Deep-IRT with temporal convolutional network for comprehensive reflection of student ability history data

Emiko Tsutsumi[1][0000−0003−3338−8892], Tetsurou Nishio[2][1111−2222−3333−4444], and Maomi Ueno[3][2222−−3333−4444−5555]

[1] The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
tsutsumi@ai.lab.uec.ac.jp
[2] Sundai Advanced Teaching Technology, 1-7-4, Kandasurugadai, Chiyoda-ku, Tokyo, 101-0062, Japan
nishio@ai.lab.uec.ac.jp
[3] The University of Electro-Communications, 1-5-1, Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan
ueno@ai.is.uec.ac.jp

Abstract. Knowledge Tracing (KT) has been studied actively to help students learn effectively by providing optimal support based on student learning data. Important tasks of KT are tracing students' evolving abilities and predicting their performance accurately. Recently, Deep item response theory (Deep-IRT) methods combining deep learning and item response theory have been proposed to provide educational parameter interpretability and to achieve accurate performance prediction. A recent study assessed a proposed Deep-IRT with hypernetwork architecture to optimize the degree of forgetting of the past latent ability variables. However, earlier Deep-IRTs estimate a student's ability value using only a most recent latent ability parameter. Because current ability estimates cannot adequately reflect past ability history data, the parameter interpretability and the performance prediction accuracy might be impaired or biased. To overcome this difficulty, we propose a new Deep-IRT with a temporal convolutional network that convolves past multi-dimensional ability states. The proposed method stores the student's latent multi-dimensional abilities at each time point and comprehensively reflects the long-term ability history data during performance prediction. The effectiveness of the proposed method was demonstrated using experiments conducted with benchmark datasets.

Keywords: Knowledge Tracing · Deep Learning · Item Response Theory.

1 Introduction

Recently, adaptive learning has been attracting attention to provide optimal problems and learning support based on the student's ability growth in online

learning systems. In the field of artificial intelligence, Knowledge Tracing (KT) has been actively studied to provide optimal supports for students to maximize learning efficiency [6, 26, 17, 22, 24, 27, 18]. The important task is discovering concepts that the student has not mastered based on the student's prior learning history data collected by online learning systems. In addition, accurately estimating students' evolving multi-dimensional ability and predicting a student's performance (correct or incorrect responses to an unknown item) are significant for adaptive learning.

Many researchers have developed various methods to solve KT tasks. Bayesian Knowledge Tracing (BKT) [6] and Item Response Theory (IRT) [3] are the most major probabilistic approaches. BKT traces a process of student ability growth following a Hidden Markov process. It estimates whether the student has mastered the skill or not and predicts the student's responses to unknown items. On the other hand, IRT predicts a student's correct answer probability to an item based on the student's latent ability parameter and item characteristic parameters. BKT and IRT have high parameter interpretability but they can not capture the multi-dimensional ability sufficiently. Therefore, they are unable to predict the students' performances accurately when a learning task is associated with multiple skills.

To overcome the limitations, various deep-learning-based methods have been proposed [17, 27, 24, 7, 18]. Recently, Deep item response theory (Deep-IRT) methods combining deep learning and item response theory have been proposed to provide educational parameter interpretability and to achieve accurate performance prediction [24, 20, 19, 18]. Yeung (2019) [24] proposed a Deep-IRT (designated as Yeung-DI) combining a memory network architecture [27] with an IRT module. Yeung-DI adds hidden layers to a memory network architecture in order to estimate the students' ability and item difficulty parameters such as IRT. However, ability parameter of Yeung-DI is difficult to interpret because it depends on each item difficulty parameter. The most difficult challenge is to incorporate the ability and item parameters independently into a deep learning-based method so as not to degrade prediction accuracy.

Tsutsumi et al. (2021) proposed a Deep-IRT (designated as Tsutsumi-DI) that has two independent redundant networks: a student network and an item network [20]. Tsutsumi-DI learns student parameters and item parameters independently to avoid impairment of predictive accuracy. Most recently, Tsutsumi et al. (2024) combined Tsutsumi-DI with a novel hypernetwork to optimizes the degree of forgetting of the past latent variables (designated as Tsutsumi-HN). Tsutsumi-HN achieves the highest ability parameter interpretability and student's response prediction accuracies compared to existing methods. Especially, it is noteworthy that Tsutsumi-HN outperforms the attentive knowledge tracing [7] (designated as AKT) which provides state-of-the-art performance of response prediction.

