctness
[4]http://ednet-leaderboard.s3-website-ap-northeast-1.amazonaws.com

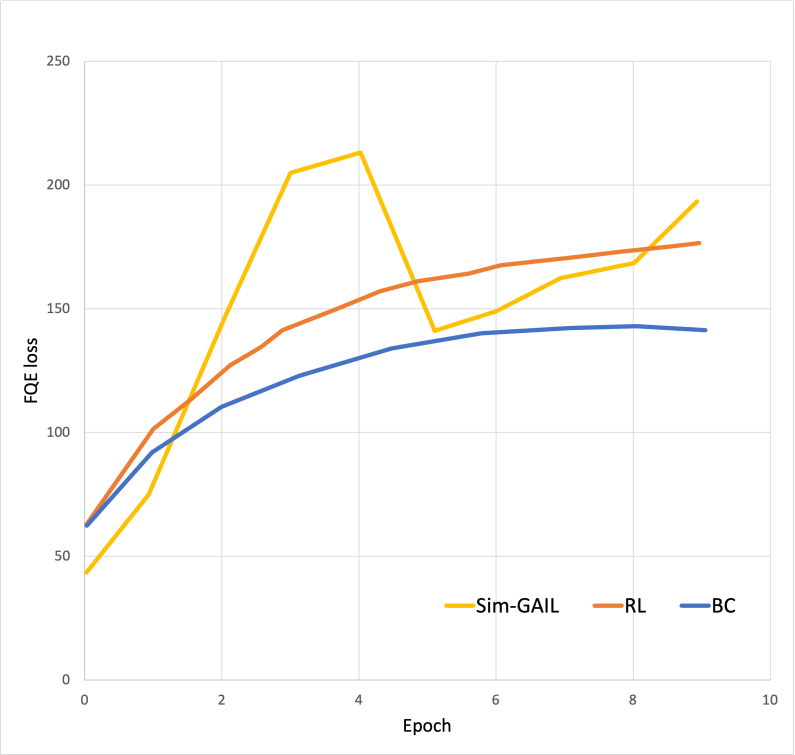


Figure 5.11: The FQE-loss.

datasets of sizes 500, 1,000, 1,500, 2,000, and 2,500 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 1,000. 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 5.12 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., 1,000, 2,000, 3,000, 4,000, and 5,000 sizes of the mixed dataset. The curve of SAINT* is higher than the curve of SAINT in the cases of 1,000, 2,000, and 3,000 sizes of data; Although the curve of SAINT* is very close to SAINT in the cases of 5,000 sizes of data, the former still outperforms the latter. In all those three pairwise comparisons, especially in the cases of smaller data sizes (1,000, 2,000, and 3,000), obviously, training on mixed data (a combination of the original and generated data) could improve the KT models. 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.

5.6 Discussion

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' policy, thus that a simple reward function built by algorithms instead of humans might result in a more optimal policy [14]. The results of the evaluation using the KT models show that Sim-GAIL could be applied in 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 doesn't match the action distribution of real students' trajectories. Besides, the distribution of 'lecture' actions whose default reward

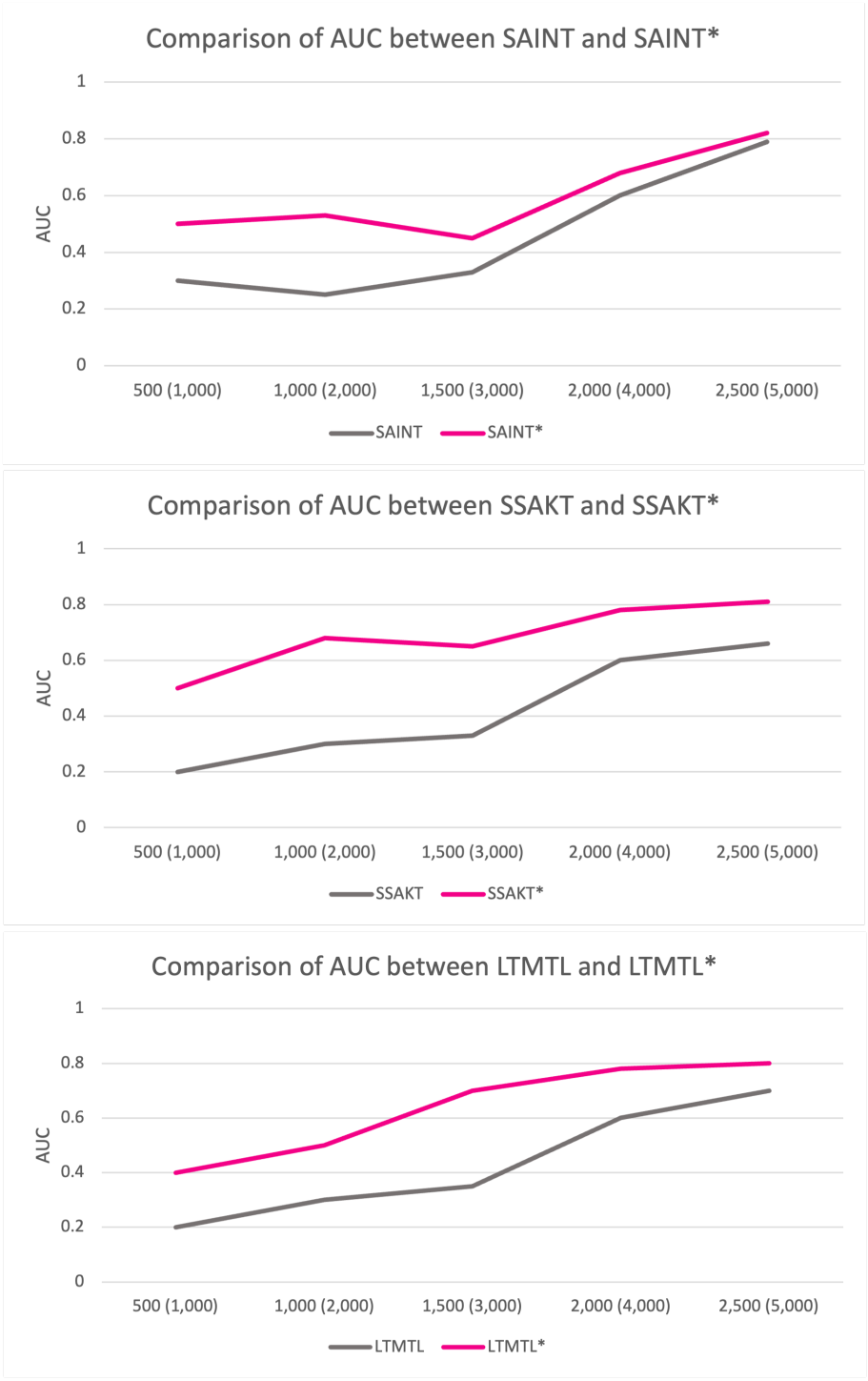


Figure 5.12: Pairwise AUC comparisons of the three KT models trained on only original students' data (SAINT, SSAKT, LTMTL, in grey) and trained on the mixed dataset (SAINT*, SSAKT*, LTMTL*, in red). On the horizontal axis, 500, 1,000,...,2,500 indicate that the grey curve model uses the original dataset, and (1,000),(2,000),...,(5,000) indicate that the red curve model uses the mixed dataset.

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 make 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 Figure 5.12, 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 2,000. After the data size is larger than 2,000, 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.

5.6.1 Comparison with SimStu

In the previous Chapter, I proposed the SimStu, a method used for simulating student behaviour, but it may exhibit suboptimal performance when dealing with reward sparsity. This means that when students have relatively few opportunities to receive rewards during the learning process, SimStu's performance might be affected because it relies on reward signals to guide simulation and prediction. In such cases, the predictive accuracy of SimStu may decrease.

In contrast, Sim-GAIL employs the Generative Adversarial Imitation Learning (GAIL) approach, which excels at addressing the issue of reward sparsity. Through

Sim-GAIL, we can generate more accurate and realistic simulated data from observed behaviours of actual students without being constrained by the scarcity of rewards. This means that Sim-GAIL is capable of better capturing reward distributions and strategies from real data, providing more reliable simulation results.

In conclusion, SimStu and Sim-GAIL each have their own strengths and limitations, with Sim-GAIL performing better in handling reward sparsity and being suitable for scenarios where more accurate student behaviour simulation is required.

5.7 Summary

This chapter has proposed Sim-GAIL, the first (to the best of our knowledge) student simulation method built upon the Generative Adversarial Imitation Learning algorithm. This method could train ITS using simulated student behaviour data to potentially alleviate the high-cost, resource-intensive, and time-consuming issues of collecting real student data and the cold start problem in early-stage ITS training.

Our student simulation method, Sim-GAIL, is constructed in the form of Generative Adversarial Imitation Learning, leveraging the EdNet dataset. The experiments show that Sim-GAIL obtains performance gains over the baseline methods: a Reinforcement Learning method based on Conservative Q-learning and an Imitation Learning method based on Behavioural Cloning. We evaluated our method from four aspects. We started with evaluating the action distribution discrepancy based on the Kullback–Leibler divergence. Then, we evaluated the reward function using the Expected Cumulative Rewards (ECR). After that, we used two Offline Policy Evaluation (OPE) methods to compare the performances of the three methods. The first OPE method was Importance Sampling and included two variants, WIS and PIS. The second OPE method was Fitted Q Evaluation, a low-variance alternative method of Importance Sampling. Compared with the baseline models, Sim-GAIL performed the best. Furthermore, we applied our method to state-of-the-art knowledge tracing models. The results indicate that our method could improve the knowledge tracing models' performance, especially in cold-start scenarios. This in turn proves the efficiency of our proposed Sim-GAIL method's ability to simulate

and generate students' historical trajectory data.

Epilogue

The study presented in this chapter has addressed *RO 3.1:* To propose a student modelling method based on Generative Adversarial Imitation Learning (GAIL) approach. The performance of the Sim-GAIL method, a novel approach, was assessed by comparing it with traditional methods that relied on Reinforcement Learning and Imitation Learning, which is addressed in the *RO 3.2.* Moreover, the effectiveness of the Sim-GAIL method in improving the prediction accuracy of the Knowledge Tracing (KT) model was investigated, which has addressed the *RO 3.3.* These research objectives aimed to contribute to the advancement of Student Modelling techniques and provided valuable insights for future research in this field. The process of addressing this research objective has answered the **Research Question 3**: How can we generate realistic and diverse simulated student behaviour data for training Intelligent Tutoring Systems (ITS) through reinforcement learning techniques?

By accomplishing these research objectives, this chapter successfully addressed the research question and provided valuable contributions to the field of ITS. The proposed Sim-GAIL method offers a promising approach to generating realistic and diverse simulated student behaviour data, thereby enhancing the accuracy and effectiveness of ITS applications.

CHAPTER 6

Broader and Deeper: A Multi-Features with Latent Relations BERT Knowledge Tracing Model

Prologue

In Chapters 5 and 6, we have focused on the Student-to-ITS process and presented two student simulation methods, SimStu and SimGAIL. By leveraging these two methods, it is possible to maximise the performance of ITS while minimising the input and interaction required from students. In this and the next chapters, we move to focus on the ITS-to-student process. The aim of this process is to assist students in achieving higher levels of learning performance. This necessitates a thorough comprehension of the learning process and the student's level of mastery of the subject matter. Therefore, developing efficient Knowledge Tracing (KT) is crucial in ITS by modelling students' learning progress and predicting their future actions based on their past behaviour data. However, traditional knowledge tracing algorithms generally use one or a few features to predict students' behaviour and do not consider the latent relations between these features, which could be limiting and disregarding important information in the features.

In this chapter, we propose MLFBK: a multi-features with latent relations BERT

knowledge tracing model, which is a novel BERT-based knowledge tracing approach that utilises multiple features and mines latent relations between features to improve the performance of the KT model. Specifically, our algorithm leverages four data features (*student_id*, *skill_id*, *item_id*, and *response_id*, as well as three meaningful latent relations among features to improve the performance: individual *skill mastery*, *ability profile* of students (learning transfer across skills), and *problem difficulty*. By incorporating these explicit features, latent relations, and the strength of the BERT model, we achieve higher accuracy and efficiency in knowledge tracing tasks. We use t-SNE as a visualisation tool to analyse different embedding strategies. Moreover, we conduct ablation studies and activation function evaluation to evaluate our model. Experimental results demonstrate that our algorithm outperforms baseline methods and demonstrates good interpretability.

Declaration: This chapter is based on the following publication:

Li, Z., Shi, L., Zhou, Y., & Wang, J. (2023, September) ***Broader and Deeper: A Multi-Features with Latent Relations BERT Knowledge Tracing Model.*** 18th European Conference on Technology Enhanced Learning, EC-TEL 2023, Aveiro, Portugal, September 4–8, 2023, Proceedings. Cham: Springer International Publishing, 2023.

This chapter is presented largely as accepted, although referencing and notation have been altered and cross-referencing added for consistency across this thesis. Some stylistic changes have been made for consistency. The majority of the text is verbatim, with some minor wording and formatting changes.

6.1 Introduction

In the preceding chapters, we introduced two student behaviour simulation methods from the student-to-AI perspective. Such approaches effectively minimize the interaction required from users to assist in the ITS training process. In the current chapter and the following one, our focus shifts to enhancing the prediction accuracy of the existing ITS system from the ITS-to-student perspective in order to

deliver more personalised and effective learning experiences for students. To achieve personalisation for student learning, ITS requires a reliable method for estimating students' knowledge state and learning progress. This method is known as Knowledge Tracing (KT). KT estimates students' knowledge state or skill mastering level based on the student's interaction data collected from ITS [15]. Accurate and efficient KT models are essential for ITS and educators to provide personalised learning experiences and support to students, such as tailored feedback, targeted hints, and relevant additional learning resources.

Generally, there are three kinds of KT models. Bayesian Knowledge Tracing (BKT), Logistic KT models, and Deep learning based Knowledge Tracing (DKT) [58]. BKT is one of the earliest and most influential KT models. It uses a probabilistic framework to model student knowledge and learning state over time [59]. Logistic KT models are developed based on the concept of logistic regression, a statistical technique utilized to model the probability of a binary outcome by utilizing one or more predictor variables. While BKT and Logistic models have achieved significant success in predicting student performance, they have also been criticized for their inability to capture the complex relationships between different skills and concepts [17]. To address this limitation, the more recent DKT models have utilised deep learning techniques to capture the complex interactions between student responses, skills, and questions [66]. DKT models have achieved state-of-the-art performance on benchmark datasets but require a large amount of training data to achieve good results.

Previous KT models have often been limited by their reliance on a single or few features, which could fail to capture the complexity of student learning behaviour data. Numerous studies have demonstrated that incorporating one or two additional features could enhance the performance of KT models [18,19]. Additionally, Minn *et al.* suggested that identifying latent features could further improve the performance of KT [280]. However, to date, there has been no research that has investigated the effectiveness of combining multiple features and latent relations to improve the performance of KT models.

Therefore, the research question of this Chapter is: *Whether incorporating mul-*

tiple features and mining the latent relations between features together could improve the accuracy and efficiency of KT models?.

In this chapter, we present the **M**ulti-**L**atent **F**eature **B**ERT **K**nowledge Tracing model that is both "broader" and "deeper" than the previous models to address the abovementioned limitation by incorporating multiple features with mined latent relations that provide richer and more diverse contextual information. By "broader", we incorporate four different types of features into our model: *student_id*, *skill_id*, *question_id*, and *response_id*. These features provide additional contextual information to help the model better capture individual differences in learning and problem-solving strategies. By "deeper", we employ a feature engineering method to extract three meaningful latent relations important for representing student's behaviour: *skill mastery*, *ability profile* of students (learning transfer across skills), and *problem difficulty*. We utilise a monotonic convolutional multi-head self-attention mechanism to combine the above explicit features and latent relations. By incorporating these explicit features and latent relations, our model could better account for the nuances and complexities of student learning and achieve superior performance compared to existing KT models. The experimental results show that MLFBK outperforms the four baseline models on five benchmark datasets. Furthermore, the t-SNE as the visualisation tool was used to analyse the interpretability of MLFBK and the embedding strategies. The experimental results show that MLFBK outperforms the baseline models and could effectively enhance the interpretability of deep learning based KT models.

The main contributions of our Chapter lie in the following three aspects:

1. We propose MLFBK, a novel **M**ulti- **F**eatures with **L**atent relations **B**ERT **K**nowledge Tracing model, which not only considers the multiple explicit features but also deeply mines the latent relations between the features by using a feature engineering method. [1]

2. Our model achieves state-of-the-art performance, outperforming four existing state-of-the-art models on five benchmark datasets. Moreover, we conduct

[1]Source code and datasets are available at https://github.com/Zhaoxing-Li/MLFBK.

ablation experiments and demonstrate different embedding strategies with a visualisation tool to investigate the contribution of different latent relations.

3. MLFBK exhibits good interpretability as a deep learning based KT method and has advantages in training efficiency.

6.2 Related Work

This Chapter aims to present a novel BERT-based KT model that incorporates multi-features and latent relations. Therefore, we first review the cornerstone of the BERT model – the Transformer based models and their applications. Then we review the development of KT methods in general. At last, we review existing KT methods from the perspective of the number of integrated features.

6.2.1 Transformer-based Models and Application

The Transformer architecture, proposed by Vaswani *et al.* [40], is a type of neural network that has gained widespread popularity in natural language processing (NLP), and other domains due to its ability to effectively model long-range dependencies and capture complex patterns in sequential data. Transformers have been used in various NLP tasks, including language translation, question answering, and text classification, and have achieved state-of-the-art performance on many benchmarks [281].

In addition to NLP, Transformers have also been applied in other domains, such as computer vision [282], speech recognition [231], and recommendation systems [283]. For example, the Vision Transformer (ViT) has recently been proposed as an alternative to convolutional neural networks (CNNs) for image classification tasks, achieving competitive performance on several benchmark datasets.

Besides the basic Transformer models, many powerful evolutions of Transformer-based methods were proposed, such as the GPT [130] and BERT [41]. The well-known ChatGPT originated from GPT [284]. BERT (Bidirectional Encoder Representations from Transformers), introduced by Devlin *et al.* [41], is a pre-trained Transformer-based language model that has achieved state-of-the-art performance

on various NLP tasks. BERT utilises self-attention and masked language modelling (MLM) techniques to train the Transformer bidirectionally. Its remarkable ability to process natural language text effectively and generate high-quality embeddings has made it a popular choice and a superior performer in many Deep Learning tasks [281]. BERT has also been adapted to various other fields with excellent results. For instance, ConvBERT utilises the original BERT architecture in image processing task [42], BERT4Rec enhances the performance recommendation systems [43], and LakhNES improves the quality of music generation by incorporating BERT [44].

6.2.2 Knowledge Tracing

Knowledge Tracing is a technique utilised in educational data mining that aims to model students' knowledge state and mastering level of the learning concepts or subjects [227]. Generally, the KT models could be classified into three categories based on the different structures of the modelling approach that the model used: probabilistic models, logistic regression KT models, and deep learning-based models [17].

Probabilistic KT models assume a student's learning process follows a Markov process. They use a probabilistic graphical model such as Hidden Markov Model (HMM) or Bayesian Belief Network to track their changing learning states [285]. Bayesian Knowledge Tracing (BKT) is a classic probabilistic model that has been used for this purpose, but it has several limitations: BKT does not account for the complexity or difficulty of concepts and skills and assumes that each question requires only one skill. This makes it difficult to process complex problems involving multiple skills and complex relationships between concepts, questions, and skills. To address these limitations, researchers have proposed models including Dynamic BKT (DBKT), which uses Dynamic Bayesian Network (DBN) to model prerequisite hierarchies and dependencies of multiple skills [286]. The logistic KT models are based on the principle of logistic regression, which is a statistical method used to estimate the probability of a binary outcome by using one or more predictor variables. However, both BKT and logistic KT models struggle to process multiple topics or

skills and fail to account for other features that may impact student learning.

To overcome these limitations, researchers have turned to deep learning technologies to develop Deep Knowledge Tracing (DKT) [285]. DKT models a knowledge tracing task as a sequence prediction problem and has shown promise in achieving better performance than BKT and logistic KT models. The self-attention mechanisms were widely used in deep learning architectures, which have also been applied to KT models, resulting in models such as SAKT [69] and SAINT+ [236]. These methods have achieved higher performance than traditional DL-based methods. More recently, several BERT-based methods were proposed that achieved state-of-the-art performance. BEKT [45] is a deep knowledge tracing with bidirectional encoder representations from transformers. MonaconBERT [287] utilised the monotonic attention based ConvBERT to improve the knowledge tracing.

6.2.3 KT models with different feature numbers

Single feature KT models Single-feature KT models use only one feature, usually exercise or skill, to predict a student's knowledge or mastery of a particular skill or concept. Deep Knowledge Tracing (DKT) [46] and Self-Attentive Knowledge Tracing (SAKT) [69] are examples of single-feature models that have been proposed to improve performance by using different techniques, such as LSTM networks and attention mechanisms to deal with the sparsity of exercise data.

