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

Zhaoxing Li[1][0000−0003−3560−3461], Jujie Yang[2][0000−0001−6024−2720], Jindi Wang[2][0000−0002−0901−8587], Lei Shi[3][0000−0001−7119−3207], Jiayi Feng[4][0000−0002−9009−4295], and Sebastian Stein[1][0000−0003−2858−8857]

[1] School of Electronics and Computer Science, University of Southampton, Southampton, UK
[2] Department of Computer Science, Durham University, Durham, UK
[3] Open Lab, School of Computing, Newcastle University, Newcastle upon Tyne, UK
[4] Beijing Jiaotong University, Beijing, China
zhaoxing.li@soton.ac.uk,{jujie.yang, jindi.wang}@durham.ac.uk, jyfeng@bjtu.edu.cn, lei.shi@ncl.ac.uk, ss2@ecs.soton.ac.uk

Abstract. The field of Knowledge Tracing (KT) aims to understand how students learn and master knowledge over time by analyzing their historical behaviour data. To achieve this goal, many researchers have proposed KT models that use data from Intelligent Tutoring Systems (ITS) to predict students' subsequent actions. However, 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, we propose 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.

Keywords: Knowledge Tracing · BERT · Student Modelling · Long-Sequence Data Processing · Intelligent Tutoring Systems

1 Introduction

As one of the widely applied intelligent educational 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 [2]. With the development of ITS, large-scale datasets such as *EdNet* [5] and *Junyi Academy* [4] 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 [18]. 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. 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 [15].

In general, KT models could be divided into three categories: probabilistic KT models, logistic KT models, and deep learning based KT methods (DKT)[2]. 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 accuracy, speed, and memory usage [18]. Therefore, allowing the processing of very long sequence data is key to achieving high performance for next-generation KT models. 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 [8].

To address the above issues, in this paper, 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 [7] 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 [21], which could be utilised to construct raw questions and knowledge embeddings for KT tasks [8]. 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 our paper lie in the following two aspects:

1. We propose LBKT, 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).

2 Related Work

2.1 Knowledge Tracing

Knowledge Tracing (KT) models and predicts students' mastery levels over time in Intelligent Tutoring Systems, using observable behaviors to infer hidden knowledge states [1]. It aims to personalize feedback and instruction, enhancing learning outcomes. KT methods are categorized into probabilistic, logistic, and deep learning-based models [6, 31, 29].

Probabilistic models, like Bayesian Knowledge Tracing (BKT), utilize Hidden Markov Models or Bayesian Belief Networks to track learning states, but struggle with complexity and multi-skill scenarios [6, 27, 10, 30]. Logistic models apply logistic regression to predict mastery levels, incorporating factors like prior performance and response time [3, 20, 10, 28].

Deep learning-based KT, leveraging advancements like self-attention mechanisms and Transformer architectures, has introduced models such as SAKT and SAINT+ for higher performance through sequence prediction and attention to temporal learning dynamics [19, 22, 8]. BERT-based KT models, though innovative, have not surpassed state-of-the-art KT methods in handling long-sequence, large-scale datasets [11, 25].

2.2 Transformer-based Model and Application

Transformers, with self-attention mechanisms, have revolutionized NLP and image generation, exemplified by BERT and GPT [26, 7]. BERT's bidirectional training and large pre-training corpus have set new benchmarks in understanding natural language, with applications extending into image processing, recommendation systems, and music generation [7, 9, 23]. Despite their success, BERT variants in KT have not achieved superior performance on complex, long-sequence datasets [25, 13, 14, 17, 12, 16].

3 Methodology

3.1 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 [25]. 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 [21]. 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 among different features in long-sequence data, as illustrated in Fig. 1.

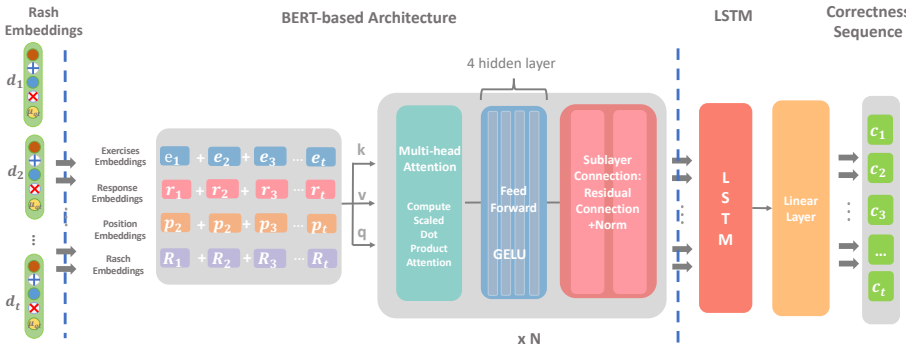

Fig. 1. The architecture of LBKT.