Nevertheless, room for improvement remains in the prediction accuracy of the Deep-IRTs (Tsutsumi-DI and Tsutsumi-HN). They estimate a student's ability using only a most recent latent ability parameter. In general, the latest ability

depends on the past ability values while a student addresses items in the same skill. Because current ability estimates cannot adequately reflect past ability values, it interrupts the accurate estimation of the ability transition. As a result, the performance prediction accuracy might be impaired or biased.

To resolve that problem, we propose a new Deep-IRT with a Temporal Convolutional Network (TCN) [15, 2] that reflects features of the past multi-dimensional abilities to the latest ability estimate. TCN has been reported to predict time-series data more accurately than RNN-based models such as LSTM [11] and GRU [5]. TCN stores features of longer-term latent states, different from LSTM and GRU which only refer to the previous latent state. Therefore, the proposed method stores the student's latent multi-dimensional abilities at each time point and comprehensively reflects the long-term ability history data during the student's performance prediction. We conducted experiments to compare the proposed method's performance and those of earlier KT methods. The results demonstrate that the proposed method improves the performance prediction accuracy of earlier Deep-IRT methods while maintaining the high parameter interpretability. In particular, the proposed method outperforms a state-of-the-art method, Tsutsumi-HN which provides the highest performance among the current knowledge tracing methods.

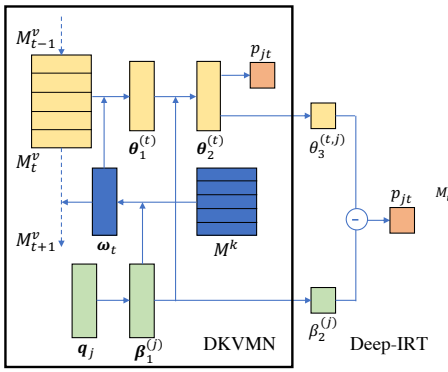

Fig. 1. The structure of Yeung-DI

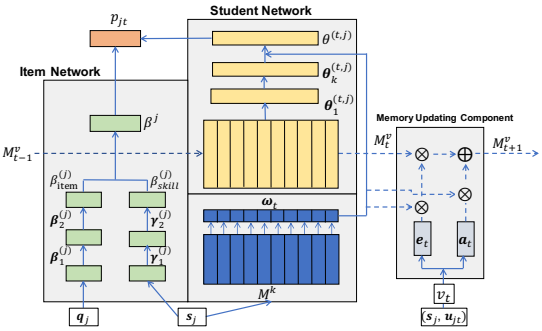

Fig. 2. The structure of Tsutsumi-DI

2 Previous Deep-IRT methods

Several Deep-IRT methods have been proposed to provide educational parameter interpretability and achieve accurate performance prediction by combining deep learning and item response theory. Yeung proposed a Deep-IRT method (Yeung-DI) combining a memory network architecture [27] with an IRT module [24]. Yeung-DI adds a hidden layer to a memory network architecture and estimates

ability and item difficulty parameters. Fig.1 presents a simple illustration. Yeung-DI predicts a student's response probability p_{jt} to an item j at time t using the student's ability $\theta_3^{(t,j)}$ and item difficulty $\beta_2^{(j)}$ such as IRT [24].

$$p_{jt} = sigmoid\left(3.0 * \theta_3^{(t,j)} - \beta_2^{(j)}\right). \tag{1}$$

However, in Yeung-DI, the ability parameter $\theta_3^{(t,j)}$ depends on each item because it is estimated using the features of the item difficulty parameter. Therefore, the ability and the item difficulty parameters cannot be interpreted separately.

To resolve the difficulty, Tsutsumi et al. propose a novel Deep-IRT method (Tsutsumi-DI) comprising two independent neural networks: the student network and the item network[20, 18], as presented in Fig.2. Tsutsumi-DI can estimate student parameters and item parameters independently such that the prediction accuracy does not decline because the two independent networks are designed to be redundant [8, 13, 14]. In addition, the item network of Tsutsumi-DI estimates two difficulty parameters of item j: the item characteristic difficulty β_{item}^{j} and the skill difficulty β_{skill}^{j}. A most recent study assessed a Deep-IRT with hypernetwork architecture (Tsutsumi-HN) to optimize the degree of forgetting of the past latent ability variables [19, 18]. Tsutsumi-HN shows the highest ability parameter interpretability and response prediction accuracies compared to existing methods. Furthermore, it can identify a relation among multi-dimensional skills and capture the multi-dimensional ability transitions. Tsutsumi-DI and Tsutsumi-HN predict a student's response probability p_{jt} to an item j at time t using the difference between a student's ability $\theta^{(t,j)}$ and the sum of two difficulty parameters β_{item}^{j} and β_{skill}^{j} as follow.