Double-feature KT models Double-feature KT models use both exercise and skill features, resulting in significant performance gains compared to single-feature models. Deep Hierarchical Knowledge Tracing (DHKT) [288], Bi-Interaction Deep Knowledge Tracing (BIDKT) [289], and Attentive Knowledge Tracing (AKT) [68] are examples of double-feature models that have been proposed to improve performance by modelling the hierarchical relations between skills and exercises and proposing new attention mechanisms and embedding methods.

Multi-feature KT models Multi-factor KT models integrate multiple learning-related factors into the model to improve performance. Exercise-aware Knowledge

Tracing (EKT) [16] and Relation-aware self-attention Knowledge Tracing (RKT) [290] are examples of multi-feature models that have been proposed to integrate information such as the exercise-making sequence, the relations between skills and time delay since the last interaction, and the text information of the exercise content.

6.3 Methodology

6.3.1 Problem Statement

The goal of knowledge tracing is to use a series of interaction data from Online Learning Systems (OLS) or Intelligent Tutoring Systems (ITS) to predict the correctness of a student's next answers. The student's interactions are represented by a data sequence, denoted as $x_1, ..., x_t$, where $t-th$ is represented as $x_t = (qt, at)$. Here, q_t refers to the $t-th$ question and indicates whether the student's answer is correct (1) or not (0).

6.3.2 Proposed Model Architecture

We propose a novel Knowledge Tracing model, Multi Features with Latent Relations BERT Knowledge Tracing (MLFBK), to improve the traditional KT models by incorporating multi-features and mining latent relations between different features in the student historical interaction data. Fig. 7.1 shows the architecture of MLFBK, which consists of three parts: embedding on the left; BERT-based architecture in the middle; the correctness sequence output on the right. The embedding part on the left further contains two components: the Multi-Features embedding on the top, and the Latent-Relations embedding at the bottom.

Multi-Features Embedding

In a general ITS system, student characteristics such as *student_id*, *skill_id*, *question_id*, and *response_id* are recorded in the dataset. However, the hidden information behind these features is also valuable to be mined. By mining the hidden latent relations, the prediction efficiency of the KT model could be greatly improved. In

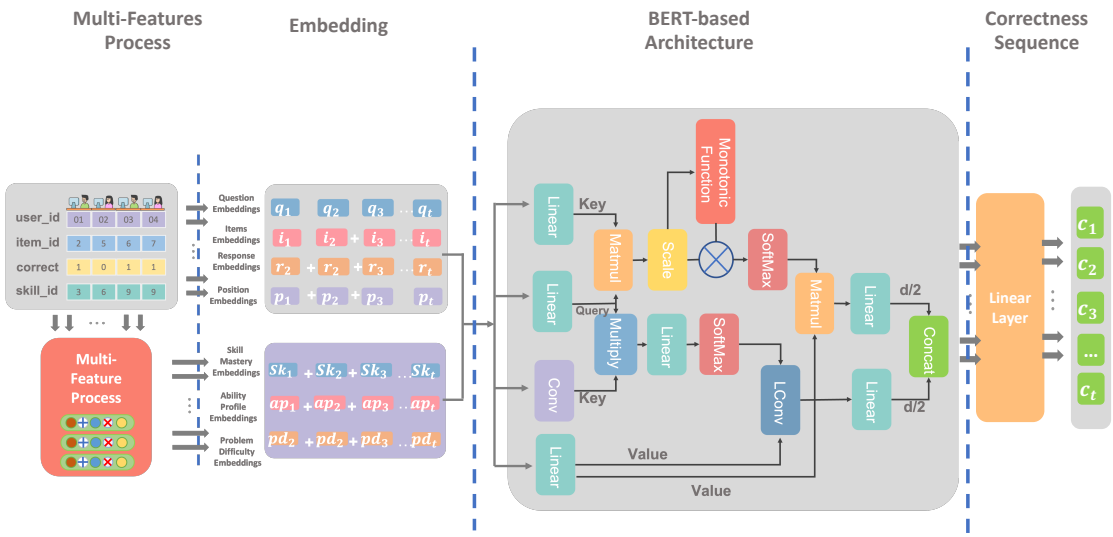


Figure 6.1: The architecture of MLFBK. MLFBK consists of three parts: 1) the multi-features process (on the left), 2) the BERT-based architecture (in the middle), and 3) the correctness sequence output part(on the right).

this Chapter, we utilise three meaningful latent relations: individual skill mastery, ability profile of students (learning transfer across skills), and problem difficulty. We utilise conventional machine learning techniques, such as hidden Markov models and K-means clustering, to extract meaningful features.

Therefore, in the Multi-Features Embedding part, we incorporate four different types of features into our model: *student_id*, *skill_id*, *question_id*, and *response_id*. Particularly, *student_id* is utilised to generate the interaction sequences. It is also used in the Latent Relation Embedding component for calculating the *skill mastery* embedding and the *ability profile* embedding. These features need *student_id* to keep track of a single student.

Latent Relations Embedding

In the Latent Relations Embedding component, we use a feature engineering method proposed by work [280]. Using this method, we mine three different meaningful latent relations among the behaviour data of individual students: *skill mastery*, *ability profile* (learning transfer across skills), and *problem difficulty*.

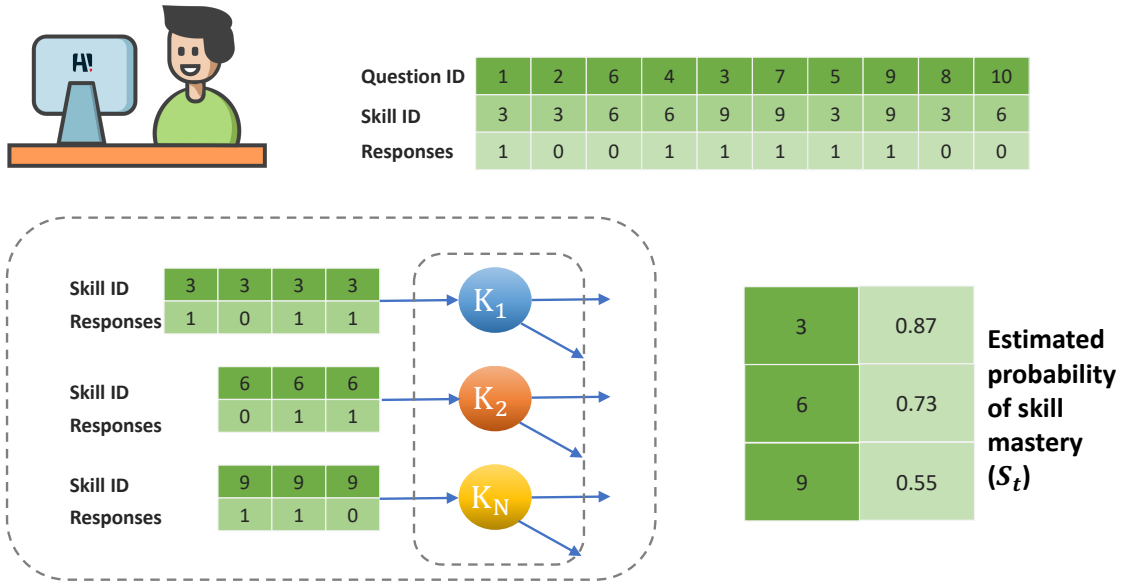


Figure 6.2: Estimated the probability of skill mastery at each timestamp.

Skill Mastery. The formulation of skill mastery is based on the Bayesian Knowledge Tracing (BKT) model, which uses four parameters to represent probabilities related to a student's mastery of a skill. These parameters include $P(L_o)$, the probability that a student masters the skill before attempting the first problem associated with it; $P(T)$, the probability that a student will master the skill after the next practice opportunity; $P(G)$, the probability that a student guesses the correct answer to a question despite not knowing the skill; and $P(S)$, the probability that a student answers a question incorrectly despite knowing the skill. Skill mastery is the probability of learning a skill rather than the probability that a student applies the skill correctly. A BKT model is trained for each skill, and the inputs to each skill model are the binary responses of a student on that single skill. Fig. 6.2 shows the Skill Mastery mining process.

Ability Profile. Students' interactions are divided into multiple time intervals, and past performance is encoded to estimate their ability profile. The ability profile is encoded as a cluster ID and updated after each time interval using all previous attempts on each skill. The K-means algorithm is used to evaluate the temporal long-term learning ability of students in both training and testing at each time

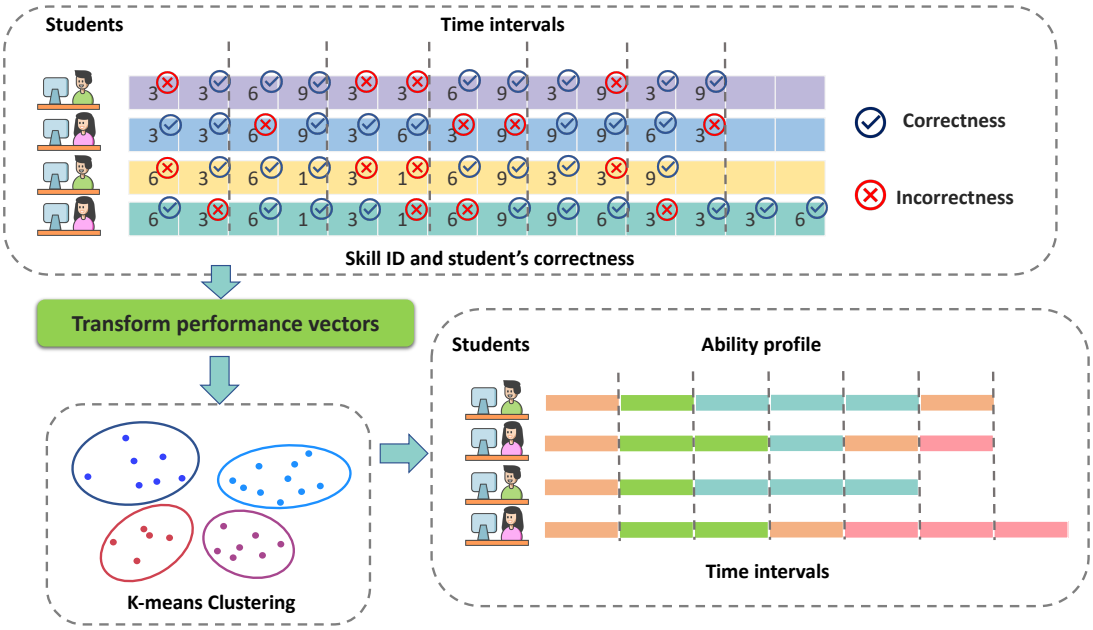


Figure 6.3: Estimated the probability of skill mastery at each timestamp.

interval. Fig. 6.3 shows the ability profile extraction process.

Problem Difficulty. This is calculated on a scale between 1 and 10, with 1 being the easiest and 10 being the most difficult. We use function 6.1 to map the average success rate of a problem onto the 10-level scale. The problem difficulty (p_j) could be calculated as:

$$\delta\left(p_j\right) = \left\lfloor \frac{\sum_i^{|N_j|} O_i\left(p_j\right)}{|N_j|} \cdot 10 \right\rfloor \tag{6.1}$$

where P_j is the j^{th} problem. N_j is the set of the students who tried to solve problem p_j. $O_i\left(p_j\right)$ is the first attempt of student i to solve problem p_j. Problems with a higher success rate are considered easier, while problems with a lower success rate are considered more difficult.

Overall, in the first part of our model, we incorporate question embedding E_q, items embedding E_i, response embedding E_r, learnable positional embedding E_{pos}, skill mastery embedding E_{SK}, ability profile embedding E_{AP}, and problem difficulty embedding E_{PD}. The final input embedding is denoted as:

$$E_{\text{input}} = E_q + E_i + E_r + E_{pos} + E_{SK} + E_{AP} + E_{PD}. \tag{6.2}$$

BERT based Architecture

The second part of our proposed architecture is a BERT-based method (shown in Fig. 7.1). Earlier studies [291] have indicated that training the Transformer model can be challenging unless a specific training strategy, like a warm-up start, is employed. In contrast, the pre-LN Transformer doesn't require a warm-up start and converges to a solution much more rapidly than the original Transformer [292]. Additionally, our approach involved a distinct strategy for both training and testing. During training, our proposed model predicted the positions to be masked. The masking proportions used in training were identical to those used in the original BERT model, with 15% of the embeddings masked, of which 80% were genuinely masked, 10% were reversed, and 10% remained unchanged.

The encoder blocks use the pre-LN Transformer architecture with 12 layers to normalise the input vectors E_{input} . The pre-LN can be formulated as follows:

$$z = LN_{\text{pre}}\left(E_{\text{input}}\right) \tag{6.3}$$

The normalized value z is then transformed into the query, key, and value of monotonic convolutional multi-head attention. This result is passed through a dropout layer and added to the embedding vectors as a residual connection.

$$a = x + D(\text{MonoConvMulAttn}(z, z, z)) \tag{6.4}$$

The output is normalized and passed through fully connected layers with a LeakyReLU activation function. The results are again normalized through a dropout layer, and the second result is added as a residual connection.

$$fc = W_{fc2}\left(\text{ LeakyReLU }\left(W_{fc1}\right)\right) \tag{6.5}$$

Moreover, we utilise a monotonic convolutional multi-head attention proposed by [287], which is combined with mixed-attention and monotonic attention, to represent forgetting in sequence data. Monotonic multi-head attention uses an exponential decay mechanism to measure the distance between sequences, while span-based

dynamic convolution uses a lightweight convolution to combine query and key vectors.

6.3.3 Experiment Setting

Datasets

We adopted four benchmark datasets to validate the performance of the MLFBK model, including EdNet [235] [2], assist09[3], assist12[4], algebra06 [5].

Baseline Models

In this study, we evaluated the performance of our MLFBK model by comparing it with three state-of-the-art models: MonaCoBERT [287], BEKT [45], and AKT [68], as well as the top two baseline models (SSAKT and LTMTI) in the Riiid Answer Correctness Prediction Competition hosted on Kaggle[6].

Evaluation Metrics and Validation

We used the area under the curve (AUC) as the evaluation metric to compare the model's performance on four benchmark datasets. After that, we conducted an activation function evaluation to compare the different activation functions. We also conducted an ablation study to identify the contribution of different latent relations. Furthermore, we applied t-SNE as the visualisation tool to evaluate our method's embedding strategies and interpretability.

Hyperparameters for Experiments

For a fair comparison, all baseline models were trained using the same set of parameters. Specifically, training was conducted with a batch size of 64 and a train/test

[2]https://github.com/riiid/ednet
[3]https://sites.google.com/site/assistmentsdata/home
[4]https://sites.google.com/site/assistmentsdata/home
[5]https://pslcdatashop.web.cmu.edu/KDDCup
[6]https://www.kaggle.com/code/datakite/riiid-answer-correctness

split ratio of 0.8/0.2. The model was trained for 100 epochs with the Adam optimizer and a learning rate of 0.001. The loss function used was binary cross-entropy, and the model utilized a total of 12 encoder layers with a hidden size of 512 and 8 attention heads. The data was preprocessed by splitting it into interaction sequences with a maximum length of 100. In cases where a student had less than 100 interactions, the remaining sequence was padded with zeros. For students with more than 100 interactions, the sequence was split into multiple subsequences of length 100.

6.4 Results and Discussion

6.4.1 Overall Performance

Table 7.3 presents the comparison results of MLFBK with five other KT models, including MonaCoBERT, BEKT, AKT, SSAKT, and LTMTL, on four benchmark datasets, including EdNet, assist09, algebra06, and assist12. It is clear from Table 7.3 that MLFBK outperforms the other five KT models on all four datasets in terms of AUC, indicating that MLFBK is a promising method for KT. Take the algebra06 dataset as an example: MLFBK achieves an AUC of 0.8327, which is 1.4%, 2.9%, 3.9%, 5.2%, and 2.2% higher than the AUC values of MonaCoBERT, BEKT, SSAKT, LTMTI, and AKT, respectively. The average improvement on this dataset is 3.12%. MonaCoBERT and BEKT also perform relatively well, with AUC values close to those of MLFBK on some datasets. SSAKT and LTMTI, on the other hand, have lower AUC values, indicating weaker performance. The results suggest that MLFBK is a competitive method for knowledge tracing and could potentially improve the accuracy of student modelling by incorporating more student action features and mining latent relations.

6.4.2 Ablation Study

In order to identify the contribution of each latent relation in the MLFBK model to the overall performance, we conducted an ablation study. The results are sum-

Table 6.1: Comparison of different KT models on five benchmark datasets. The best performance is denoted in bold.

Dataset	Metrics	MLFBK	Monaco	BEKT	SSAKT	LTMTI	AKT
EdNet	AUC	**0.8278**	0.7336	0.8204	0.7981	0.8023	0.7982
assist09	AUC	**0.8524**	0.8059	0.8227	0.6754	0.8132	0.7691
algebra06	AUC	**0.8412**	0.8201	0.8165	0.7937	0.7915	0.8143
assist12	AUC	**0.8350**	0.8132	0.7167	0.7356	0.6834	0.8034

marised in Table 6.2. MLFBK* in the table indicates the basic model structure with explicit features. *ap* represents ability profile, *sm* represents skill mastery, and *pd* represents problem difficulty. Table 6.2 also shows the AUC values for different versions of MLFBK* that were trained with different combinations of latent relations. It is clear that the performance of the MLFBK model is influenced by the different embedding strategies used for different relations. The models incorporating all three latent relations achieved the highest AUC values on three of the four datasets, except the assist09. Nevertheless, it also achieved the second highest score in the assist09.

The *problem difficulty* contributed significantly to the model's performance, with the models that used only the problem difficulty embedding achieving the highest AUC values on four datasets compared to other single latent relation embeddings. The combination of *problem difficulty* and *ability profile* achieved the best performance on the assist09 dataset and the second-highest performance on EdNet, indicating that the combination of these two latent relations has more weight in the predictions. The *skill mastery* feature had a comparatively lower impact on the model's performance, with the models that used only the *skill mastery* feature achieving the lowest AUC values on four datasets. It may be caused by *skill mastery* levels across students being relatively uniform or lacking a solid correlation with our target variable, student performance. In such cases, including the *skill mastery* feature may not have significantly improved our model's predictions. Moreover, there is the possibility of redundancy or high correlation between the *skill mastery* feature and other features in our dataset, such as ability profiles or problem difficulty. If these features capture similar information, the *skill mastery* feature's inclusion might not have provided substantial additional value to our model. However, the models that

used a combination of features achieved higher AUC values than the models that used a single feature, indicating that the three latent relations are complementary to each other.

Overall, the ablation study results suggest that the MLFBK model's performance could be effectively improved by incorporating multi-features and multiple latent relations. The more features and/or latent relations embeddings were incorporated, the higher AUC scores could be achieved.

Table 6.2: Ablation Study of MLFBK. The abbreviations used in there are as follows: *ap* for ability profile, *sm* for skill mastery and *pd* for problem difficulty. The best performance is denoted in bold.

Model	EdNet	assist09	algebra06	assist12
MLFBK*	0.7221	0.8002	0.7997	0.8065
MLFBK* + *ap*	0.7503	0.7922	0.8139	0.7713
MLFBK* +*sm*	0.7454	0.7891	0.7983	0.7611
MLFBK* +*pd*	0.8194	0.8411	0.8256	0.8304
MLFBK* +*ap* + *sm*	0.7429	0.8078	0.8201	0.7989
MLFBK* +*ap* + *pd*	0.8270	**0.8560**	0.8344	0.8287
MLFBK* +*sm* + *pd*	0.8233	0.8445	0.8362	0.8216
MLFBK* +*ap* + *sm* + *pd*	**0.8278**	0.8524	**0.8412**	**0.8350**

Activation Function evaluation

To investigate the impact of activation functions on the performance of our MLFBK model, we conducted a study where we tested our model with three different activation functions: Leaky ReLU, Sigmoid, and Linear. Fig. 6.7 shows the results of activation function evaluation. We trained and validated the models for 50 epochs with early stopping. Upon analysing the results, we found that all three activation functions produced similar results in terms of both training and validation behaviour. However, the Linear activation function performed slightly better than the other two. Specifically, it had the highest accuracy and AUC score on the validation set, which indicates that it may be the most suitable activation function for our MLFBK model. It is worth noting that the Leaky ReLU activation function stopped early during the training process, which may be due to its high learning rate. Overall, our findings suggest that the choice of an activation function has a

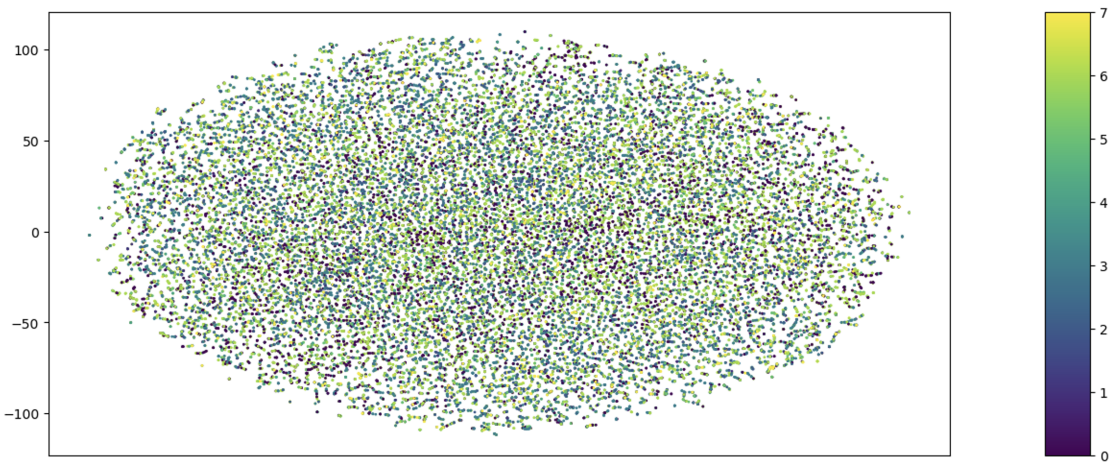


Figure 6.4: The general embedding strategy, utilizing t-SNE as the visualization method.

relatively minor impact on the performance of our MLFBK model, but using the Linear activation function may lead to slightly better results.