The first component of LBKT is the Rasch model-based embeddings proposed by Ghosh [8]. 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:

$$E = E_{\text{Rasch}} + E_{\text{Bert Token}} + E_{\text{Position}} \tag{1}$$

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

$$E_{\text{Rasch}} = E_{\mathrm{d}} + E_{\mathrm{d}} \times E_{\mathrm{q}} \tag{2}$$

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 feed-forward 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:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

$$\text{FFN}(x) = \text{GELU}(W_1 x + b_1)W_2 + b_2 \tag{4}$$

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.

3.2 Experiment Setting

Datasets We used five benchmark datasets to validate the effectiveness of the LBKT model, including assist12[5], assist17[6], algebra06 [7], EdNet [5] [8], and Junyi Academy[4][9]. 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. 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

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

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.

Baseline Models We compared our LBKT to three state-of-the-art models, BEKT [25], AKT [8], DKVMN [24], as well as the two top baseline models in the Riiid Answer Correctness Prediction Competition provided by Kaggle[10], including SSAKT[32], and LTMTI[5].

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.

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 block/encoder layer was set to 12.

4 Results and Discussion

4.1 Overall Performance

LBKT outperforms four baseline models on most metrics in the experiments on five benchmark datasets. Tabel 1 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 the unique architecture of our LBKT model. 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.

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

Table 1. 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	**0.799**	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	ACC	**0.832**	0.807	0.777	0.797	0.791	0.790
Academy	AUC	**0.851**	0.831	0.845	0.812	0.799	0.769

Tabel 2 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 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 2. 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

4.2 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. 2-*left* shows the results of No-Rasch-embedding, and Fig.2-*right* 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 2-*right*, the difficult level embeddings were separated to avoid mixing with easy level embeddings.

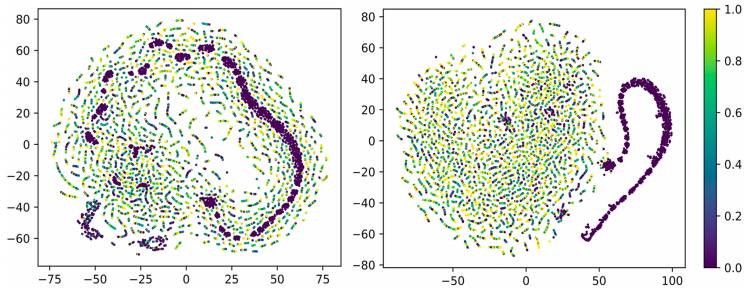

Fig. 2. Visualisation of the embedding vector using t-SNE: *without* Rasch embeddings (on the left) and *with* Rasch embeddings (on the right). The colour bar is the predicted probability of the outputs.

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.

5 Conclusion

In this study, 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. 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.

6 Acknowledgments

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through a Turing AI Fellowship (EP/V022067/1) on Citizen-Centric AI Systems (https://ccais.ac.uk/) and through the AutoTrust Platform Grant (EP/R029563/1).