$$p_{jt} = sigmoid\left(3.0 * \theta^{(t,j)} - (\beta_{item}^{j} + \beta_{skill}^{j})\right). \tag{2}$$

3 The proposed method

3.1 Temporal Convolutional Network

Recently, convolutional neural network (CNN) [12] and Transformer [21] have attracted attention as prediction methods for time-series data. CNN is a neural network with a convolutional layer and a pooling layer. It extracts features from two-dimensional data such as images by compressing local elements into a feature using the sliding window method. In addition, temporal convolutional network (TCN) has been developed as a method to reflect past data in prediction by convolving long-term time-series data in multiple layers [15, 2]. TCN has been reported to predict time-series data more accurately than RNN-based models such as LSTM [11] and GRU [5]. TCN stores features of longer-term latent states, different from LSTM and GRU which only refer to the previous latent state. On the other hand, the Transformer stores features of longer-term data by calculating the relative distance and the weight of the relationships between

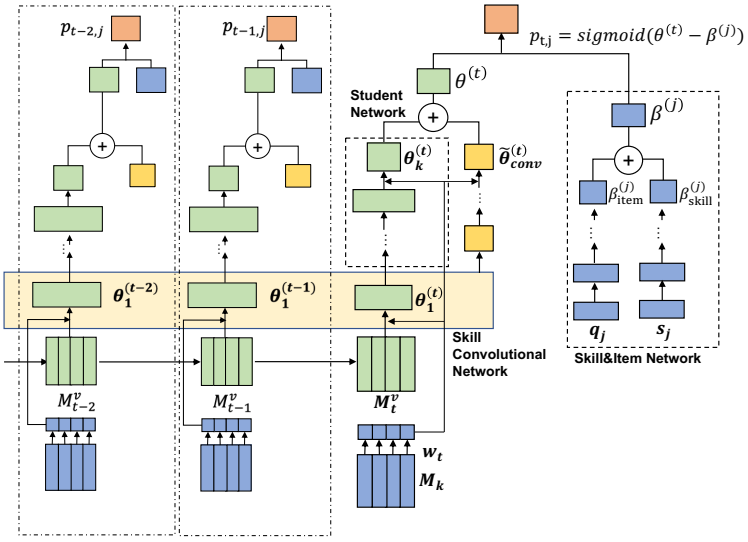

Fig. 3. The structure of the proposed method

each element of the input vector. Transformers provide highly accurate data prediction in many fields.

In a recent study comparing the performances of CNNs and Transformers, the prediction accuracies of Transformers are superior to those of CNNs when the training data size is sufficiently large [1, 4]. However, Transformer is known to often cause overfitting for small or sparse datasets. On the other hand, although CNN has slightly lower prediction accuracy than a Transformer, it can handle sparse data and train efficiently a model with less memory cost. Furthermore, for the field of KT, the earlier methods based on Transformer ([7, 16]) have shown high prediction accuracies. However, they have low parameter interpretability and then their educational applicability remains limited.

3.2 Deep-IRT with Temporal Convolutional Network

Although Tsutsumi-DI and Tsutsumi-HN provide the high parameters independently and performance prediction, they estimate the student's ability using only a most recent latent ability parameter. Therefore, current ability estimates cannot adequately reflect past ability history data. However, the latest ability depends on the past ability values while a student addresses items in the same skill. This problem might impair the performance prediction accuracy.

To resolve the problem of Tsutsumi-DI and Tsutsumi-HN, we propose a new Deep-IRT with temporal convolutional network. Fig. 3 shows the structure of the proposed method. We add TCN as the Skill Convolutional Network to Deep-IRT [18] in order to reflect features of the past multi-dimensional abilities to the

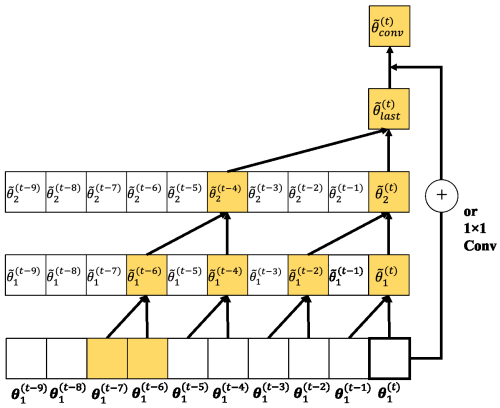

Fig. 4. Skill Convolutional Network

ability estimate. The Skill Convolutional Network stores the student's latent multi-dimensional abilities at each time point and estimates the latent ability comprehensively reflecting the long-term past ability history data. In Section 3.3, we describe the details of the Skill Convolutional Network. In addition, the proposed method has the student network and the item network. In the student network, the ability parameters θ_{it} of the student i at time t are estimated based on the latent multi-dimensional ability variable $\boldsymbol{M}_v^t$. In the item network, the model estimates the item characteristic difficulty parameter β_{item}^j and the skill difficulty β_{skill}^j. In Section 3.4 and Section 3.5, we describe the details of the student network and the item network, respectively.