Analysis of Embedding Strategy

We conducted a t-SNE analysis to visualize the entire embedding vector created in our MLFBK model. The results show the good interpretability of our methods' embedding strategies. Fig. 6.4 and Fig. 6.5 show the comparison of general embedding and MLFBK embedding strategies, utilizing t-SNE as the visualization method. Here, we take the embedding strategy of AKT as an example (shown in Fig.6.4) and our MLFBK embedding strategy Fig.6.5 on the assistments09 dataset. Each data point in the plot represents a learning interaction associated with a student, question, response, correctness, item, ability profile, skill mastery, and problem difficulty. The data points were coloured based on the ability profile value associated with them, specifically the transfer across skills value for the relevant student at the relevant time.

The Fig. 6.4 shows the general embedding could not distinguish different features as all the features mixed together. In contrast, Fig. 6.5 shows that the MLFBK embedding strategy could distinguish different embedding with different colours well.

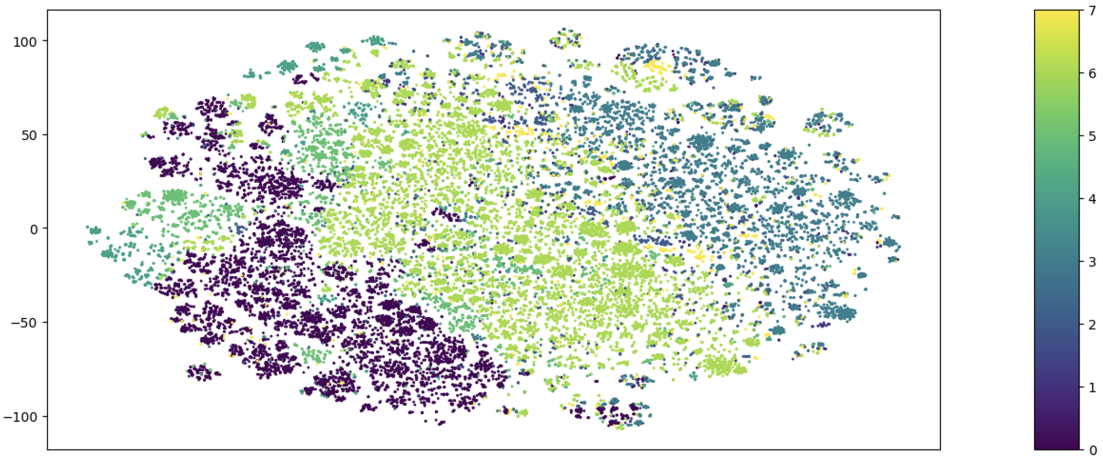


Figure 6.5: The MLFBK embedding strategies, utilizing t-SNE as the visualization method.

The t-SNE plot shows that the students with small ability profile values at the current interaction were grouped together by the embedding, as were students with large values. This grouping could be used by the model to differentiate between interactions with correct responses and incorrect responses. The ability profile values provide additional information about students' performance, which could be useful for predicting their future performance. Overall, the t-SNE analysis demonstrated the effectiveness of the MLFBK model in capturing and utilizing complex student interaction data.

Fig. 6.6 shows the different embedding strategies of different single latent relations. The top is the embedding for the ability profile. It is the embedding without the additional features and then coloured according to the problem difficulty. It is easy to see that the problem difficulty feature is heavily considered in the feature embedding. The middle is embedding for problem difficulty. It only colours the learning interactions based on the problem difficulty of the relevant question. In this figure, the embedding doesn't seem to generate groupings, but more of a constant gradient, where the more difficult problems are in the top left and the easier problems are in the bottom right. The bottom image is the embedding for skill mastery. Here the skill mastery of the student on the relevant *item* is highlighted.

This feature is multiplied by 100 and rounded to convert it to a categorical feature instead of a continuous one. The embedding also seems to be a gradient instead of groupings.

Analysis of Estimating Problem Difficulty

We compared our model with MonaCoBERT regarding estimating problem difficulty for specific questions in the assistments2009 dataset. While MonaCoBERT uses a classical test theory (CTT) approach to estimate difficulties, our model calculates problem difficulty as a feature to use as input for the BERT model. The comparison was visually represented in Fig.6.8, with difficulty levels on the x-axis and the number of students answering correctly (green) or incorrectly (red) on the y-axis. Common sense dictates that harder questions should have more incorrect answers, although there may be exceptions. Surprisingly, the MonaCoBERT method showed that many students answered easy questions incorrectly but answered more difficult questions correctly, which seemed unlikely given the number of students evaluated. In contrast, our model revealed that as question difficulty increased, fewer students answered correctly, aligning with expectations. The results show that our method of estimating problem difficulty is far superior to the CTT difficulty estimation used by MonaCoBERT. Our method provides much more predictive value in estimating problem difficulty. This highlights the effectiveness of using our MLFBK model in predicting student performance in educational settings.

6.5 Summary

In this chapter, we have proposed MLFBK, which employs a BERT-based architecture incorporating multi-features and latent relations to improve the performance of Knowledge Tracing models. Experimental results show that MLFBK outperforms the five baseline models in every benchmark dataset on the metric of AUC. Moreover, we conducted an ablation study for different embedding strategies. The results indicate that combining different features and latent relations could improve performance effectively. Incorporating additional embeddings resulted in increased

Figure 6.6: Different Embedding Strategies.

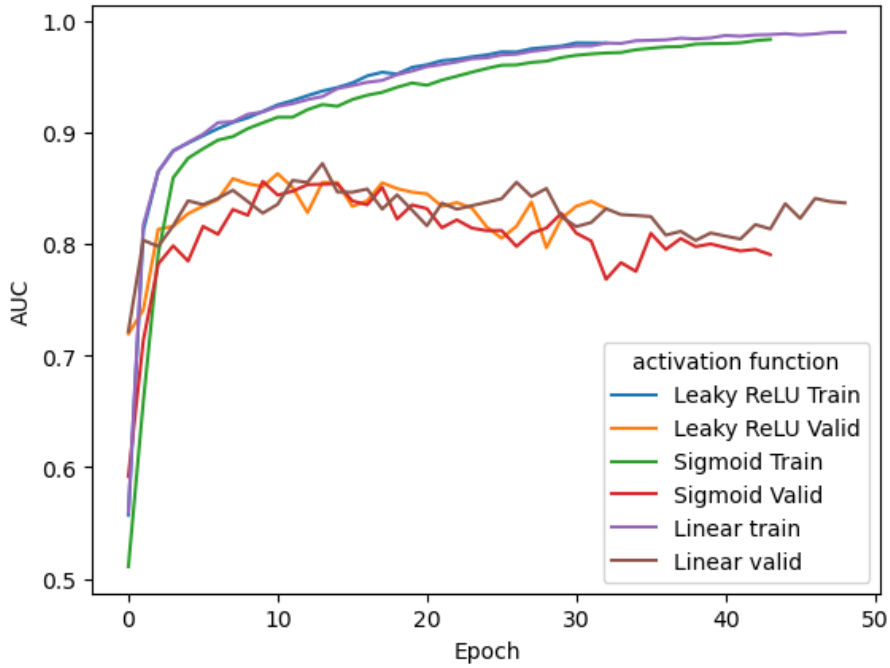


Figure 6.7: Activation Function evaluation.

AUC scores. Moreover, we utilise the t-SNE as the visualisation tool to compare different embedding strategies. The results show that our method not only improves the performance of the models but also improves the model's interpretability.

Epilogue

The approach presented in this chapter succeeds in including identifying relevant features and latent relations in student interaction data that have the potential to contribute to the improvement of KT models *RO 3.1*. We developed a novel BERT-based method to integrate these multiple features and latent relations into KT models to address *RO 3.2*. We have conducted an extensive evaluation to assess the effectiveness of the Sim-GAIL method in enhancing the prediction accuracy of the Knowledge Tracing (KT) model as outlined in *RO 3.3*.

By accomplishing these research objectives, this chapter successfully addressed Research Question 4: "How can multiple features and latent relations in the student interaction data be integrated to improve the accuracy and efficiency of Knowledge Tracing (KT) models to improve the efficiency of ITS?" The proposed methods offer

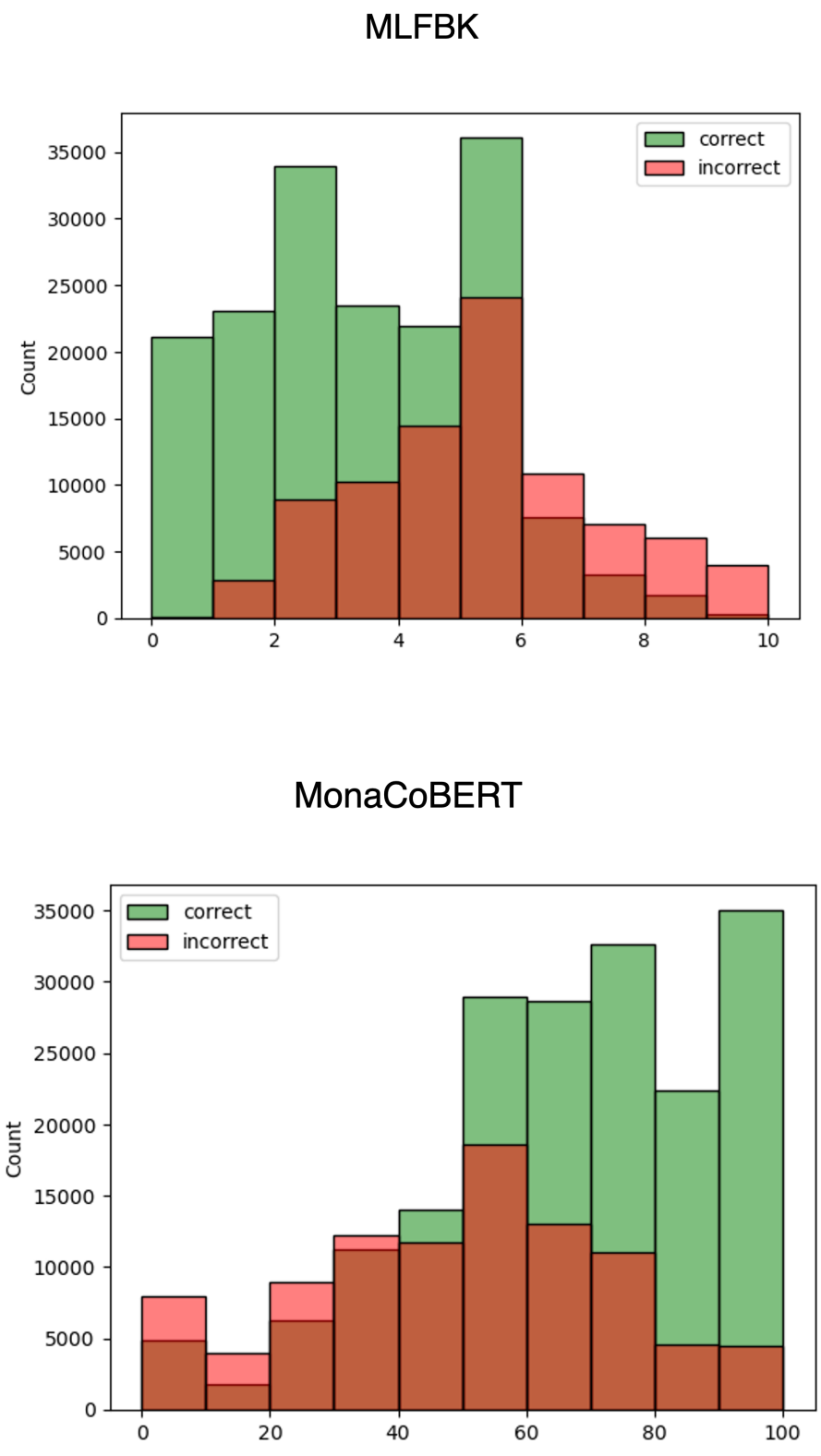


Figure 6.8: Estimating Problem Difficulty.

promising approaches to improving the accuracy, efficiency, and interpretability of KT models, thereby facilitating more effective and personalised learning experiences in ITS.

CHAPTER 7

LBKT: A LSTM BERT-based Knowledge Tracing for Long-Sequence Data

Prologue

In Chapter6, we have presented MLFBKT, which is a novel Knowledge Tracing method based on BERT. It leverages multiple features and explores latent relationships between these features to enhance the performance of the KT model. Another challenge faced by the development of KT is that, with the development of ITS, large-scale datasets containing long-sequence data began to emerge. Recent deep learning based KT models face obstacles such as low efficiency, low accuracy, and low interpretability when dealing with large-scale datasets containing long-sequence data.

To address these issues and promote the sustainable development of ITS This chapter proposed a **L**STM **B**ERT-based **K**nowledge **T**racing model for long sequence data processing, namely **LBKT**, which uses a BERT-based architecture with a Rasch model-based embeddings block to deal with different difficulty levels information and an LSTM block to process the sequential characteristic in students' actions. LBKT achieves the best performance on most benchmark datasets on the

metrics of ACC and AUC. Additionally, an ablation study is conducted to analyse the impact of each component of LBKT's overall performance. Moreover, we used t-SNE as the visualisation tool to demonstrate the model's embedding strategy. The results indicate that LBKT is faster, more interpretable, and has a lower memory cost than the traditional deep learning based KT methods.

Declaration: This chapter is based on the following publications:

Li, Z., Shi, L., Zhou, Y., & Wang, J. (2023, November) ***LBKT: A LSTM BERT-based Knowledge Tracing for Long-Sequence Data***. The 30th International Conference on Neural Information Processing (ICONIP2023). ***Submitted and Under Review***

This chapter is presented largely as the manuscript submitted, although referencing and notation have been altered, and cross-referencing has been added for consistency across this thesis. Some stylistic changes have been made for consistency. The majority of the text is verbatim, with some minor wording and formatting changes.

7.1 Introduction

The development of online learning systems has made it possible to use Intelligent Tutoring Systems (ITS) to store and analyse a sizable amount of student behaviour data to improve intelligent educational services. As one of the widely applied TEL technologies, Knowledge Tracing (KT) has drawn a lot of attention. KT is the field of modelling students' learning trajectories and predicting their sequential actions based on historical interaction data between students and ITS [15].

With the development of ITS, large-scale datasets such as *EdNet* [235] and *Junyi Academy* [293] began to emerge. In these datasets, long-sequence student interaction data were gathered as an increasing number of students used the ITS for an extended period. The long- and short-sequence data in these datasets are unbalanced, which satisfies the long-tail distribution [21]. For instance, within the EdNet dataset, a substantial amount of student action sequences are included, ranging from the shortest sequence that may comprise just a single action to the longest sequence

that encompasses 40,157 actions. Notably, the average action sequence length of the EdNet dataset is 121.5, indicating a moderate length of data sequences overall. However, it is important to note that the distribution of sequence lengths is highly skewed, and this unbalanced distribution has an impact on the overall performance of the KT models [20]. Although the quantity of short-sequence data is larger than the long-sequence data, the latter is of more weight than the former in prediction tasks [21]. Studies have demonstrated that the precision of conventional deep learning-based knowledge tracing models diminishes when handling sequences exceeding 400 actions. Yet, as ITS gains wider adoption, longer sequence data becomes increasingly prevalent, thereby presenting significant challenges for knowledge tracing model predictions [285].

In general, KT models could be divided into three categories: probabilistic KT models, logistic KT models, and deep learning based KT methods (DKT) [17]. Traditional probabilistic KT models and logistic KT models are forced to confront difficulties such as decreased processing efficiency and increased memory usage as growing amounts of longer sequence data are released. Deep learning based KT models are known to suffer from inefficiencies when processing long-sequence action data problems, including issues related to the accuracy, speed, and memory usage [20–22]. Therefore, allowing the processing of very long sequence data is key to achieving high performance for next-generation KT models [20]. Moreover, due to the black-box nature of traditional deep learning methods, the current deep learning based KT models also struggle with the lack of interpretability [68].

To address the above issues, in this Chapter, we propose LBKT, a novel **L**STM **B**ERT **K**nowledge **T**racing model, for processing long sequence data. The model combines the strength of the Bidirectional Encoder Representations from Transformers (BERT) model in capturing the relations of complex data [41] with the strength of the LSTM model in handling long sequential data to improve its performance on large-scale datasets containing long-sequence data (here, the long-sequence data indicates a length longer than 400 interactions). Moreover, we utilise a Rasch model-based embedding method to process the difficulty level information in the historical behaviour data of students. The Rasch model is a classic yet powerful model in

psychometrics [294], which could be utilised to construct raw questions and knowledge embeddings for KT tasks [68]. Rasch model based embedding could improve the model's performance and interpretability. The experimental results show that our proposed LBKT outperforms the baseline models in five datasets on metrics ACC and AUC. Moreover, it is faster at processing long-sequence data at two long-sequence datasets we extract from the two large-scale datasets. Furthermore, we use t-SNE as the visualisation tool to demonstrate the interpretability of the embedding strategy.

The main contributions of this chapter lie in the following two aspects:

1. We propose LBKT [1], a novel **L**STM **B**ERT **K**nowledge **T**racing model for long sequence data processing. The LBKT leverages the power of BERT, Rasch-based embedding strategies, and LSTM.

2. The experimental results show that LBKT outperforms the baseline models on five ITS datasets on the metric of AUC(assist12, assist17, algebra06, EdNet, and Junyi Academy). Another comparative experiments show the effectiveness of LBKT when processing long-sequence datasets. LBKT model exhibits better interpretability than traditional deep learning based KT models and has advantages in training efficiency.

7.2 Related Work

7.2.1 Knowledge Tracing

Knowledge Tracing (KT) is used in Intelligent Tutoring Systems (ITS) to model and predict a student's mastery level of a specific skill or concept over time [285]. It is based on the assumption that a student's knowledge state is a hidden variable that can be inferred from their observable behaviour, such as their responses to questions or tasks related to the skill or concept being measured [59]. Its goal is to provide personalised feedback and support to students by tracking their progress

[1]Source code and datasets are available at https://github.com/Zhaoxing-Li/LBKT

and adapting instruction to meet their individual needs. This can help to improve student learning outcomes and enhance educational effectiveness. Broadly, there are three categories of KT methods: probabilistic KT models, logistic KT models, and deep learning-based KT models [17].

Probabilistic KT models assume that the student's learning process follows a Markov Process, where students' knowledge mastery could be measured by their observed learning performance [59]. Bayesian KT, or BKT, is the earliest and most classic probabilistic model, which was inspired by cognitive mastery learning [295]. BKT models generally use a probabilistic graphical model, such as the Hidden Markov Model (HMM) [59] and Bayesian Belief Network [296], to track students' changing learning states. The major shortcoming of BKT is that it assumes a simplistic two-state student modelling framework, where a student's knowledge is either learned or unlearned, and there is no concept of forgetting or decay in the model. However, in reality, a student's knowledge could be complex and multi-faceted and could change over time due to various factors such as decay and interference. Therefore, BKT may not be able to capture the nuances of student learning and may not provide an accurate representation of their knowledge state over time. For example, BKT assumes that each question only required one skill and that the various skills were irrelevant to each other [59, 227, 297, 298]. Therefore, in general, BKT models cannot process complicated problems, including the multiple skills and the complex relationship among the concepts, questions, and skills. To address this limitation, Käser *et al.* proposed Dynamic BKT, or DBKT, based on Dynamic Bayesian Network (DBN), to model the prerequisite hierarchies and dependencies of multiple skills [61]. However, both BKT and DBKT still struggle with processing multiple topics or skills, failing to account for contextual factors that may impact student learning.