References

1. Abdelrahman, G., Wang, Q., Nunes, B.: Knowledge tracing: A survey. ACM Computing Surveys **55**(11), 1–37 (2023)
2. Abdelrahman, G., Wang, Q., Nunes, B.P.: Knowledge tracing: A survey. ACM Computing Surveys (2022)
3. Cen, H., Koedinger, K., Junker, B.: 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)
4. Chang, H.S., Hsu, H.J., Chen, K.T.: Modeling exercise relationships in e-learning: A unified approach. In: EDM. pp. 532–535 (2015)
5. Choi, Y., Lee, Y., Shin, D., Cho, J., Park, S., Lee, S., Baek, J., Bae, C., Kim, B., Heo, J.: Ednet: A large-scale hierarchical dataset in education. In: International Conference on Artificial Intelligence in Education. pp. 69–73. Springer (2020)
6. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction (1994)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
8. Ghosh, A., Heffernan, N., Lan, A.S.: Context-aware attentive knowledge tracing. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2330–2339 (2020)
9. Jiang, Z.H., Yu, W., Zhou, D., Chen, Y., Feng, J., Yan, S.: Convbert: Improving bert with span-based dynamic convolution. Advances in Neural Information Processing Systems **33**, 12837–12848 (2020)
10. Käser, T., Klingler, S., Schwing, A.G., Gross, M.: Dynamic bayesian networks for student modeling. IEEE Transactions on Learning Technologies (2017)
11. Lee, U., Park, Y., Kim, Y., Choi, S., Kim, H.: Monacobert: Monotonic attention based convbert for knowledge tracing. arXiv preprint arXiv:2208.12615 (2022)
12. Li, Z.: Deep Reinforcement Learning Approaches for Technology Enhanced Learning. Ph.D. thesis, Durham University (2023)
13. Li, Z., Jacobsen, M., Shi, L., Zhou, Y., Wang, J.: Broader and deeper: A multi-features with latent relations bert knowledge tracing model. In: European Conference on Technology Enhanced Learning. pp. 183–197. Springer (2023)
14. Li, Z., Shi, L., Cristea, A., Zhou, Y., Xiao, C., Pan, Z.: Simstu-transformer: A transformer-based approach to simulating student behaviour. In: International Conference on Artificial Intelligence in Education. pp. 348–351. Springer (2022)
15. Li, Z., Shi, L., Cristea, A.I., Zhou, Y.: A survey of collaborative reinforcement learning: interactive methods and design patterns. In: Proceedings of the 2021 ACM Designing Interactive Systems Conference. pp. 1579–1590 (2021)
16. Li, Z., Shi, L., Wang, J., Cristea, A.I., Zhou, Y.: Sim-gail: A generative adversarial imitation learning approach of student modelling for intelligent tutoring systems. Neural Computing and Applications **35**(34), 24369–24388 (2023)
17. Li, Z., Shi, L., Zhou, Y., Wang, J.: Towards student behaviour simulation: a decision transformer based approach. In: International Conference on Intelligent Tutoring Systems. pp. 553–562. Springer (2023)
18. Liu, Y., Zhou, J., Lin, W.: 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)

19. Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. arXiv preprint arXiv:1907.06837 (2019)
20. Pavlik Jr, P.I., Cen, H., Koedinger, K.R.: Performance factors analysis–a new alternative to knowledge tracing. Online Submission (2009)
21. Rasch, G.: Probabilistic models for some intelligence and attainment tests. (1993)
22. Shin, D., Shim, Y., Yu, H., Lee, S., Kim, B., Choi, Y.: Saint+: Integrating temporal features for ednet correctness prediction. In: LAK21: 11th International Learning Analytics and Knowledge Conference. pp. 490–496 (2021)
23. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: 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)
24. Sun, X., Zhao, X., Ma, Y., Yuan, X., He, F., Feng, J.: Muti-behavior features based knowledge tracking using decision tree improved dkvmn. In: Proceedings of the ACM Turing Celebration Conference-China. pp. 1–6 (2019)
25. Tiana, Z., Zhengc, G., Flanaganb, B., Mic, J., Ogatab, H.: Bekt: Deep knowledge tracing with bidirectional encoder representations from transformers. In: Proceedings of the 29th International Conference on Computers in Education (2021)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
27. Villano, M.: Probabilistic student models: Bayesian belief networks and knowledge space theory. In: International Conference on Intelligent Tutoring Systems. pp. 491–498. Springer (1992)
28. Wang, J., Ivrissimtzis, I., Li, Z., Shi, L.: Comparative efficacy of 2d and 3d virtual reality games in american sign language learning. In: The 31st IEEE Conference on Virtual Reality and 3D User Interfaces. Newcastle University (2024)
29. Wang, J., Ivrissimtzis, I., Li, Z., Shi, L.: Impact of personalised ai chat assistant on mediated human-human textual conversations: Exploring female-male differences. In: Companion Proceedings of the 29th International Conference on Intelligent User Interfaces. pp. 78–83 (2024)
30. Wang, J., Ivrissimtzis, I., Li, Z., Zhou, Y., Shi, L.: Exploring the potential of immersive virtual environments for learning american sign language. In: European Conference on Technology Enhanced Learning. pp. 459–474. Springer (2023)
31. Wang, J., Ivrissimtzis, I., Li, Z., Zhou, Y., Shi, L.: User-defined hand gesture interface to improve user experience of learning american sign language. In: International Conference on Intelligent Tutoring Systems. pp. 479–490. Springer (2023)
32. Zhang, X., Zhang, J., Lin, N., Yang, X.: Sequential self-attentive model for knowledge tracing. In: International Conference on Artificial Neural Networks. pp. 318–330. Springer (2021)