The logistic KT models are built on the principle of logistic regression, which is a statistical method used to model the probability of a binary outcome based on one or more predictor variables [17]. In the context of educational data, the predictor variables could include a student's prior performance on a set of related skills or concepts, their response time, and their correctness or incorrectness in

answering assessment questions. The output of the logistic regression KT model is a probability estimate of a student's mastery level on a particular skill or concept, which can be used to inform personalised learning interventions and improve student outcomes. There are three logistic models. The Learning Factor Analysis model (LFA) incorporates the initial knowledge state, easiness of knowledge components (KCs), and learning rate of KCs to estimate the student's initial knowledge state, the easiness of different KCs, and the learning rate of KCs [63]. The Performance Factor Analysis (PFA) model is an extension of the LFA model and takes into account the student's performance. PFA considers parameters for previous failures (f) and successes (s) for the KC, in addition to the easiness of KCs [64]. The Knowledge Tracing Machines (KTM) model uses factorization machines (FMs) to extend logistic models to higher dimensions [61].

Inspired by the recent success of deep learning (DL) [31], researchers have applied deep learning technologies into the KT field to develop DL-based Knowledge Tracing [66]. DL-based KT typically models a knowledge tracing task as a sequence prediction problem. With the self-attention architectures applied in the deep learning field, KT models based on the self-attention mechanism began to emerge. For example, SAKT [69] and SAINT+ [236] apply the self-attention mechanism to KT models and achieve higher performance than the traditional DL-based methods. With the development of the self-attention mechanism, Transformer based knowledge tracing models also have been proposed. Ghosh [68] proposes context-aware attentive knowledge tracing (AKT), which introduces a novel monotonic attention mechanism that accounts for the temporal nature of the learning process and the decay of students' knowledge. Nakagawa *et al.* proposed the Graph-based Knowledge Tracing (GKT) model, which incorporates the potential graph structure of KCs into a graph [70]. There were also KT methods based on BERT that had been proposed. MonacoBERT [287] is a BERT-based KT model that incorporates the monotonic convolutional multi-head attention and classical test-theory-based (CTT-based) embedding strategy to improve performance. BEKT [45] is a Bidirectional Encoder representation from the Transformers-based model that predicts student knowledge state by combining historical learning performance.

7.2.2 Transformer-based Model and Application

Transformer is a prominent neural network model proposed by Vaswani *et al.*, which utilises the self-attention mechanism to extract inherent features [40]. Transformer-based models have achieved significant success in the Deep Learning field, especially in Nature Language Processing (NLP) and image generation tasks [231, 282].

The evolution of Transformer-based models, such as BERT [41] and GPT [130], has achieved outstanding performance in the above tasks. BERT, first proposed by Devlin *et al.*, is a successful application of Transformer [41]. BERT utilises the self-attention mechanism and the masked language model (MLM) to train the Transformer bidirectionally in the NLP fields [41]. BERT is renowned for its exceptional ability to process and comprehend natural language text efficiently. It has consistently outperformed other deep learning models in a broad range of tasks, extending beyond the field of NLP. BERT's success can be attributed to several key features, including its bidirectional context, which allows it to capture the dependencies between both preceding and succeeding tokens in a sequence. Additionally, BERT's large pre-training corpus enables it to learn a robust language representation that can be fine-tuned for downstream tasks with relatively small amounts of labelled data. BERT's transformer architecture, which uses self-attention mechanisms to capture global dependencies between tokens in a sequence, is also a significant factor contributing to its performance. The self-attention mechanism allows BERT to weigh the importance of different tokens in a sequence dynamically, which improves its ability to capture complex patterns and relationships in the data [41]. BERT is also known for its ability to generate high-quality embeddings, which are crucial for many natural language processing tasks [41]. There have been a lot of BERT variants applied in other deep learning fields, demonstrating their outstanding performances. For example, ConvBERT [42] applies the original BERT architecture in the image processing field; BERT4Rec uses BERT model to improve recommendation systems [43]; LakhNES uses BERT model to enhance Music Generation [44]. However, in the Knowledge Tracing field, although some BERT-based models, such as BEKT [299] and BiDKT [46], are proposed to improve performance, they are unable to outperform state-of-the-art KT methods in large-scale datasets containing

long-sequence data.

7.3 Methodology

7.3.1 Problem Statement

The key to knowledge tracing is to predict the correctness of a student's next answer in a sequence. Let $x_1, \ldots, x_t$ denote the student's actions, and let the t-th action be represented as $x_t = (q_t, a_t)$, where q_t is the question presented to the student and a_t is the student's response. The goal is to estimate the correctness $P(a_t = 1|x_1, \ldots, x_{t-1})$, that is, the correctness of student's response to the current question, given their previous actions in the sequence.

7.3.2 Proposed Model Architecture

We propose a novel model, LBKT, for the task of knowledge tracing on large-scale datasets containing long-sequence data. While previous BERT-based KT models have shown remarkable success in capturing the relations of complex data, they also have inefficiencies when dealing with long sequence student action data [45]. On the other hand, LSTM models have been proven to excel in handling long sequential data. In response to these challenges, we propose a novel KT model that combines the strengths of both the BERT and LSTM models to improve performance on large-scale datasets containing long-sequence data (where long-sequence data indicates a length longer than 400 interactions). The Rasch embedding (also known as the 1PL IRT model) is a method to represent questions and concepts in a mathematical space [294]. The embeddings are created using a vector that summarizes the variation in questions covering a concept and a scalar difficulty parameter that controls how far a question deviates from the concept it covers. The embeddings are used as raw embeddings for questions and responses, which is a way to track a learner's knowledge state. By leveraging the strengths of a BERT-based model, Rasch model-based embeddings, and long short-term memory (LSTM) unit, our proposed model architecture has the potential to effectively process and understand relationships

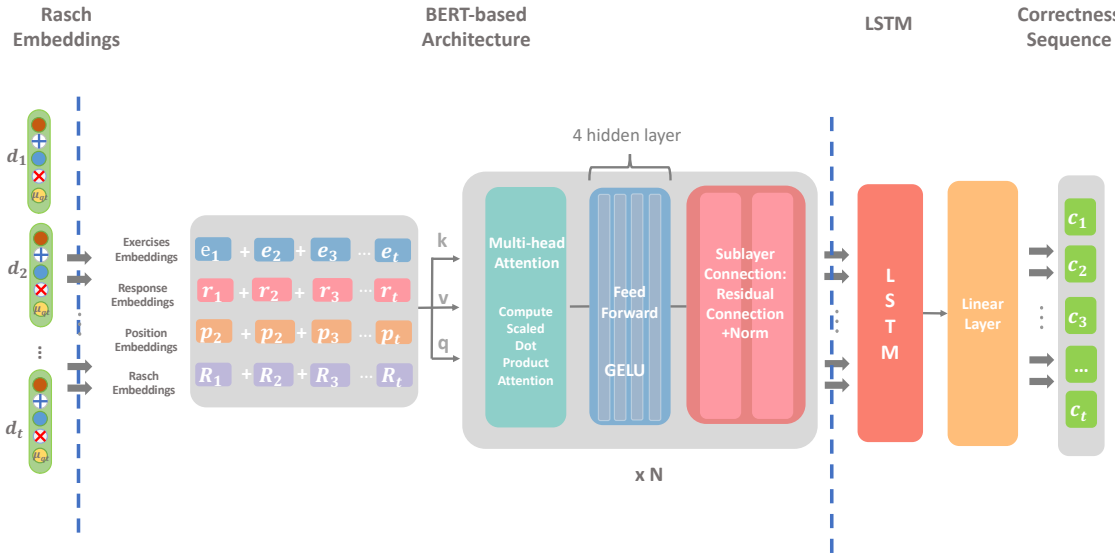


Figure 7.1: The architecture of LBKT. LBKT consists of three components: 1) the Rasch model-based embeddings (on the left), 2) the BERT-based architecture (in the middle), and 3) the LSTM block (on the right).

among different features in long-sequence data, as illustrated in Fig. 7.1.

The first component of LBKT is the Rasch model-based embeddings proposed by Ghosh [68]. The Rasch model-based embeddings consist of difficulty level embeddings E_{d} and question embeddings E_{q}. These embeddings are multiplied and added to the BERT token embeddings and the *sin* and *cos* positional embeddings to build the final embeddings, as shown in the following equation:

$$E = E_{\text{Rasch}} + E_{\text{Bert Token}} + E_{\text{Position}} \tag{7.1}$$

where the Rasch model-based embeddings E_{Rasch} are defined as:

$$E_{\text{Rasch}} = E_{\mathrm{d}} + E_{\mathrm{d}} \times E_{\mathrm{q}} \tag{7.2}$$

The segment embeddings, which are typically used to represent information about the segment in the BERT model, are replaced by the Rasch embeddings mentioned above in our model's architecture. Rasch model-based embeddings are able to more accurately estimate students' knowledge states, as explained earlier,

making them a key contributor to the effectiveness of LBKT for knowledge tracing tasks.

The second component of LBKT is a BERT-based block, which consists of 12 Transformer blocks. Each includes a multi-head attention mechanism, a feedforward network (FFN), and sublayer connections. The multi-head attention mechanism uses the "Scaled Dot Product Attention" method as implemented in BERT, along with queries Q, keys K, values V, and an attention mask for padded tokens. The FFN has a feedforward hidden layer with a size of four times that of the model's hidden layer and uses the GELU activation function rather than RELU.

The sublayer connections in the Transformer block include a residual connection followed by layer normalization. The formulas for the attention mechanism and the FFN are as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7.3}$$

$$\text{FFN}(x) = \text{GELU}(W_1 x + b_1)W_2 + b_2 \tag{7.4}$$

In the third component of LBKT, we use a neural network (NN) linear transformation instead of the attention projection typically used in conjunction with the LSTM unit. This is based on our observed improved performance with the NN linear transformation in our experiments. It should be noted that this choice is not necessarily related to the length or complexity of the sequence but rather to the specific characteristics of the data and the task at hand.

Overall, LBKT is a model that is tailored specifically for use in the field of knowledge tracing. It combines the natural language processing capabilities of the BERT model with the ability to accurately estimate knowledge states using Rasch model-based embeddings and the ability to effectively handle long sequences of data using the LSTM unit and the NN linear transformation. This makes it an ideal choice for the task of knowledge tracing in large-scale datasets containing long-sequence data with unbalanced data distribution.

7.3.3 Experiment Setting

Datasets

We used five benchmark datasets to validate the effectiveness of the LBKT model, including assist12[2], assist17[3], algebra06 [4], EdNet [235] [5], and Junyi Academy [293][6]. Table 7.1 shows the sizes of the above datasets. In general datasets, such as assist 12 and assist 17, it could be challenging to identify and extract large amounts of long-sequence data. Therefore, we validated the speed performance of every model on two datasets with long-sequence student action data extracted from EdNet and Junyi Academy. The mean action sequence length of EdNet is 121.5. The mean interaction length of Junyi Academic is 104.7. Table 7.2 shows the action sequence length statistics of EdNet and Junyi Academy. Here, we define the longer action sequence as longer than 100 records. We extract 200 students' action sequences that include interactions longer than 100 actions from each dataset as the long-sequence dataset to validate the performance of different KT models. Lastly, we selected different lengths of action sequences from Ednet to test the speed performance of each model. We selected four groups with average records lengths of 100, 200, 300, and 400, respectively. Each of these groups included 50 students.

Table 7.1: Benchmark dataset data statistics

Dataset	Students	Concepts	Questions	Interactions
assist12	24,429	264	51,632	1,968,737
assist17	1,708	411	3,162	934,638
algebra06	1,318	1,575	549,821	1,808,533
EdNet	784,309	1,472	11,957	641,712
Junyi Academy	247,606	13,169	722	25,925,922

[2]https://sites.google.com/site/assistmentsdata/home
[3]https://sites.google.com/site/assistmentsdata/home
[4]https://pslcdatashop.web.cmu.edu/KDDCup
[5]https://github.com/riiid/ednet
[6]https://pslcdatashop.web.cmu.edu/Files?datasetId=1275

Table 7.2: Action Length Statistics of EdNet and Junyi Academy

Features	Junyi Academy	EdNet
Students	247606	784309
Max action length	22067	40157
Mean action length	104.7	121.5

Baseline Models

We compared our LBKT to three state-of-the-art models, BEKT [45], AKT [68], DKVMN [300], as well as the two top baseline models in the Riiid Answer Correctness Prediction Competition provided by Kaggle[7], including SSAKT [301], and LTMTI [235].

Evaluation Metrics and Validation

We used the accuracy (ACC) and the area under the curve (AUC) as performance metrics to compare the models' performance in five datasets. We also used the training speed, speed ratio, and memory usage as metrics to compare the performance in the large-scale datasets containing long-sequence data (i.e., EdNet and Junyi Academy). Moreover, we used five-fold cross-validation for the evaluation.

Hyperparameters for Experiments

To compare with each model, the same parameters were used for model training. The batch size was set to 64, and the train/test split was 0.8/0.2. The model used an embedding size of 128 and the Adam optimizer with a learning rate of 0.001. The loss function used was the Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss). The scheduler was set to OneCycleLR with a maximum learning rate of 0.002. Dropout was also being used at a rate of 0.2. The training ran for a total of 100 epochs, with early stopping set to 10 epochs. If the validation loss does not decrease for the first three epochs, the training stops, in order to prevent overfitting and save resources. The maximum sequence length was 200, with an eight-attention head. Hidden sizes were 128 for BERT, 512 for FFN, and 128 for LSTM. The Transformer

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

block/encoder layer was set to 12.

7.4 Results and Discussion

7.4.1 Overall Performance

LBKT outperforms four baseline models on most metrics in the experiments on five benchmark datasets. Tabel 7.3 shows the overall performance of each model. We used five-fold cross-validation to estimate their performances. LBKT performed the best on EdNet and Junyi Academy datasets on both ACC and AUC metrics. It also achieved the best performance on the ACC metric on assist12 and AUC on assist17. On algebra06, AKT achieved the best performance on the ACC metric, BEKT achieved the best performance on the AUC metric, and LBKT achieved the second-best performance on both metrics. This result indicates that LBKT is an efficient KT model on most datasets, especially large-scale datasets containing long-sequence interaction data. This was affected by our LBKT model's unique architecture. The LSTM block enables the model to learn the sequential features of the long sequence and gives more importance to the recent actions of the students, which prevents the model from giving too much weight to the long-ago and low-relevance actions and thus improving the training efficiency.

Table 7.3: Comparison of different KT models on five benchmark datasets. The best performance is denoted in bold.

Dataset	Metrics	LBKT	BEKT	SSAKT	LTMTI	AKT	DKVMN
assist12	ACC	**0.814**	0.786	0.675	0.813	0.769	0.756
	AUC	0.768	**0.813**	0.741	0.785	0.753	0.701
assist17	ACC	0.792	0.795	0.771	0.796	0.733	**0.797**
	AUC	**0.814**	0.801	0.735	0.683	0.803	0.709
algebra06	ACC	0.801	0.797	0.795	0.811	**0.831**	0.800
	AUC	0.799	**0.815**	0.774	0.791	0.814	0.793
EdNet	ACC	**0.803**	0.781	0.761	0.799	0.756	0.800
	AUC	**0.815**	0.795	0.798	0.802	0.798	0.796
Junyi Academy	ACC	**0.832**	0.807	0.777	0.797	0.791	0.790
	AUC	**0.851**	0.831	0.845	0.812	0.799	0.769

Tabel 7.6 shows the performance comparison on the two large-scale datasets. On both datasets, LBKT achieved the best training efficiency. It was 4.29x faster than

BEKT on EdNet and 4.77x faster than BEKT on Junyi Academy. Compared with the second-best model, AKT, LBKT was 1.32x faster on EdNet and 1.42x faster on Junyi Academy. For the memory cost, LBKT was about one-third of BEKT and lower than LTMTL on both datasets. Although the memory cost of LBKT was not the smallest, LBKT has achieved the best results in both ACC and AUC metrics running on the same GPU. This allows LBKT to run on middle-range GPUs. To improve the training efficiency, we used a last input as the query method [20] in the Transformer block instead of the whole sequence, which decreased the complexity of the encoder to improve training speed and reduce memory cost.

Table 7.4: Performance comparison on the two large-scale datasets, EdNet and Junyi Academy. The best performance is denoted in bold.

Model	EdNet			Junyi Academy		
	speed ↑	speed ratio ↑	memory ↓	speed↑	speed ratio ↑	memory ↓
BEKT	4.93	1.00x	16.7 GB	4.85	1.00x	16.6 GB
SSAKT	7.13	1.44x	**3.4 GB**	6.22	1.28x	**3.2 GB**
LTMTI	13.8	1.32x	7.69 GB	12.1	1.19x	8.82 GB
AKT	17.1	3.25x	4.32 GB	16.4	3.35x	4.37 GB
DKNMN	5.97	2.34x	7.68 GB	4.67	3.75x	8.53 GB
LBKT	**21.3**	**4.29x**	6.09 GB	**22.2**	**4.77x**	6.08 GB

To estimate the model performance on different lengths of data sequences, we sorted the data in the two datasets by length and divided them into four sub-datasets according to the average length. The average lengths of the four sub-datasets were set as 100, 200, 300, and 400, respectively. Sequences shorter than the mean were padded with 0s, and sequences longer than the mean were pruned.

Fig. 7.2 shows the results of the speed performance comparison of each model processing different lengths of data sequences. LBKT has a relatively high-speed performance compared to other KT models when processing data sequences with varying lengths. LBKT is the fastest model in all four groups of data lengths. AKT and DKMN also have relatively high speeds, with AKT being the second-fastest model in all groups and DKMN being the third-fastest model. Overall, the results suggest that LBKT is the fastest model, and that it is particularly efficient at dealing with long sequences of data. The fact that LBKT maintains its high speed even when processing longer sequences of data indicates that it is well-suited

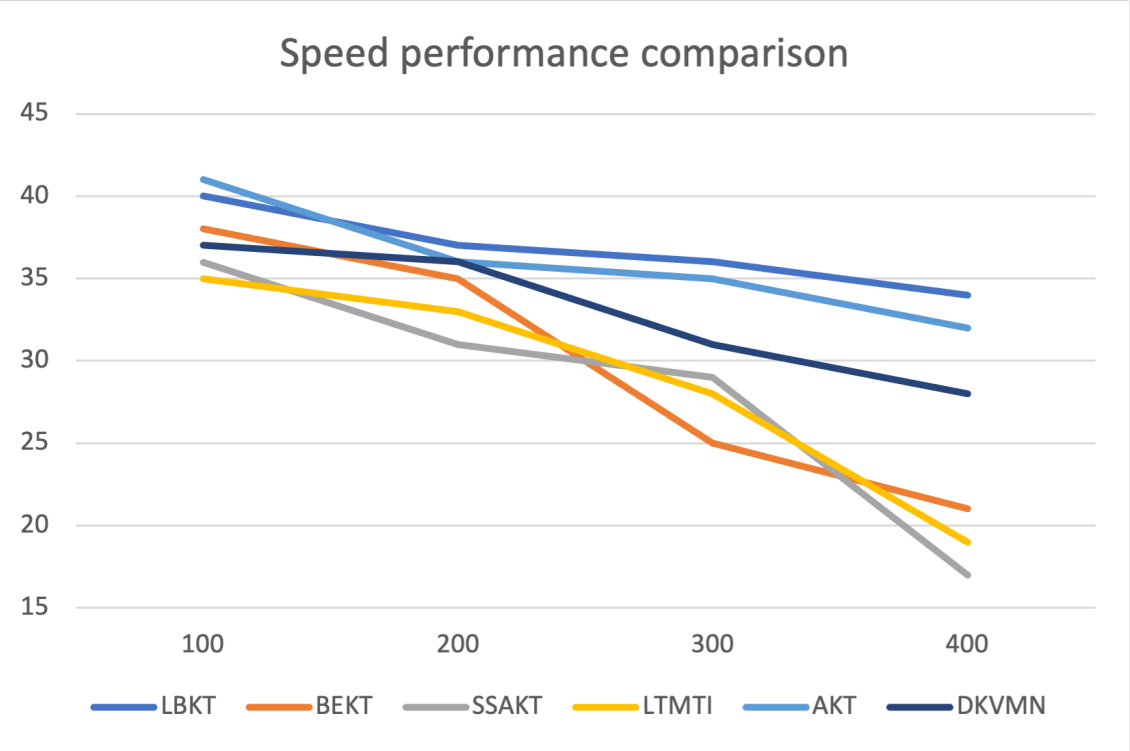


Figure 7.2: Speed performance comparison of each model when processing data sequences with varying lengths. The vertical axis is the speed (10^4 samples per sec).

for tasks that require the analysis of large amounts of data over extended periods of time.

7.4.2 Ablation Study

In this section, we explore why LBKT performed better than other methods and which components affected the overall performance. Table 7.5 shows the results of the ablation study. We compared LBKT, LBKT without Rasch model-based embeddings block (denoted as LBKT-Rasch), LBKT without LSTM block (denoted as LBKT-LSTM), and LBKT without both Rasch model-based embeddings LSTM (denoted as BERT). The results show that LBKT achieved the best performance on EdNet and Junyi Academy on both ACC and AUC metrics. It also achieved the best performance on one metric in every dataset. BERT-only achieved the best performance on assist17 on ACC, which shows that the combination with Rasch embeddings and LSTM could improve the performance of a single BERT model.

Table 7.5: Results of the ablation study. LBKT-Rasch denotes LBKT without Rasch embedding; LBKT-LTSM denotes LBKT without LSTM block; and BERT denotes only the transformer structure-based blocks are included. The best performance is denoted in bold.

Dataset	Metrics	LBKT	LBKT-Rasch	LBKT-LSTM	BERT
assist12	ACC	**0.804**	0.785	0.799	0.793
	AUC	0.768	0.768	**0.783**	0.750
assist17	ACC	0.784	**0.792**	0.782	**0.792**
	AUC	**0.814**	0.709	0.779	0.799
algebra06	ACC	**0.801**	0.796	0.792	0.798
	AUC	0.799	0.756	**0.809**	0.765
EdNet	ACC	**0.803**	0.729	0.722	0.801
	AUC	**0.815**	0.758	0.794	0.809
Junyi Academy	ACC	**0.882**	0.856	0.874	0.879
	AUC	**0.907**	0.893	0.877	0.901

Furthermore, we conducted additional paired t-tests to verify the significance of the observed differences in performance metrics. These tests confirm that the improvements, while numerically small, are statistically significant ($p < 0.05$). In the context of educational data mining, even small improvements in predictive accuracy can have meaningful implications for student support systems.

Table 7.6 shows the ablation study of speed performance comparison on the two large-scale datasets. The results show that LBKT has the highest speed performance among all models on both datasets, with a speed of 21.3 samples per second on EdNet and 22.2 samples per second on Junyi Academy. This suggests that LBKT is a highly efficient model for processing large amounts of data in real-time. Interestingly, LBKT-LSTM, which removes the LSTM layer from the proposed model, has a significantly lower speed performance compared to LBKT, with a speed ratio of only 1.29x on EdNet and 1.19x on Junyi Academy. This suggests that the LSTM layer is an important component in the proposed model and contributes significantly to its speed performance. This is likely due to the ability of LSTM to capture long-term dependencies and sequential patterns in the data, which can be crucial in educational applications.

Table 7.6: Speed performance comparison ablation study on the two large-scale datasets, EdNet and Junyi Academy. The best performance is denoted in bold.

Model	EdNet			Junyi Academy		
	speed	speed ratio	memory	speed	speed ratio	memory
LBKT	**21.3**	**4.29x**	6.09GB	**22.2**	**4.77x**	6.08GB
LBKT-Rash	19.1	3.51x	5.6GB	20.31	3.67x	**3.1 GB**
LBKT-LSTM	13.8	1.29x	7.87GB	12.1	1.09x	8.37GB
BERT	12.1	3.27x	**4.32GB**	11.4	3.52x	4.34GB

7.4.3 Analysis of Embedding Strategy

In this section, We used t-SNE as the visualisation tool to show the interpretability of LBKT's embedding strategy. Fig. 7.3 shows the results of No-Rasch-embedding, and Fig.7.4 shows the Rasch embedding strategy. We can see that, in the No-Rasch-embedding scenario, the difficult questions' embeddings (dark blue vectors) mixed with the easy questions' embeddings (yellow to light blue vectors). In figure 7.4, the difficult level embeddings were separated to avoid mixing with easy level embeddings.

Questions at a higher difficulty level are typically associated with longer sequence data, as students spend more time and steps on difficult exercises, which results in longer interaction sequences. Rasch model-based embeddings could divide different difficulty-level parts before the start of the model training and not mix them with other difficulty-level embeddings. As a result, it might increase training efficiency to converge faster.

7.4.4 Comparison with MLFBK

Chapter 6 introduced the MLFBK Model, a novel BERT-based Knowledge Tracing (KT) model that accentuates the utilisation of multiple features and the exploration of latent relations amongst them. This model is particularly focused on harnessing a diverse set of features, such as individual skill mastery and problem difficulty, to augment the accuracy and efficiency of knowledge tracing.

When it comes to selecting between these models, several factors need consideration. For scenarios involving long-sequence student interaction data, particularly in large-scale datasets, the LBKT model presented in Chapter 7 is the preferred

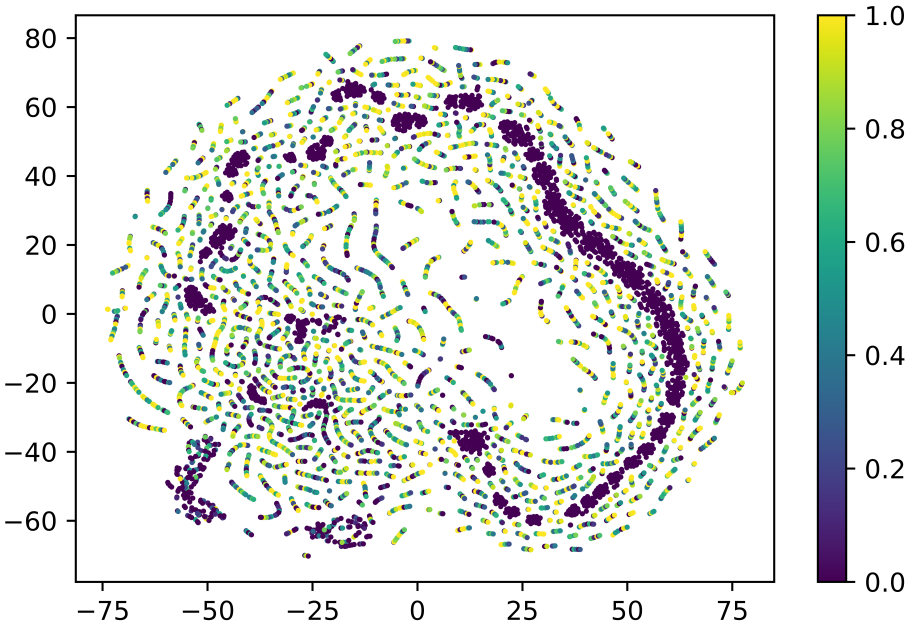


Figure 7.3: Visualisation of the embedding vector using t-SNE: *without* Rasch embeddings. The colour bar is the predicted probability of the outputs.

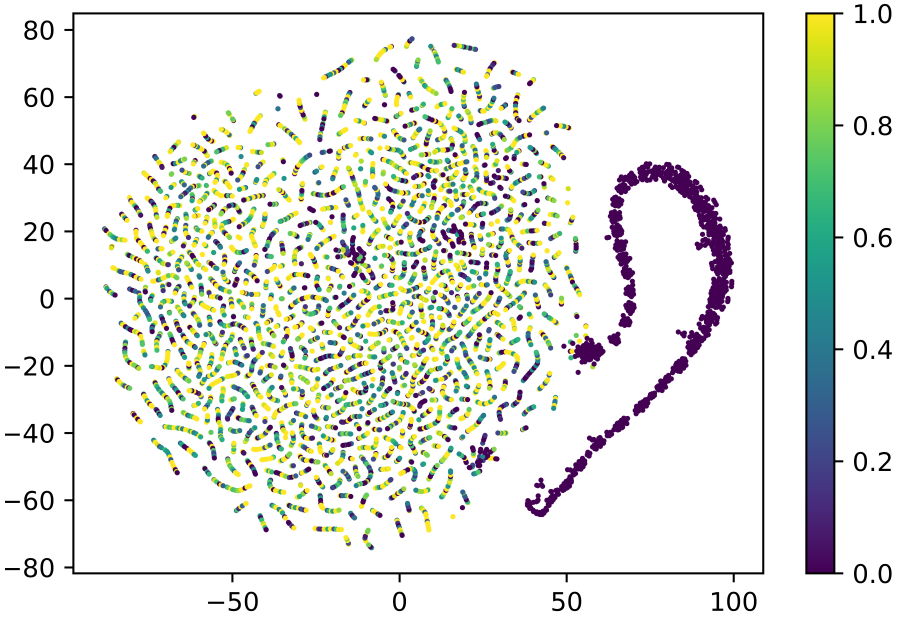


Figure 7.4: Visualisation of the embedding vector using t-SNE: *with* Rasch embeddings. The colour bar is the predicted probability of the outputs.

choice. In contrast, the MLFBK model from Chapter 6 is more suitable for situations where a rich feature set and the exploration of latent relations are imperative for a comprehensive understanding of student learning patterns.

Regarding computational resources, it is noteworthy that both models, owing to their deep learning nature, are resource-intensive. The choice between them may, therefore, hinge on the available computational capabilities and the size of the dataset being handled.

In different educational contexts, the models have their distinct advantages. The MLFBK model is preferable in contexts where a nuanced understanding of student learning, facilitated by multiple features and latent relationships, is essential. On the other hand, the LBKT model is more apt for environments characterised by extensive sequential interaction data that require efficient processing.

Integrating the MLFBK and LBKT models offers a promising route to enhance accuracy and efficiency in large-scale educational settings, combining in-depth feature analysis with extensive sequence management. However, this endeavour is not without substantial challenges. Merging the two models significantly increases complexity, demanding more resources and advanced expertise and potentially exacerbating issues such as overfitting and data processing complexities.

7.5 Summary

In this Chapter, we have developed LBKT, which employs a BERT-based architecture with an LSTM block for processing long-sequence data, and Rasch model-based embeddings for different difficulty levels of questions. Experiments show that LBKT outperforms baseline models on most benchmark datasets. We also conducted the speed performance experiment on the two large-scale datasets containing long-sequence data. The results suggest that LBKT could process long-sequence data faster and is more resource-efficient. Moreover, we conducted an ablation study for different components of LBKT. The results indicate that the LSTM component aided in improving the performance of dealing with long-sequence data. Furthermore, we conducted an analysis of the embedding strategy using t-SNE. The result

shows that Rasch embedding could process the difficulty-level features effectively.

Epilogue

This chapter addressed research question 5: "How to effectively deal with large-scale datasets and process long sequence data to improve the performance of Knowledge Tracing (KT)?" This research question was approached through the following research objectives:

RO 5.1 involved proposing a novel LSTM BERT-based Knowledge Tracing model, LBKT, specifically designed to handle long sequence data in ITS. By leveraging the power of LSTM and BERT, LBKT aimed to capture the temporal dependencies and semantic representations in the data, enabling more accurate and reliable knowledge tracing.

RO5.2 focused on evaluating the performance and scalability of the proposed LBKT model in processing large-scale datasets and long-sequence data. Benchmark datasets were utilized to compare the performance of LBKT with existing methods.

RO 5.3 aimed to conduct an ablation study to analyze the impact of each component of LBKT on its overall performance. By systematically evaluating and removing specific components of LBKT, the study sought to identify the contributions of individual elements and highlight their importance in enhancing the model's effectiveness. Furthermore, the t-SNE visualization tool was utilized to demonstrate the interpretability of the embedding strategy employed in LBKT.

CHAPTER 8

Conclusions and Future Work

8.1 Summary and Contributions

This thesis aims to explore and enhance the collaboration between humans and Artificial Intelligence (AI) in the context of Intelligent Tutoring Systems (ITS) by addressing key challenges and proposing innovative approaches. The main contributions and findings are summarised as follows:

We have presented a comprehensive survey on Collaborative Systems to enhance research into human-AI interactions and collaborative designs in Chapter 3. The study has led us to propose a novel classification method for Collaborative Reinforcement Learning (CRL), called the "CRL Design Trajectory Map". Additionally, we have introduced a new CRL taxonomy as a systematic modelling tool to aid in the selection and enhancement of new collaborative systems designs. Researchers can utilise our Trajectory Map to develop a Human-AI collaborative system from scratch or adapt specific aspects of it to refine their existing systems. For instance, they can select the desired system structure from the Human-AI Collaborative Design Pattern, identify and address the requirements of various components in Collaborative Levels and Parties and Collaborative Capabilities, and choose different design

components in Algorithmic Models and Interactive Methods. This comprehensive design approach encompasses both top-to-bottom and macro-to-micro perspectives.

This framework offered valuable insights into the collaborative process between students and Intelligent Tutoring Systems (ITS), thereby facilitating effective and personalised ITS design and development. Drawing upon the theories discussed in Chapter 3, we modelled the Student-ITS collaborative process as a bidirectional interaction. In the Student-to-ITS process, we proposed two student modelling methods: SimStu and SimGAIL. These methods were designed to optimise the performance of ITS while minimising the input and interaction necessary from students.

The SimStu model was introduced in Chapter 4, which utilised a Transformer-based approach to generate simulated student behavioural data. SimStu addressed the data scarcity problem in ITS training by simulating interactions between sim students and ITS. The experimental results showed that SimStu successfully modelled real student behaviour and improved the efficiency of ITS training. The Sim-GAIL presented in Chapter 5 model leverages Generative Adversarial Imitation Learning (GAIL) to generate high-fidelity and diverse simulated student behavioural data. Sim-GAIL addressed the lacking training data problem in ITS by providing simulated data for training ITS models when labelled interaction data is limited. The experimental results demonstrated the effectiveness of Sim-GAIL in improving the prediction accuracy of knowledge tracing models in a cold-start scenario.

Comparing SimStu and SimGAIL, SimStu is more suitable for continuous and dense data conditions. Sim-GAIL, on the other hand, generates more accurate and realistic simulations of student behaviour, even with sparse rewards, by better-capturing reward distributions and strategies from real data. Thus, Sim-GAIL is more suitable for situations requiring precise simulations of student behaviour, especially where reward signals are scarce.

In the ITS-to-Student process, we proposed two knowledge tracing methods: MLFBK and LBKT. MLFBK focuses on capturing complex interactions and individual differences by incorporating multiple features and latent relations. LBKT, on the other hand, addresses the efficient processing of long sequence data while maintaining interpretability. Each method tackles different challenges in Knowledge

Tracing. These methods were designed to enhance the accuracy and efficiency of the ITS. Their purpose was to enable ITS to operate autonomously and deliver personalised support, empowering students to learn at their own pace with minimal effort. MLFBKT model was presented in Chapter 6, which integrated multiple features and mined latent relations in student interaction data to enhance the accuracy and efficiency of Knowledge Tracing (KT) models. The experiments demonstrated that MLFBKT outperformed baseline models and improved the performance of KT in estimating students' knowledge states and predicting their learning behaviours. Chapter 7 presented the LBKT model, a novel LSTM BERT-based Knowledge Tracing model. LBKT effectively processed long sequence data and improved the performance of KT models by combining the strengths of the BERT model and the LSTM model. The experimental results showcased the superiority of LBKT in terms of accuracy and scalability.

Comparing MLFBK and LBKT, LBKT is favoured for long-sequence, large-scale datasets, while MLFBK is more appropriate for contexts that demand a rich set of features and the exploration of latent relationships. Both models are resource-intensive due to their deep learning nature, and available computational resources and the size of the dataset may also influence the decision. In educational environments, MLFBK is fitting where a nuanced understanding of student learning is required, while LBKT is optimal for efficiently processing extensive sequential interaction data.

To sum up, the contributions of this thesis have advanced the field of Human-AI collaboration in educational settings, enhanced the capabilities of Intelligent Tutoring Systems, and improved the accuracy and interpretability of Knowledge Tracing models. These findings provide valuable insights and methodologies for designing and developing collaborative systems between humans and AI in the context of education.

8.2 Answers to Research Questions

In this thesis, I introduced a trajectory map as a foundational framework for de-

signing human-AI collaboration systems (Chapter 3). This map allows researchers to select collaboration models ranging from macro to micro perspectives. For this thesis's specific case, I employ a two-directional collaboration pattern to guide the development of a student-ITS collaboration system. I proposed two student behaviour simulation methods, Sim-Stu (Chapter 4) and Sim-GAIL (Chapter 5), to enhance ITS training efficiency by minimising student actions. Additionally, we develop two knowledge tracking models, MLFBK (Chapter 6) and LBKT (Chapter 7),to improve ITS accuracy in predicting student behaviour and providing better recommendations to students. By implementing these four methods, our approach enhances the overall collaboration efficiency between students and ITS.

Concretely, this thesis presents the first research theme in exploring the design trajectories map of Human-AI Collaborative Systems, with a comprehensive survey in Chapter 3 to answer **RQ1**:"What trajectory could we follow to develop a user-friendly Student-ITS collaboration system?"

- A comprehensive Human-AI Collaborative Design Trajectory Map was developed, serving as a systematic modelling tool for selecting collaboration theories and collaborative frameworks *RO 1.1, RO 1.2*.

- Conceptualised a framework by considering design patterns, collaborative levels, parties and capabilities and reviewing interactive methods and algorithmic models to address *RO 1.3*.

- Thorough examination of the generic challenges associated with Human-AI Collaborative Systems addresses *RO 1.4*.

We addressed the second research theme in proposing efficient approaches to solve the cold-start problems and scarcity of data for ITS training, with two student modelling methods proposed in Chapter 4 and 5 to answer **RQ2** and **RQ3**:

RQ2: "How to generate high-fidelity and diverse simulated student behavioural data for training Intelligent Tutoring Systems (ITS) using deep learning methods?"

Chapter 4 addressed this research question by proposing SimStu, a student behaviour data simulation method based on an emerging deep learning method, the Transformer.

- We have presented SimStu, a Transformer-based model to generate simulated student behavioural data. The aim was to develop a model capable of producing realistic and diverse student behaviours *RO 2.1*.

- We have evaluated the performance of the proposed model and compared it with the real student behaviour data *RO 2.2*.

- We have explored the application of the generated data from the proposed model to Knowledge Tracing, which aimed to assess the performance of the proposed model in improving the accuracy of Knowledge Tracing *RO 2.3*.

RQ3: "How can we generate realistic and diverse simulated student behavioural data for training Intelligent Tutoring Systems (ITS) through reinforcement learning techniques?"

Chapter 5 addressed this research question by proposing the SimGAIL method, a student behaviour data simulation method based on an emerging reinforcement learning method, Generative Adversarial Imitation Learning (GAIL) approach.

- We have proposed SimGAIL, a student modelling method based on the Generative Adversarial Imitation Learning (GAIL) approach *RO 3.1*.

- We have compared the performance of the proposed Sim-GAIL method with two traditional methods based on Reinforcement Learning and Imitation Learning *RO 3.2*.

- We have evaluated the effectiveness of the proposed Sim-GAIL method in improving the prediction accuracy of the Knowledge Tracing (KT) model, particularly in a cold-start scenario *RO 3.3*.

Moreover, This Thesis addresses research theme 3 in improving the performance of Knowledge Tracing models, with two novel Knowledge tracing methods proposed in Chapter 6 and Chapter7 to answer Research Questions 4 and 5:

RQ4: How can multiple features and latent relations in student interaction data be integrated to improve the accuracy and efficiency of Knowledge Tracing (KT) models for Intelligent Tutoring Systems (ITS)?

Chapter 6 addressed this research question by proposing the MLFBKT model, which integrated multiple features and mined latent relations in student interaction data to enhance the accuracy and efficiency of KT models. The experimental results demonstrated the improved performance of MLFBKT in estimating students' knowledge states and predicting their learning behaviours.

- We have developed MLFBKT, a novel method to integrate multiple features and latent relations into KT models *RO 4.1* and *RO 4.2*.
- Benchmark datasets were utilised to compare the performance of MLFBKT with existing state-of-the-art methods *RO 4.3*.
- An ablation study was conducted to analyse the impact of each latent relation of MLFBKT on its overall performance. The t-SNE visualisation tool was also used to showcase the interpretability of the embedding strategy *RO 4.3*.

RQ5: How to effectively deal with large-scale datasets, process long-sequence data, and improve the performance of KT models for ITS?

Chapter 7 addressed this research question by proposing the LBKT model, an LSTM BERT-based Knowledge Tracing model designed to process long sequence data. The experiments confirmed the effectiveness of LBKT in handling large-scale datasets, processing long sequences, and improving the performance of KT models.

- We have presented a novel LSTM BERT-based Knowledge Tracing model, LBKT, specifically designed to handle long sequence data in ITS. By leveraging the power of LSTM and BERT, LBKT aimed to capture the temporal dependencies and semantic representations in the data, enabling more accurate and reliable knowledge tracing *RO 5.1*.
- Benchmark datasets were utilised to compare the performance of LBKT with existing state-of-the-art methods *RO 5.2*.
- An ablation study was conducted to analyse the impact of each component of LBKT on its overall performance. Furthermore, the t-SNE visualisation tool was utilised to demonstrate the interpretability of the embedding strategy employed in LBKT *RO 5.3*.

8.3 Limitations

While this thesis has made contributions to the field of Human-AI collaboration, Intelligent Tutoring Systems, and Knowledge Tracing, there are some research limitations due to time, cost, research equipment, and other unforeseen reasons. We will present these points from three perspectives:

From a Human-AI collaboration system perspective. We have presented a comprehensive Human-AI Collaborative Design Trajectory Map. However, there are still some limitations. One limitation lies in its focus on reinforcement learning in the context of human-AI collaboration. Chapter 3 primarily concentrates on reinforcement learning due to the limited availability of studies investigating the collaboration between humans and deep learning algorithms. However, with the development of large language models such as ChatGPT or other pre-trained language models (LLMs), the collaboration between Human and Deep Learning methods draws increasing attention, and in the fine-tuning stage of ChatGPT, human trainers review and rate different model-generated responses, providing feedback on their quality and appropriateness. This iterative process helps refine the model's behaviour and ensures that it generates more contextually relevant and coherent responses. Although our survey did not encompass methods specifically focusing on human interaction with deep learning, our work has covered this collaborative model in human-AI collaboration patterns. However, there is still room for further exploration and investigation of methods specifically tailored to collaboration with deep learning algorithms. Moreover, the thesis primarily focused on the technical aspects of Human-AI collaboration in the context of ITS. Future research could consider the human factors, social aspects, and ethical considerations of deploying such collaborative systems in educational settings.

From the student modelling perspective, the simulated student behavioural data generated by Sim-GAIL and SimStu were evaluated based on their impact on ITS training. However, the limitation is that we relied solely on published datasets, which may not encompass all the relevant and valuable student action features. As a result, the student data generated through simulation may contain inherent biases. Furthermore, due to the unavailability of an Intelligent Tutoring System (ITS) with

an open API that is widely used, we were unable to measure the real impact of our methods on ITS during the validation process. These constraints hindered our ability to fully explore the practical implications and real-world effectiveness of our approach.

From the knowledge tracing perspective, the MLFBKT and LBKT models demonstrated improved performance in predicting accuracy. One limitation of our study is that we focused on improving the accuracy of the Knowledge Tracing (KT) model by combining multiple features and latent relations in the MLFBKT approach. However, we did not explore the integration of MLFBKT with LBKT, which could potentially enhance the accuracy of the KT model when processing long sequences. Future research should aim to develop general high-efficiency models that combine both MLFBKT and LBKT approaches. Another limitation is the lack of an Intelligent Tutoring System (ITS) model with an open API that could be used on a large scale. As a result, our KT model could not be directly applied to students' actual learning experiences, nor could it measure the real impact on student performance. In future work, it would be valuable to apply our model to the actual teaching process, allowing for a more comprehensive evaluation of its effectiveness and impact on student learning outcomes.

8.4 Lessons Learned and Future Directions

Throughout the research journey, several lessons have been learned, and new directions for future research have emerged:

Integration of human factors and user-centric design: While the proposed models and frameworks provide valuable insights into Human-AI collaboration, it is crucial to consider the perspectives and needs of end-users, such as students and teachers, during the design and development process. The current student modelling and knowledge tracing methods are hard to apply in real-world ITS. It is challenging to collect users' advice. Therefore, future research should prioritize user-centric design methodologies and involve stakeholders from educational institutions to ensure the practicality and acceptance of collaborative systems.

Interdisciplinary collaboration: Human-AI collaboration in education requires collaboration between researchers and practitioners from various disciplines, including education, computer science, human-computer interaction, and psychology. Future research could foster interdisciplinary collaboration to leverage diverse perspectives and expertise to address the complex challenges of designing effective, user-friendly collaborative systems.

Ethical considerations and responsible AI: As AI technologies continue to evolve and impact educational settings, it is essential to address ethical considerations and ensure responsible AI practices. Future research should explore the ethical implications of Human-AI collaboration in education, including issues related to privacy, data security, algorithmic bias, and fairness. Additionally, guidelines and frameworks should be developed to ensure the ethical deployment and use of Intelligent Tutoring Systems.

Long-term evaluation and impact assessment: To assess the long-term effectiveness and impact of collaborative systems in education, future research should conduct longitudinal studies and evaluate students' learning outcomes and educational experiences over an extended period. This will provide insights into the sustained benefits and challenges of Human-AI collaboration in educational settings.

Real-time feedback and interaction: The knowledge tracing method proposed in this thesis builds on the students' historical dataset. It is hard to model the real-time student knowledge mastery state. Therefore, exploring real-time feedback and interaction between students and Intelligent Tutoring Systems is a promising direction for future research. Collaborative systems can foster deeper engagement and more effective learning experiences by enabling dynamic and interactive exchanges.

Large-scale deployment and practical implementation: Future research should address the challenges of large-scale deployment and practical implementation of collaborative systems in real educational settings.

By addressing these lessons learned and future directions, we could further advance the field of Human-AI collaboration in education and contribute to developing innovative and effective educational technologies.

In conclusion, this thesis has contributed to the field of Human-AI collabora-

tion in the context of the student and ITS collaborative process. This research has advanced the effectiveness and performance of collaborative systems between Students and ITS by addressing key challenges and proposing innovative student modelling and knowledge tracing approaches. The proposed frameworks, models, and methodologies have demonstrated improved performance and effectiveness in enhancing students' learning experiences and outcomes. However, further research is needed to address the limitations, consider human factors, ensure ethical practices, and evaluate the long-term impact of collaborative systems. The lessons learned and future directions outlined in this thesis provide a foundation for future research and development in the field of Human-AI collaboration in education.

Bibliography

[1] Z.-H. Zhou, *Machine learning*. Springer Nature, 2021. 1.1.1

[2] S. Y. Park, P.-Y. Kuo, A. Barbarin, E. Kaziunas, A. Chow, K. Singh, L. Wilcox, and W. S. Lasecki, "Identifying challenges and opportunities in human-ai collaboration in healthcare," in *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing*, pp. 506–510, 2019. 1.1.1

[3] D. Wang, E. Churchill, P. Maes, X. Fan, B. Shneiderman, Y. Shi, and Q. Wang, "From human-human collaboration to human-ai collaboration: Designing ai systems that can work together with people," in *Extended abstracts of the 2020 CHI conference on human factors in computing systems*, pp. 1–6, 2020. 1.1.1, 2.1

[4] A. Kirkwood and L. Price, "Technology-enhanced learning and teaching in higher education: what is 'enhanced'and how do we know? a critical literature review," *Learning, media and technology*, vol. 39, no. 1, pp. 6–36, 2014. 1.1.1

[5] L. Daniela, D. Kalniņa, and R. Strods, "An overview on effectiveness of technology enhanced learning (tel)," *International Journal of Knowledge Society Research (IJKSR)*, vol. 8, no. 1, pp. 79–91, 2017. 1.1.1

[6] E. Mousavinasab, N. Zarifsanaiey, S. R. Niakan Kalhori, M. Rakhshan, L. Keikha, and M. Ghazi Saeedi, "Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods," *Interactive Learning Environments*, vol. 29, no. 1, pp. 142–163, 2021. 1.1.1

[7] A. Almasri, A. Ahmed, N. Almasri, Y. S. Abu Sultan, A. Y. Mahmoud, I. S. Zaqout, A. N. Akkila, and S. S. Abu-Naser, "Intelligent tutoring systems survey for the period 2000-2018," 2019. 1.1.1, 4

[8] J. Lin, Z. Ma, R. Gomez, K. Nakamura, B. He, and G. Li, "A review on interactive reinforcement learning from human social feedback," *IEEE Access*, vol. 8, pp. 120757–120765, 2020. 1.1.1

[9] C. Arzate Cruz and T. Igarashi, "A survey on interactive reinforcement learning: Design principles and open challenges," in *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pp. 1195–1209, 2020. 1.1.1, 2.1, 3.1, 3.2, 3.2, 3.1, 3.3.2, 3.3.3, 3.7, 3.8.1, 3.8.3, 3.8.4, 3.9

[10] J. Bassen, B. Balaji, M. Schaarschmidt, C. Thille, J. Painter, D. Zimmaro, A. Games, E. Fast, and J. C. Mitchell, "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, 2020. 1.1.1, 5.1, 5.3.2, 5.3.2

[11] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 253–260, 2002. 1.1.1

[12] F. Pedro, M. Subosa, A. Rivas, and P. Valverde, "Artificial intelligence in education: Challenges and opportunities for sustainable development," 2019. 1.1.1

[13] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15084–15097, 2021. 1.1.1, 3.2, 4.1, 4.2.3, 4.3.1

[14] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016. 1.1.1, 2.3.4, 5.1, 5.2.2, 5.2.3, 5.3.3, 5.6

[15] G. Abdelrahman, Q. Wang, and B. P. Nunes, "Knowledge tracing: A survey," *ACM Computing Surveys*, 2022. 1.1.1, 6.1, 7.1

[16] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, and G. Hu, "Ekt: Exercise-aware knowledge tracing for student performance prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 1, pp. 100–115, 2019. 1.1.1, 2.4.3, 6.2.3

[17] Q. Liu, S. Shen, Z. Huang, E. Chen, and Y. Zheng, "A survey of knowledge tracing," *arXiv preprint arXiv:2105.15106*, 2021. 1.1.1, 2.4, 5.2.4, 5.5.4, 6.1, 6.2.2, 7.1, 7.2.1

[18] L. He, J. Tang, X. Li, P. Wang, F. Chen, and T. Wang, "Multi-type factors representation learning for deep learning-based knowledge tracing," *World Wide Web*, vol. 25, no. 3, pp. 1343–1372, 2022. 1.1.1, 6.1

[19] C. Zhang, Y. Jiang, W. Zhang, and C. Gu, "Muse: Multi-scale temporal features evolution for knowledge tracing," *arXiv preprint arXiv:2102.00228*, 2021. 1.1.1, 6.1

[20] S. Jeon, "Last query transformer rnn for knowledge tracing," *arXiv preprint arXiv:2102.05038*, 2021. 1.1.1, 7.1, 7.4.1

[21] Y. Liu, J. Zhou, and W. Lin, "Efficient attentive knowledge tracing for long-tail distributed records," in *2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*, pp. 104–109, IEEE, 2021. 1.1.1, 7.1

[22] Y. Wang and Z. Pardos, "Does chronology matter? sequential vs contextual approaches to knowledge tracing," 2022. 1.1.1, 7.1

[23] K. R. Skene, *Artificial Intelligence and the Environmental Crisis: Can Technology Really Save the World?* Routledge, 2019. 2.1, 3.1

[24] E. Yudkowsky *et al.*, "Artificial intelligence as a positive and negative factor in global risk," *Global catastrophic risks*, vol. 1, no. 303, p. 184, 2008. 2.1, 3.1

[25] E. Hollnagel and D. D. Woods, "Cognitive systems engineering: New wine in new bottles," *International journal of man-machine studies*, vol. 18, no. 6, pp. 583–600, 1983. 2.1, 3.1, 3.1, 3.4.1, 3.9

[26] K. Schmidt, J. Rasmussen, B. Brehmer, and J. Leplat, "Cooperative work: A conceptual framework," *Distributed decision making: Cognitive models for cooperative work*, pp. 75–110, 1991. 2.1, 3.1, 3.1, 3.3.2, 3.3.3, 3.4, 3.5.1, 3.5.2, 3.5.2, 3.5.3

[27] M. Johnson, C. Jonker, B. Van Riemsdijk, P. J. Feltovich, and J. M. Bradshaw, "Joint activity testbed: Blocks world for teams (bw4t)," in *International Workshop on Engineering Societies in the Agents World*, pp. 254–256, Springer, 2009. 2.1, 3.1, 3.5.2

[28] A. Najar and M. Chetouani, "Reinforcement learning with human advice. a survey," *arXiv preprint arXiv:2005.11016*, 2020. 2.1, 3.1, 3.3.2

[29] G. Li, R. Gomez, K. Nakamura, and B. He, "Human-centered reinforcement learning: a survey," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 4, pp. 337–349, 2019. 2.1, 3.1

[30] E. Puiutta and E. Veith, "Explainable reinforcement learning: A survey," *arXiv preprint arXiv:2005.06247*, 2020. 2.1, 3.1, 3.5.1

[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015. 2.2, 7.2.1

[32] K. Chowdhary and K. Chowdhary, "Natural language processing," *Fundamentals of artificial intelligence*, pp. 603–649, 2020. 2.2

[33] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*. prentice hall professional technical reference, 2002. 2.2

[34] D. R. Reddy, "Speech recognition by machine: A review," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 501–531, 1976. 2.2

[35] D. Mandic and J. Chambers, *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. Wiley, 2001. 2.2.1

[36] S. Egan, W. Fedorko, A. Lister, J. Pearkes, and C. Gay, "Long short-term memory (lstm) networks with jet constituents for boosted top tagging at the lhc," *arXiv preprint arXiv:1711.09059*, 2017. 2.2.1, 2.2.2

[37] S. Jastrzebski, D. Arpit, N. Ballas, V. Verma, T. Che, and Y. Bengio, "Residual connections encourage iterative inference," *arXiv preprint arXiv:1710.04773*, 2017. 2.2.1

[38] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, pp. 5929–5955, 2020. 2.2.2

[39] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018. 2.2.3

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. 2.2.4, 4.2.3, 4.2.4, 4.3.1, 6.2.1, 7.2.2

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. 2.2.5, 6.2.1, 7.1, 7.2.2

[42] Z.-H. Jiang, W. Yu, D. Zhou, Y. Chen, J. Feng, and S. Yan, "Convbert: Improving bert with span-based dynamic convolution," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12837–12848, 2020. 2.2.5, 6.2.1, 7.2.2

[43] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1441–1450, 2019. 2.2.5, 6.2.1, 7.2.2

[44] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. McAuley, "Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training," *arXiv preprint arXiv:1907.04868*, 2019. 2.2.5, 6.2.1, 7.2.2

[45] Z. Tiana, G. Zhengc, B. Flanaganb, J. Mic, and H. Ogatab, "Bekt: Deep knowledge tracing with bidirectional encoder representations from transformers," in *Proceedings of the 29th International Conference on Computers in Education*, 2021. 2.2.5, 6.2.2, 6.3.3, 7.2.1, 7.3.2, 7.3.3

[46] W. Tan, Y. Jin, M. Liu, and H. Zhang, "Bidkt: Deep knowledge tracing with bert," in *International Conference on Ad Hoc Networks, International Conference on Testbeds and Research Infrastructures*, pp. 260–278, Springer, 2022. 2.2.5, 6.2.3, 7.2.2

[47] S. Shen and M. Chi, "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, 2016. 2.3, 5.1, 5.2.1, 5.2.4

[48] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018. 2.3, 2.3.2, 3.1, 3.2, 3.2, 5.2.1

[49] M. L. Puterman, "Markov decision processes," *Handbooks in operations research and management science*, vol. 2, pp. 331–434, 1990. 2.3.1

[50] E. Levin, R. Pieraccini, and W. Eckert, "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, pp. 201–204, IEEE, 1998. 2.3.1, 5.2.1

[51] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017. 2.3.3, 5.2.2

[52] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988. 2.3.3, 5.2.2

[53] D. A. Pomerleau, "Efficient training of artificial neural networks for autonomous navigation," *Neural computation*, vol. 3, no. 1, pp. 88–97, 1991. 2.3.3, 5.2.2

[54] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," *arXiv preprint arXiv:1805.01954*, 2018. 2.3.3, 4.3.4, 5.4.2, 5.4.2

[55] S. Ross, G. Gordon, and D. Bagnell, "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*, pp. 627–635, JMLR Workshop and Conference Proceedings, 2011. 2.3.3, 5.2.2

[56] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, p. 1, 2004. 2.3.3, 5.2.2

[57] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *Icml*, vol. 1, p. 2, 2000. 2.3.4, 3.1, 3.6.1, 5.2.3

[58] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022. 2.4, 6.1

[59] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User modeling and user-adapted interaction*, vol. 4, pp. 253–278, 1994. 2.4.1, 2.4.1, 4.2.2, 6.1, 7.2.1

[60] X. Huang, Q. Liu, C. Wang, H. Han, J. Ma, E. Chen, Y. Su, and S. Wang, "Constructing educational concept maps with multiple relationships from multi-source data," in *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1108–1113, IEEE, 2019. 2.4.1

[61] T. Käser, S. Klingler, A. G. Schwing, and M. Gross, "Dynamic bayesian networks for student modeling," *IEEE Transactions on Learning Technologies*, vol. 10, no. 4, pp. 450–462, 2017. 2.4.1, 7.2.1

[62] R. Pelánek, "Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques," *User Modeling and User-Adapted Interaction*, vol. 27, pp. 313–350, 2017. 2.4.2

[63] H. Cen, K. Koedinger, and B. Junker, "Learning factors analysis–a general method for cognitive model evaluation and improvement," in *Intelligent Tutoring Systems: 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, 2006. Proceedings 8*, pp. 164–175, Springer, 2006. 2.4.2, 7.2.1

[64] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger, "Performance factors analysis–a new alternative to knowledge tracing.," *Online Submission*, 2009. 2.4.2, 7.2.1

[65] J.-J. Vie and H. Kashima, "Knowledge tracing machines: Factorization machines for knowledge tracing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 750–757, 2019. 2.4.2

[66] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," *Advances in neural information processing systems*, vol. 28, 2015. 2.4.3, 4.2.2, 6.1, 7.2.1

[67] Y. Su, Q. Liu, Q. Liu, Z. Huang, Y. Yin, E. Chen, C. Ding, S. Wei, and G. Hu, "Exercise-enhanced sequential modeling for student performance prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018. 2.4.3

[68] A. Ghosh, N. Heffernan, and A. S. Lan, "Context-aware attentive knowledge tracing," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2330–2339, 2020. 2.4.3, 6.2.3, 6.3.3, 7.1, 7.2.1, 7.3.2, 7.3.3

[69] S. Pandey and G. Karypis, "A self-attentive model for knowledge tracing," *arXiv preprint arXiv:1907.06837*, 2019. 2.4.3, 4.2.2, 6.2.2, 6.2.3, 7.2.1

[70] H. Nakagawa, Y. Iwasawa, and Y. Matsuo, "Graph-based knowledge tracing: modeling student proficiency using graph neural network," in *IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 156–163, 2019. 2.4.3, 7.2.1

[71] A. Sfard, "On two metaphors for learning and the dangers of choosing just one," *Educational researcher*, vol. 27, no. 2, pp. 4–13, 1998. 2.5

[72] S. Paavola and K. Hakkarainen, "Knowledge creation and the concept of a human being: A phenomenological approach," *Journal of the Learning Sciences*, vol. 13, no. 3, pp. 377–395, 2004. 2.5

[73] E. Duval, "Attention please!: Learning analytics for visualization and recommendation," *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, pp. 9–17, 2011. 2.5

[74] J. Lave and E. Wenger, *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, 1991. 2.5

[75] E. Wenger, *Communities of practice: Learning, meaning, and identity*. Cambridge university press, 1998. 2.5

[76] P. Brusilovsky, "Adaptive and intelligent technologies for web-based education," in *Special Issue on Intelligent Systems and Teleteaching*, vol. 4, pp. 19–25, Künstliche Intelligenz, 1999. 2.5

[77] M. Scardamalia and C. Bereiter, "Computer support for knowledge-building communities," in *The Journal of the Learning Sciences*, vol. 3, pp. 265–283, Taylor & Francis, 1994. 2.5

[78] D. H. Jonassen, *Constructing learning environments on the web: Engaging students in meaningful learning*. Educational Technology Publications Englewood Cliffs, NJ, 1999. 2.5

[79] V. J. Shute, M. Ventura, M. Bauer, and D. Zapata-Rivera, "Stealth assessment in computer-based games to support learning," *Computer games and instruction*, vol. 55, no. 2, pp. 503–524, 2013. 2.5

[80] G. Stahl, "Group cognition: Computer support for building collaborative knowledge," *Acting with technology*, 2006. 2.5

[81] J. L. Kolodner, *Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting learning by design(tm) into practice*. The Journal of the Learning Sciences, 2003. 2.5

[82] G. Zarkadakis, *In our own image: will artificial intelligence save or destroy us?* Random House, 2015. 3.1

[83] A. Sloman, "Did searle attack strong strong or weak strong ai," *Artificial Intelligence and Its Applications, John Wiley and Sons*, 1986. 3.1

[84] E. Gibney, "Google ai algorithm masters ancient game of go," *Nature News*, vol. 529, no. 7587, p. 445, 2016. 3.1, 3.2

[85] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *Ai Magazine*, vol. 35, no. 4, pp. 105–120, 2014. 3.1

[86] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015. 3.1, 3.1

[87] Y. Gao, J. Peters, A. Tsourdos, S. Zhifei, and E. M. Joo, "A survey of inverse reinforcement learning techniques," *International Journal of Intelligent Computing and Cybernetics*, 2012. 3.1

[88] J. Leike, D. Krueger, T. Everitt, M. Martic, V. Maini, and S. Legg, "Scalable agent alignment via reward modeling: a research direction," *arXiv preprint arXiv:1811.07871*, 2018. 3.1

[89] S. Suran, V. Pattanaik, and D. Draheim, "Frameworks for collective intelligence: A systematic literature review," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–36, 2020. 3.1

[90] A. Classen, P. Heymans, and P.-Y. Schobbens, "What's in a feature: A requirements engineering perspective," in *International Conference on Fundamental Approaches to Software Engineering*, pp. 16–30, Springer, 2008. 3.1

[91] Z. Li, L. Shi, A. I. Cristea, and Y. Zhou, "A survey of collaborative reinforcement learning: Interactive methods and design patterns," in *Designing Interactive Systems Conference 2021*, pp. 1579–1590, 2021. 3.1, 5.2.1

[92] G. Schwalbe and M. Schels, "A survey on methods for the safety assurance of machine learning based systems," in *10th European Congress on Embedded Real Time Software and Systems (ERTS 2020)*, 2020. 3.2

[93] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018. 3.2, 3.1, 3.5.1

[94] R. Bellman, "Dynamic programming and a new formalism in the theory of integral equations," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 41, no. 1, p. 31, 1955. 3.2

[95] R. Bellman, "A markovian decision process," *Journal of mathematics and mechanics*, pp. 679–684, 1957. 3.2

[96] J. H. Andreae, "Stella: A scheme for a learning machine," *IFAC Proceedings Volumes*, vol. 1, no. 2, pp. 497–502, 1963. 3.2

[97] J. H. Holland *et al.*, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* MIT press, 1992. 3.2

[98] D. Michie, "Problem decomposition and the learning of skills," in *European Conference on Machine Learning*, pp. 17–31, Springer, 1995. 3.2

[99] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*, vol. 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994. 3.2

[100] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 2, pp. 1322–1328, IEEE, 1999. 3.2

[101] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013. 3.2, 3.1

[102] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," *arXiv preprint arXiv:1712.01815*, 2017. 3.2, 3.1

[103] G. Lample and D. S. Chaplot, "Playing fps games with deep reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017. 3.2

[104] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013. 3.2

[105] A. Bernstein and E. Burnaev, "Reinforcement learning in computer vision," in *Tenth International Conference on Machine Vision (ICMV 2017)*, vol. 10696, p. 106961S, International Society for Optics and Photonics, 2018. 3.2, 3.5.1

[106] G. Branwen, "Gpt-3 creative fiction," 2020. 3.2

[107] D. Rohde, S. Bonner, T. Dunlop, F. Vasile, and A. Karatzoglou, "Recogym: A reinforcement learning environment for the problem of product recommendation in online advertising," *arXiv preprint arXiv:1808.00720*, 2018. 3.2

[108] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. Christiano, "Learning to summarize from human feedback," *arXiv preprint arXiv:2009.01325*, 2020. 3.2

[109] X. Zhang, L. Yao, X. Wang, J. Monaghan, D. Mcalpine, and Y. Zhang, "A survey on deep learning-based non-invasive brain signals: recent advances and new frontiers," *Journal of Neural Engineering*, vol. 18, no. 3, p. 031002, 2021. 3.2, 3.1, 3.6.2

[110] F. Cruz, G. I. Parisi, J. Twiefel, and S. Wermter, "Multi-modal integration of dynamic audiovisual patterns for an interactive reinforcement learning scenario," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 759–766, IEEE, 2016. 3.3.2, 3.1, 3.7.1

[111] K. van den Bosch and A. Bronkhorst, "Human-ai cooperation to benefit military decision making," NATO, 2018. 3.1, 3.4.2, 3.9

[112] M. Johnson, J. M. Bradshaw, P. J. Feltovich, C. M. Jonker, M. B. Van Riemsdijk, and M. Sierhuis, "Coactive design: Designing support for interdependence in joint activity," *Journal of Human-Robot Interaction*, vol. 3, no. 1, pp. 43–69, 2014. 3.1, 3.4.3, 3.5.2, 3.5.2, 3.9

[113] S. Zieba, P. Polet, F. Vanderhaegen, and S. Debernard, "Principles of adjustable autonomy: a framework for resilient human–machine cooperation," *Cognition, Technology & Work*, vol. 12, no. 3, pp. 193–203, 2010. 3.1, 3.4.4, 3.5.2, 3.9

[114] I. Moon, M. Lee, J. Ryu, and M. Mun, "Intelligent robotic wheelchair with emg-, gesture-, and voice-based interfaces," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 4, pp. 3453–3458, IEEE, 2003. 3.1, 3.7.2

[115] M. Wu, S. Parbhoo, M. C. Hughes, R. Kindle, L. A. Celi, M. Zazzi, V. Roth, and F. Doshi-Velez, "Regional tree regularization for interpretability in deep neural networks.," in *AAAI*, pp. 6413–6421, 2020. 3.1, 3.5.1

[116] A. Verma, V. Murali, R. Singh, P. Kohli, and S. Chaudhuri, "Programmatically interpretable reinforcement learning," *arXiv preprint arXiv:1804.02477*, 2018. 3.1, 3.5.1

[117] G. Liu, O. Schulte, W. Zhu, and Q. Li, "Toward interpretable deep reinforcement learning with linear model u-trees," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 414–429, Springer, 2018. 3.1, 3.5.1

[118] P. Madumal, T. Miller, L. Sonenberg, and F. Vetere, "Explainable reinforcement learning through a causal lens," *arXiv preprint arXiv:1905.10958*, 2019. 3.1, 3.5.1

[119] K. van den Bosch, T. Schoonderwoerd, R. Blankendaal, and M. Neerincx, "Six challenges for human-ai co-learning," in *International Conference on Human-Computer Interaction*, pp. 572–589, Springer, 2019. 3.1, 3.5.2

[120] P. S. Chacón and M. Eger, "Pandemic as a challenge for human-ai cooperation," in *Proceedings of the AIIDE workshop on Experimental AI in Games*, 2019. 3.1, 3.5.2

[121] R. E. Cardona-Rivera and R. M. Young, "Games as conversation," in *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014. 3.1, 3.5.2

[122] C. Liang, J. Proft, E. Andersen, and R. A. Knepper, "Implicit communication of actionable information in human-ai teams," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2019. 3.1, 3.5.2

[123] R. KLING, "Routine decision-making-the future of bureaucracy-inbar, m," 1981. 3.1, 3.5.3

[124] G. Irving, P. Christiano, and D. Amodei, "Ai safety via debate," *arXiv preprint arXiv:1805.00899*, 2018. 3.1, 3.5.3

[125] J. R. Frederiksen, B. Y. White, and J. Gutwill, "Dynamic mental models in learning science: The importance of constructing derivational linkages among models," *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, vol. 36, no. 7, pp. 806–836, 1999. 3.1

[126] D. Arumugam, J. K. Lee, S. Saskin, and M. L. Littman, "Deep reinforcement learning from policy-dependent human feedback," *arXiv preprint arXiv:1902.04257*, 2019. 3.1, 3.8.3

[127] M. Hagl and D. R. Kouabenan, "Safe on the road–does advanced driver-assistance systems use affect road risk perception?," *Transportation research part F: traffic psychology and behaviour*, vol. 73, pp. 488–498, 2020. 3.1, 4

[128] A. Dafoe, E. Hughes, Y. Bachrach, T. Collins, K. R. McKee, J. Z. Leibo, K. Larson, and T. Graepel, "Open problems in cooperative ai," *arXiv preprint arXiv:2012.08630*, 2020. 3.1, 3.3.3, 3.5.4, 3.6, 3.6.2, 3.6.2, 3.6.2, 3.6.4, 3.6.4

[129] R. Zhang, Q. Lv, J. Li, J. Bao, T. Liu, and S. Liu, "A reinforcement learning method for human-robot collaboration in assembly tasks," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102227, 2022. 3.1

[130] L. Floridi and M. Chiriatti, "Gpt-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, no. 4, pp. 681–694, 2020. 3.1, 6.2.1, 7.2.2

[131] J. Hovi, *Games, threats, and treaties: understanding commitments in international relations*. Burns & Oates, 1998. 3.1, 3.6.1

[132] S. V. Albrecht and P. Stone, "Autonomous agents modelling other agents: A comprehensive survey and open problems," *Artificial Intelligence*, vol. 258, pp. 66–95, 2018. 3.1, 3.6.1

[133] S. Armstrong and S. Mindermann, "Occam's razor is insufficient to infer the preferences of irrational agents," *arXiv preprint arXiv:1712.05812*, 2017. 3.1, 3.6.1

[134] K. Amin, N. Jiang, and S. Singh, "Repeated inverse reinforcement learning," *arXiv preprint arXiv:1705.05427*, 2017. 3.1, 3.6.1

[135] S. Bai, J. Wang, F. Chen, and B. Englot, "Information-theoretic exploration with bayesian optimization," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1816–1822, IEEE, 2016. 3.1, 3.6.1

[136] H. de Weerd, R. Verbrugge, and B. Verheij, "Negotiating with other minds: the role of recursive theory of mind in negotiation with incomplete information," *Autonomous Agents and Multi-Agent Systems*, vol. 31, no. 2, pp. 250–287, 2017. 3.1, 3.6.1

[137] N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes, *et al.*, "The hanabi challenge: A new frontier for ai research," *Artificial Intelligence*, vol. 280, p. 103216, 2020. 3.1, 3.6.1

[138] K. Wagner, J. A. Reggia, J. Uriagereka, and G. S. Wilkinson, "Progress in the simulation of emergent communication and language," *Adaptive Behavior*, vol. 11, no. 1, pp. 37–69, 2003. 3.1, 3.6.2

[139] S. Sukhbaatar, R. Fergus, *et al.*, "Learning multiagent communication with backpropagation," *Advances in neural information processing systems*, vol. 29, pp. 2244–2252, 2016. 3.1, 3.6.2

[140] J. N. Foerster, Y. M. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *arXiv preprint arXiv:1605.06676*, 2016. 3.1, 3.6.2

[141] K. Cao, A. Lazaridou, M. Lanctot, J. Z. Leibo, K. Tuyls, and S. Clark, "Emergent communication through negotiation," *arXiv preprint arXiv:1804.03980*, 2018. 3.1, 3.6.2

[142] A. Raza, M. A. Shah, H. A. Khattak, C. Maple, F. Al-Turjman, and H. T. Rauf, "Collaborative multi-agents in dynamic industrial internet of things using deep reinforcement learning," *Environment, Development and Sustainability*, pp. 1–19, 2022. 3.1

[143] J. D. Fearon, "Rationalist explanations for war," *International organization*, vol. 49, no. 3, pp. 379–414, 1995. 3.1, 3.6.3

[144] A. Sen, "Goals, commitment, and identity," *JL Econ. & Org.*, vol. 1, p. 341, 1985. 3.1, 3.6.3

[145] D. C. North, "Institutions and credible commitment," *Journal of Institutional and Theoretical Economics (JITE)/Zeitschrift für die gesamte Staatswissenschaft*, pp. 11–23, 1993. 3.1, 3.6.3

[146] K. Bagwell, "Commitment and observability in games," *Games and Economic Behavior*, vol. 8, no. 2, pp. 271–280, 1995. 3.1, 3.6.3

[147] M. O. Jackson and M. Morelli, "The reasons for wars: an updated survey," in *The handbook on the political economy of war*, Edward Elgar Publishing, 2011. 3.1, 3.6.3

[148] J. Criswell, N. Dautenhahn, and V. Adve, "Kcofi: Complete control-flow integrity for commodity operating system kernels," in *2014 IEEE Symposium on Security and Privacy*, pp. 292–307, IEEE, 2014. 3.1, 3.6.3

[149] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 254–269, 2016. 3.1, 3.6.3

[150] C. K. Frantz and M. Nowostawski, "From institutions to code: Towards automated generation of smart contracts," in *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W)*, pp. 210–215, IEEE, 2016. 3.1, 3.6.3

[151] R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," *IEEE Transactions on computers*, vol. 29, no. 12, pp. 1104–1113, 1980. 3.1, 3.6.4

[152] B. Horling and V. Lesser, "A survey of multi-agent organizational paradigms," *The Knowledge engineering review*, vol. 19, no. 4, pp. 281–316, 2004. 3.1, 3.6.4

[153] J. Ferber and G. Weiss, *Multi-agent systems: an introduction to distributed artificial intelligence*, vol. 1. Addison-Wesley Reading, 1999. 3.1, 3.6.4

[154] A. H. Bond and L. Gasser, *Readings in distributed artificial intelligence*. Morgan Kaufmann, 2014. 3.1, 3.6.4

[155] J. Vázquez-Salceda, V. Dignum, and F. Dignum, "Organizing multiagent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 307–360, 2005. 3.1, 3.6.4

[156] P. Dütting, F. Fischer, P. Jirapinyo, J. K. Lai, B. Lubin, and D. C. Parkes, "Payment rules through discriminant-based classifiers," 2015. 3.1, 3.6.4

[157] P. Dütting, Z. Feng, H. Narasimhan, D. Parkes, and S. S. Ravindranath, "Optimal auctions through deep learning," in *International Conference on Machine Learning*, pp. 1706–1715, PMLR, 2019. 3.1, 3.6.4

[158] A. Tacchetti, D. Strouse, M. Garnelo, T. Graepel, and Y. Bachrach, "A neural architecture for designing truthful and efficient auctions," *arXiv preprint arXiv:1907.05181*, 2019. 3.1, 3.6.4

[159] O. Amir, G. Sharon, and R. Stern, "Multi-agent pathfinding as a combinatorial auction," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. 3.1, 3.6.4

[160] W. B. Knox and P. Stone, "Interactively shaping agents via human reinforcement: The tamer framework," in *Proceedings of the fifth international conference on Knowledge capture*, pp. 9–16, 2009. 3.1, 3.7.1, 3.8.1

[161] R. Loftin, B. Peng, J. MacGlashan, M. L. Littman, M. E. Taylor, J. Huang, and D. L. Roberts, "Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning," *Autonomous agents and multi-agent systems*, vol. 30, no. 1, pp. 30–59, 2016. 3.1, 3.7.1, 3.8.3

[162] J. MacGlashan, M. K. Ho, R. Loftin, B. Peng, D. Roberts, M. E. Taylor, and M. L. Littman, "Interactive learning from policy-dependent human feedback," *arXiv preprint arXiv:1701.06049*, 2017. 3.1, 3.7.1, 3.8.3

[163] A. L. Thomaz, G. Hoffman, and C. Breazeal, "Real-time interactive reinforcement learning for robots," in *AAAI 2005 workshop on human comprehensible machine learning*, 2005. 3.1, 3.7.1

[164] A. L. Thomaz and C. Breazeal, "Teachable robots: Understanding human teaching behavior to build more effective robot learners," *Artificial Intelligence*, vol. 172, no. 6-7, pp. 716–737, 2008. 3.1, 3.7.1, 3.8.5, 3.9

[165] G. Warnell, N. Waytowich, V. Lawhern, and P. Stone, "Deep tamer: Interactive agent shaping in high-dimensional state spaces," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018. 3.1, 3.7.1

[166] S. Ciranka, J. Linde-Domingo, I. Padezhki, C. Wicharz, C. M. Wu, and B. Spitzer, "Asymmetric reinforcement learning facilitates human inference of transitive relations," *Nature Human Behaviour*, vol. 6, no. 4, pp. 555–564, 2022. 3.1

[167] R. M. Voyles and P. K. Khosla, "Gesture-based programming: A preliminary demonstration," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 1, pp. 708–713, IEEE, 1999. 3.1, 3.7.2

[168] G. Li, H. Dibeklioğlu, S. Whiteson, and H. Hung, "Facial feedback for reinforcement learning: a case study and offline analysis using the tamer framework," *Autonomous Agents and Multi-Agent Systems*, vol. 34, no. 1, pp. 1–29, 2020. 3.1, 3.7.2

[169] S. C. Gadanho, "Learning behavior-selection by emotions and cognition in a multi-goal robot task," *Journal of Machine Learning Research*, vol. 4, no. Jul, pp. 385–412, 2003. 3.1, 3.7.2

[170] R. Arakawa, S. Kobayashi, Y. Unno, Y. Tsuboi, and S.-i. Maeda, "Dqn-tamer: Human-in-the-loop reinforcement learning with intractable feedback," *arXiv preprint arXiv:1810.11748*, 2018. 3.1, 3.7.2

[171] G. Gordon, S. Spaulding, J. K. Westlund, J. J. Lee, L. Plummer, M. Martinez, M. Das, and C. Breazeal, "Affective personalization of a social robot tutor for children's second language skills," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 3951–3957, 2016. 3.1, 3.7.2

[172] P. Goyal, S. Niekum, and R. J. Mooney, "Using natural language for reward shaping in reinforcement learning," *arXiv preprint arXiv:1903.02020*, 2019. 3.1, 3.7.2

[173] R. Maclin and J. W. Shavlik, "Creating advice-taking reinforcement learners," *Machine Learning*, vol. 22, no. 1-3, pp. 251–281, 1996. 3.1, 3.7.2

[174] G. Kuhlmann, P. Stone, R. Mooney, and J. Shavlik, "Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer," in *The AAAI-2004 workshop on supervisory control of learning and adaptive systems*, San Jose, CA, 2004. 3.1, 3.7.2

[175] E. C. Williams, N. Gopalan, M. Rhee, and S. Tellex, "Learning to parse natural language to grounded reward functions with weak supervision," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7, IEEE, 2018. 3.1, 3.7.2

[176] R. Zhang, J. Lv, J. Li, J. Bao, P. Zheng, and T. Peng, "A graph-based reinforcement learning-enabled approach for adaptive human-robot collaborative assembly operations," *Journal of Manufacturing Systems*, vol. 63, pp. 491–503, 2022. 3.1

[177] A. H. Yoo, H. Keglovits, and A. Collins, "The importance of linguistic information in human reinforcement learning," 2022. 3.1

[178] M. Eppe, C. Gumbsch, M. Kerzel, P. D. Nguyen, M. V. Butz, and S. Wermter, "Intelligent problem-solving as integrated hierarchical reinforcement learning," *Nature Machine Intelligence*, vol. 4, no. 1, pp. 11–20, 2022. 3.1

[179] F. Quek, D. McNeill, R. Bryll, S. Duncan, X.-F. Ma, C. Kirbas, K. E. McCullough, and R. Ansari, "Multimodal human discourse: gesture and speech," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 9, no. 3, pp. 171–193, 2002. 3.1, 3.7.3

[180] K. Weber, H. Ritschel, F. Lingenfelser, and E. André, "Real-time adaptation of a robotic joke teller based on human social signals," 2018. 3.1, 3.7.3

[181] C. Isbell, C. R. Shelton, M. Kearns, S. Singh, and P. Stone, "A social reinforcement learning agent," in *Proceedings of the fifth international conference on Autonomous agents*, pp. 377–384, 2001. 3.1, 3.7, 3.7.3

[182] P. Lessel, M. Altmeyer, L. V. Schmeer, and A. Krüger, "" enable or disable gamification?" analyzing the impact of choice in a gamified image tagging task," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2019. 3.1, 3.7.3

[183] M. K. Ho, M. L. Littman, F. Cushman, and J. L. Austerweil, "Teaching with rewards and punishments: Reinforcement or communication?," in *CogSci*, 2015. 3.1, 3.7.3

[184] S. Lee, R. Yu, J. Xie, S. M. Billah, and J. M. Carroll, "Opportunities for human-ai collaboration in remote sighted assistance," in *27th International Conference on Intelligent User Interfaces*, pp. 63–78, 2022. 3.1

[185] J. Hitsuwari, Y. Ueda, W. Yun, and M. Nomura, "Does human–ai collaboration lead to more creative art? aesthetic evaluation of human-made and ai-generated haiku poetry," *Computers in Human Behavior*, p. 107502, 2022. 3.1

[186] J. Wu, Z. Huang, Z. Hu, and C. Lv, "Toward human-in-the-loop ai: Enhancing deep reinforcement learning via real-time human guidance for autonomous driving," *Engineering*, 2022. 3.1

[187] S. Chen, J. Gao, S. Reddy, G. Berseth, A. D. Dragan, and S. Levine, "Asha: Assistive teleoperation via human-in-the-loop reinforcement learning," *arXiv preprint arXiv:2202.02465*, 2022. 3.1

[188] A. L. Thomaz, G. Hoffman, and C. Breazeal, "Reinforcement learning with human teachers: Understanding how people want to teach robots," in *RO-MAN 2006-The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 352–357, IEEE, 2006. 3.1, 3.8.1

[189] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009. 3.1, 3.8.1, 3.8.1, 3.8.5, 3.9

[190] S. Griffith, K. Subramanian, J. Scholz, C. L. Isbell, and A. L. Thomaz, "Policy shaping: Integrating human feedback with reinforcement learning," in *Advances in neural information processing systems*, pp. 2625–2633, 2013. 3.1, 3.7.3, 3.8.3, 3.9

[191] M. E. Taylor, H. B. Suay, and S. Chernova, "Integrating reinforcement learning with human demonstrations of varying ability," in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 617–624, 2011. 3.1, 3.8.4, 3.9

[192] M. Li, T. Brys, and D. Kudenko, "Introspective reinforcement learning and learning from demonstration.," in *AAMAS*, pp. 1992–1994, 2018. 3.1, 3.8.4

[193] A. L. Thomaz and C. Breazeal, "Adding guidance to interactive reinforcement learning," in *Proceedings of the Twentieth Conference on Artificial Intelligence (AAAI)*, 2006. 3.1, 3.8.5

[194] R. Ferenc, A. Beszedes, L. Fulop, and J. Lele, "Design pattern mining enhanced by machine learning," in *21st IEEE International Conference on Software Maintenance (ICSM'05)*, pp. 295–304, IEEE, 2005. 3.4

[195] W. Y. Wang, J. Li, and X. He, "Deep reinforcement learning for nlp," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pp. 19–21, 2018. 3.5.1

[196] A. Holzinger, "From machine learning to explainable ai," in *2018 world symposium on digital intelligence for systems and machines (DISA)*, pp. 55–66, IEEE, 2018. 3.5.1

[197] D. Wilson and D. Sperber, "On grice's theory of conversation," *Conversation and discourse*, pp. 155–78, 1981. 3.5.2

[198] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012. 3.5.3

[199] H. C. Barrett, "Deciding what to observe: Thoughts for a post-weird generation," *Evolution and Human Behavior*, vol. 41, no. 5, pp. 445–453, 2020. 3.6.1

[200] J. S. Rosenschein and G. Zlotkin, *Rules of encounter: designing conventions for automated negotiation among computers*. MIT press, 1994. 3.6.4

[201] E. Hutchins, *Cognition in the Wild*. Cambridge, MA, USA: MIT Press, 1995. 3.6.5, 3.6.6

[202] W. B. Knox, "Learning from human-generated reward," 2012. 3.7.1

[203] C. Arzate Cruz and T. Igarashi, "Interactive reinforcement learning for autonomous behavior design," in *Artificial Intelligence for Human Computer Interaction: A Modern Approach*, pp. 345–375, Springer, 2021. 3.8.1

[204] A. Rosenfeld, M. Cohen, M. E. Taylor, and S. Kraus, "Leveraging human knowledge in tabular reinforcement learning: A study of human subjects," *The Knowledge Engineering Review*, vol. 33, 2018. 3.8.1

[205] R. A. Bianchi, M. F. Martins, C. H. Ribeiro, and A. H. Costa, "Heuristically-accelerated multiagent reinforcement learning," *IEEE transactions on cybernetics*, vol. 44, no. 2, pp. 252–265, 2013. 3.8.1

[206] S. Mindermann, R. Shah, A. Gleave, and D. Hadfield-Menell, "Active inverse reward design," *arXiv preprint arXiv:1809.03060*, 2018. 3.8.2

[207] D. Hadfield-Menell, S. Milli, P. Abbeel, S. Russell, and A. Dragan, "Inverse reward design," *arXiv preprint arXiv:1711.02827*, 2017. 3.8.2

[208] S. Krening and K. M. Feigh, "Interaction algorithm effect on human experience with reinforcement learning," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 7, no. 2, pp. 1–22, 2018. 3.8.3

[209] H. B. Suay and S. Chernova, "Effect of human guidance and state space size on interactive reinforcement learning," in *2011 Ro-Man*, pp. 1–6, IEEE, 2011. 3.8.5

[210] C. Yu, T. Yang, W. Zhu, G. Li, *et al.*, "Learning shaping strategies in human-in-the-loop interactive reinforcement learning," *arXiv preprint arXiv:1811.04272*, 2018. 3.8.5

[211] K. VanLehn, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," *Educational psychologist*, vol. 46, no. 4, pp. 197–221, 2011. 4

[212] D. Weitekamp, E. Harpstead, and K. R. Koedinger, "An interaction design for machine teaching to develop ai tutors," in *Proceedings of the 2020 CHI conference on human factors in computing systems*, pp. 1–11, 2020. 4.1

[213] S. Vincent-Lancrin and R. Van der Vlies, "Trustworthy artificial intelligence (ai) in education: Promises and challenges," 2020. 4.1

[214] S. J. Yang, "Guest editorial: Precision education-a new challenge for ai in education.," *Journal of Educational Technology & Society*, vol. 24, no. 1, 2021. 4.1

[215] P. Brusilovsky, "Adaptive hypermedia for education and training," *Adaptive technologies for training and education*, vol. 46, pp. 46–68, 2012. 4.2.1, 5.2.4

[216] M.-E. T. Horntvedt, A. Nordsteien, T. Fermann, and E. Severinsson, "Strategies for teaching evidence-based practice in nursing education: a thematic literature review," *BMC medical education*, vol. 18, no. 1, pp. 1–11, 2018. 4.2.1

[217] D. Bouhnik and T. Marcus, "Interaction in distance-learning courses," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, pp. 299–305, 2006. 4.2.1

[218] K. Chrysafiadi and M. Virvou, "Student modeling approaches: A literature review for the last decade," *Expert Systems with Applications*, vol. 40, no. 11, pp. 4715–4729, 2013. 4.2.1

[219] A. Abyaa, M. K. Idrissi, and S. Bennani, "Learner modelling: systematic review of the literature from the last 5 years," *Educational Technology Research and Development*, vol. 67, no. 5, pp. 1105–1143, 2019. 4.2.1

[220] T. Jackson, E. Mathews, K.-I. Lin, A. Olney, and A. Graesser, "Modeling student performance to enhance the pedagogy of autotutor," in *International Conference on user modeling*, pp. 368–372, Springer, 2003. 4.2.1

[221] S. Ouf, M. Abd Ellatif, S. E. Salama, and Y. Helmy, "A proposed paradigm for smart learning environment based on semantic web," *Computers in Human Behavior*, vol. 72, pp. 796–818, 2017. 4.2.1

[222] M. Makatchev, P. W. Jordan, and K. VanLehn, "Modeling students' reasoning about qualitative physics: Heuristics for abductive proof search," in *International Conference on intelligent tutoring systems*, pp. 699–709, Springer, 2004. 4.2.1

[223] M. A. Chatti, D. Dugoija, H. Thüs, and U. Schroeder, "Learner modeling in academic networks," in *2014 IEEE 14th International Conference on Advanced Learning Technologies*, pp. 117–121, IEEE, 2014. 4.2.1

[224] N. Adel, A. Latham, and K. A. Crockett, "Towards socially intelligent automated tutors: Predicting learning style dimensions from conversational dialogue," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pp. 315–320, IEEE, 2016. 4.2.1

[225] K. Chrysafiadi, M. Virvou, *et al.*, *Advances in personalized web-based education*. Springer, 2015. 4.2.1

[226] S. P. Lajoie, "Student modeling for individuals and groups: The bioworld and howard platforms," *International Journal of Artificial Intelligence in Education*, vol. 31, no. 3, pp. 460–475, 2021. 4.2.1

[227] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized bayesian knowledge tracing models," in *International conference on artificial intelligence in education*, pp. 171–180, Springer, 2013. 4.2.2, 5.2.4, 6.2.2, 7.2.1

[228] J. Kasurinen and U. Nikula, "Estimating programming knowledge with bayesian knowledge tracing," *ACM SIGCSE Bulletin*, vol. 41, no. 3, pp. 313–317, 2009. 4.2.2

[229] M. Khajah, R. V. Lindsey, and M. C. Mozer, "How deep is knowledge tracing?," *arXiv preprint arXiv:1604.02416*, 2016. 4.2.2

[230] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, IEEE, 2018. 4.2.2

[231] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International Conference on Machine Learning*, pp. 4055–4064, PMLR, 2018. 4.2.3, 6.2.1, 7.2.2

[232] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014. 4.2.3

[233] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018. 4.2.3, 4.2.4

[234] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020. 4.2.4

[235] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, C. Bae, B. Kim, and J. Heo, "Ednet: A large-scale hierarchical dataset in education," in *International Conference on Artificial Intelligence in Education*, pp. 69–73, Springer, 2020. 4.3.2, 5.1, 5.2.4, 5.3.1, 5.3.2, 6.3.3, 7.1, 7.3.3, 7.3.3

[236] D. Shin, Y. Shim, H. Yu, S. Lee, B. Kim, and Y. Choi, "Saint+: Integrating temporal features for ednet correctness prediction," in *LAK21: 11th International Learning Analytics and Knowledge Conference*, pp. 490–496, 2021. 4.3.4, 6.2.2, 7.2.1

[237] L. N. Tolstoj and V. Gerasimov, *Anna Karenina*. BHB, 1969. 4.4

[238] F. E. Ritter, J. Nerb, E. Lehtinen, and T. M. O'Shea, *In order to learn: How the sequence of topics influences learning*. Oxford University Press, 2007. 5.1

[239] L. Shi, A. I. Cristea, M. S. K. Awan, M. Hendrix, and C. Stewart, "Towards understanding learning behavior patterns in social adaptive personalized e-learning systems.," Association for Information Systems, 2013. 5.1

[240] Z. Li, L. Shi, A. Cristea, Y. Zhou, C. Xiao, and Z. Pan, "Simstu-transformer: A transformer-based approach to simulating student behaviour," in *International Conference on Artificial Intelligence in Education*, pp. 348–351, Springer, 2022. 5.1, 5.5

[241] F. Jarboui, C. Gruson-Daniel, A. Durmus, V. Rocchisani, S.-H. Goulet Ebongue, A. Depoux, W. Kirschenmann, and V. Perchet, "Markov decision process for mooc users behavioral inference," in *European MOOCs Stakeholders Summit*, pp. 70–80, Springer, 2019. 5.1, 5.3.2

[242] M. Zimmer, P. Viappiani, and P. Weng, "Teacher-student framework: a reinforcement learning approach," in *AAMAS Workshop Autonomous Robots and Multirobot Systems*, 2014. 5.1

[243] C. W. Anderson, B. A. Draper, and D. A. Peterson, "Behavioral cloning of student pilots with modular neural networks," in *ICML*, pp. 25–32, 2000. 5.1

[244] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999. 5.1

[245] A. Iglesias, P. Martínez, R. Aler, and F. Fernández, "Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems," *Knowledge-Based Systems*, vol. 22, no. 4, pp. 266–270, 2009. 5.1, 5.2.4, 5.3.2

[246] I. Felstead, "Role modelling and students' professional development," *British Journal of Nursing*, vol. 22, no. 4, pp. 223–227, 2013. 5.1

[247] R. Bhattacharyya, B. Wulfe, D. Phillips, A. Kuefler, J. Morton, R. Senanayake, and M. Kochenderfer, "Modeling human driving behavior through generative adversarial imitation learning," *arXiv preprint arXiv:2006.06412*, 2020. 5.2.2

[248] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 661–668, JMLR Workshop and Conference Proceedings, 2010. 5.2.2

[249] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014. 5.2.3, 5.3.3

[250] L. Shi, D. Al Qudah, A. Qaffas, and A. I. Cristea, "Topolor: A social personalized adaptive e-learning system," in *User Modeling, Adaptation, and Personalization* (S. Carberry, S. Weibelzahl, A. Micarelli, and G. Semeraro, eds.), (Berlin, Heidelberg), pp. 338–340, Springer Berlin Heidelberg, 2013. 5.2.4

[251] L. Shi and A. I. Cristea, "Learners thrive using multifaceted open social learner modeling," *IEEE MultiMedia*, vol. 23, no. 1, pp. 36–47, 2016. 5.2.4

[252] L. Shi, A. I. Cristea, A. M. Toda, and W. Oliveira, "Exploring navigation styles in a futurelearn mooc," in *Intelligent Tutoring Systems* (V. Kumar and C. Troussas, eds.), (Cham), pp. 45–55, Springer International Publishing, 2020. 5.2.4

[253] S. Doroudi, V. Aleven, and E. Brunskill, "Where's the reward?," *International Journal of Artificial Intelligence in Education*, vol. 29, no. 4, pp. 568–620, 2019. 5.2.4

[254] R. K. Hambleton, H. Swaminathan, and H. J. Rogers, *Fundamentals of item response theory*, vol. 2. Sage, 1991. 5.2.4

[255] A. Segal, Y. B. David, J. J. Williams, K. Gal, and Y. Shalom, "Combining difficulty ranking with multi-armed bandits to sequence educational content," in *International conference on artificial intelligence in education*, pp. 317–321, Springer, 2018. 5.2.4, 5.3.2

[256] J. R. Tetreault and D. J. Litman, "A reinforcement learning approach to evaluating state representations in spoken dialogue systems," *Speech Communication*, vol. 50, no. 8-9, pp. 683–696, 2008. 5.2.4, 5.3.2

[257] J. Rowe, B. Pokorny, B. Goldberg, B. Mott, and J. Lester, "Toward simulated students for reinforcement learning-driven tutorial planning in gift," in *Proceedings of R. Sottilare (Ed.) 5th Annual GIFT Users Symposium. Orlando, FL*, 2017. 5.2.4

[258] M. Chi, K. VanLehn, and D. Litman, "Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutorial tactics," in *International conference on intelligent tutoring systems*, pp. 224–234, Springer, 2010. 5.2.4, 5.3.2

[259] J. Beck, B. P. Woolf, and C. R. Beal, "Advisor: A machine learning architecture for intelligent tutor construction," *AAAI/IAAI*, vol. 2000, no. 552-557, pp. 1–2, 2000. 5.2.4

[260] S. Shen and M. Chi, "Aim low: Correlation-based feature selection for model-based reinforcement learning.," *International Educational Data Mining Society*, 2016. 5.3.2

[261] J. Ho, J. Gupta, and S. Ermon, "Model-free imitation learning with policy optimization," in *International Conference on Machine Learning*, pp. 2760–2769, PMLR, 2016. 5.3.3

[262] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 5.3.3, 5.4.2

[263] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1179–1191, 2020. 5.4.2, 5.4.2

[264] S. Lefèvre, C. Sun, R. Bajcsy, and C. Laugier, "Comparison of parametric and non-parametric approaches for vehicle speed prediction," in *2014 American Control Conference*, pp. 3494–3499, IEEE, 2014. 5.4.2

[265] Z. A. Z. Azhar, "Designing an offline reinforcement learning based pedagogical agent with a large scale educational dataset," 5.4.2

[266] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators*. CRC press, 2017. 5.4.2

[267] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *International Conference on Machine Learning*, pp. 449–458, PMLR, 2017. 5.4.2

[268] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 4, pp. IV–317, IEEE, 2007. 5.5, 5.5.1

[269] C. Voloshin, H. M. Le, N. Jiang, and Y. Yue, "Empirical study of off-policy policy evaluation for reinforcement learning," *arXiv preprint arXiv:1911.06854*, 2019. 5.5.2

[270] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6023–6029, IEEE, 2019. 5.5.3

[271] M. Lapan, *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, 2018. 5.5.3

[272] L. Weaver and N. Tao, "The optimal reward baseline for gradient-based reinforcement learning," *arXiv preprint arXiv:1301.2315*, 2013. 5.5.3

[273] T. Mandel, Y.-E. Liu, S. Levine, E. Brunskill, and Z. Popovic, "Offline policy evaluation across representations with applications to educational games.," in *AAMAS*, vol. 1077, 2014. 5.5.3

[274] Y. Saito, T. Udagawa, H. Kiyohara, K. Mogi, Y. Narita, and K. Tateno, "Evaluating the robustness of off-policy evaluation," in *Fifteenth ACM Conference on Recommender Systems*, pp. 114–123, 2021. 5.5.3

[275] S. T. Tokdar and R. E. Kass, "Importance sampling: a review," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 1, pp. 54–60, 2010. 5.5.3, 5.5.3

[276] A. Tirinzoni, M. Salvini, and M. Restelli, "Transfer of samples in policy search via multiple importance sampling," in *International Conference on Machine Learning*, pp. 6264–6274, PMLR, 2019. 5.5.3, 5.5.3, 5.5.3

[277] C. R. Shelton, "Importance sampling for reinforcement learning with multiple objectives," 2001. 5.5.3

[278] S. Ju, S. Shen, H. Azizsoltani, T. Barnes, and M. Chi, "Importance sampling to identify empirically valid policies and their critical decisions.," in *EDM (Workshops)*, pp. 69–78, 2019. 5.5.3

[279] A. R. Mahmood, H. P. Van Hasselt, and R. S. Sutton, "Weighted importance sampling for off-policy learning with linear function approximation," *Advances in Neural Information Processing Systems*, vol. 27, 2014. 5.5.3

[280] S. Minn, J.-J. Vie, K. Takeuchi, H. Kashima, and F. Zhu, "Interpretable knowledge tracing: Simple and efficient student modeling with causal relations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 12810–12818, 2022. 6.1, 6.3.2

[281] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, *et al.*, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87–110, 2022. 6.2.1

[282] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, "Ammus: A survey of transformer-based pretrained models in natural language processing," *arXiv preprint arXiv:2108.05542*, 2021. 6.2.1, 7.2.2

[283] L. Wu, S. Li, C.-J. Hsieh, and J. Sharpnack, "Sse-pt: Sequential recommendation via personalized transformer," in *Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 328–337, 2020. 6.2.1

[284] B. D. Lund and T. Wang, "Chatting about chatgpt: how may ai and gpt impact academia and libraries?," *Library Hi Tech News*, 2023. 6.2.1

[285] G. Abdelrahman, Q. Wang, and B. Nunes, "Knowledge tracing: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–37, 2023. 6.2.2, 7.1, 7.2.1

[286] J. A. Drass, J. Muir-Nash, P. C. Boykin, J. M. Turek, and K. L. Baker, "Perceived and actual level of knowledge of diabetes mellitus among nurses," *Diabetes Care*, vol. 12, no. 5, pp. 351–356, 1989. 6.2.2

[287] U. Lee, Y. Park, Y. Kim, S. Choi, and H. Kim, "Monacobert: Monotonic attention based convbert for knowledge tracing," *arXiv preprint arXiv:2208.12615*, 2022. 6.2.2, 6.3.2, 6.3.3, 7.2.1

[288] T. Wang, F. Ma, and J. Gao, "Deep hierarchical knowledge tracing," in *Proceedings of the 12th international conference on educational data mining*, 2019. 6.2.3

[289] R. Krishnan, J. Singh, M. Sato, Q. Zhang, and T. Ohkuma, "Incorporating wide context information for deep knowledge tracing using attentional bi-interaction.," in *L2D@ WSDM*, pp. 1–13, 2021. 6.2.3

[290] S. Pandey and J. Srivastava, "Rkt: relation-aware self-attention for knowledge tracing," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1205–1214, 2020. 6.2.3

[291] L. Liu, X. Liu, J. Gao, W. Chen, and J. Han, "Understanding the difficulty of training transformers," *arXiv preprint arXiv:2004.08249*, 2020. 6.3.2

[292] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, "On layer normalization in the transformer architecture," in *International Conference on Machine Learning*, pp. 10524–10533, PMLR, 2020. 6.3.2

[293] H.-S. Chang, H.-J. Hsu, and K.-T. Chen, "Modeling exercise relationships in e-learning: A unified approach.," in *EDM*, pp. 532–535, 2015. 7.1, 7.3.3

[294] G. Rasch, *Probabilistic models for some intelligence and attainment tests.* ERIC, 1993. 7.1, 7.3.2

[295] A. Corbett, "Cognitive mastery learning in the act programming tutor," in *Adaptive User Interfaces. AAAI SS-00-01. Retrieved from https://aaai. org/Library/Symposia/Spring/ss00-01. php*, 2000. 7.2.1

[296] M. Villano, "Probabilistic student models: Bayesian belief networks and knowledge space theory," in *International Conference on Intelligent Tutoring Systems*, pp. 491–498, Springer, 1992. 7.2.1

[297] J. I. Lee and E. Brunskill, "The impact on individualizing student models on necessary practice opportunities.," *International Educational Data Mining Society*, 2012. 7.2.1

[298] Z. A. Pardos and N. T. Heffernan, "Modeling individualization in a bayesian networks implementation of knowledge tracing," in *International conference on user modeling, adaptation, and personalization*, pp. 255–266, Springer, 2010. 7.2.1

[299] Z. Tiana, G. Zhengc, B. Flanaganb, J. Mic, and H. Ogatab, "Bekt: Deep knowledge tracing with bidirectional encoder representations from transformers," 7.2.2

[300] X. Sun, X. Zhao, Y. Ma, X. Yuan, F. He, and J. Feng, "Muti-behavior features based knowledge tracking using decision tree improved dkvmn," in *Proceedings of the ACM Turing Celebration Conference-China*, pp. 1–6, 2019. 7.3.3

[301] X. Zhang, J. Zhang, N. Lin, and X. Yang, "Sequential self-attentive model for knowledge tracing," in *International Conference on Artificial Neural Networks*, pp. 318–330, Springer, 2021. 7.3.3