3.3 Skill Convolutional Network

In Skill Convolutional Network, the proposed method estimates the optimal weight parameters related to students' past ability values by convolving the latent ability valuables using the sliding window method. The structure of Skill Convolutional Network is shown in Fig. 4. Skill Convolutional Network employs Causal Dilated Convolution [25, 15] and Residual Connection[10, 9]. Causal Dilated Convolution extracts features of long time-series data by convolving each input sequence according to "dilation" to avoid increasing the number of parameters. Dilation represents the distance between the elements of the input sequence used to compute the output value. When the dilation is 1, this convolution method is the same as the general convolution. Residual Connection adds the input value of the first layer to the last output value to avoid the vanishing gradient for the deep layers. The proposed method uses the above two methods to convolve the latent ability values in multiple layers.

The input vector $\boldsymbol{\theta}_1^{(t)}$ is encoded values of the latent variable $\boldsymbol{M}_t^v$ which represents the latent multu-dimensional ability at each time t. A student's latent

ability $\boldsymbol{\theta}_1^{(t)} \in \mathbb{R}^N$ is calculated by the following formula.

$$\boldsymbol{\theta}_1^{(t)} = \sum_{l=1}^{N} w_{tl} \left(\boldsymbol{M}_{tl}^{v}\right)^{\top}. \tag{3}$$

Where, the vector $\{\boldsymbol{\theta}_1^{(t)}, \boldsymbol{\theta}_1^{(t-1)}, \boldsymbol{\theta}_1^{(t-2)}, \cdots\}$ is N-dimensional abilities. Simplicity, we explain using one-dimensional column vector $\{\theta_1^{(t)}, \theta_1^{(t-1)}, \theta_1^{(t-2)}, \cdots\}$. In the first layer, the input vector is calculated as

$$\tilde{\theta}_1^{(t)} = \sum_{i=0}^{k-1} f_i^{(1)} \cdot \theta_1^{(t-d_1 \cdot i)}. \tag{4}$$

In the n-th layer, the latent ability values are calculated as

$$\tilde{\theta}_n^{(t)} = \sum_{i=0}^{k-1} f_i^{(n)} \cdot \tilde{\theta}_{n-1}^{(t-d_n \cdot i)}. \tag{5}$$

Where, $f_i^{(n)}$ is weight parameter and $d_n = \{1, 2, 4, \cdots, 2^n\}$ is the dilation parameter in n-th layer. k is the kernel size: $k = k_{last}$ in last layer; $k = 2$ otherwise. Finaly, the output of the Skill Convolutional Network is calculated as

$$\tilde{\theta}_{conv}^{(t)} = \tilde{\theta}_{last}^{(t)} + \theta_1^{(t)}. \tag{6}$$

$\tilde{\theta}_{conv}^{(t)}$ is a latent ability value reflecting the student's past ability history data.

3.4 Estimation of student parameters

In the student network, the student ability $\boldsymbol{\theta}_m^{(t)} = \{\theta_{m,1}^{(t)}, \theta_{m,2}^{(t)}, \cdots, \theta_{m,N}^{(t)} | 2 \leq m\}$ estimated from the latent variable $\boldsymbol{M}_v^t$ in the neural networks same as Tsutsumi et al. [20]. Then, we calculate weighted linear summations of the student ability $\boldsymbol{\theta}_m^{(t)}$ and the output of Skill Convolutional Network $\tilde{\boldsymbol{\theta}}_{conv}^{(t)} = \{\tilde{\theta}_{conv,1}^{(t)}, \tilde{\theta}_{conv,2}^{(t)}, \cdots, \tilde{\theta}_{conv,N}^{(t)}\}$ respectively.

$$\theta'^{(t)} = \sum_{l=1}^{N} \omega_{tl} \theta_{m,l}^{(t)}, \tag{7}$$

$$\tilde{\theta}^{(t)} = \sum_{l=1}^{N} \omega_{tl} \tilde{\theta}_{conv,l}^{(t)} \tag{8}$$

Where, w_{tl} is a attention weight which signifies the degree of the relation between the latent skill and the actual skill of item j.

3.5 Estimation of item and skill parameters

In the item network, the input of the item network is an embedding vector $\boldsymbol{q}_j \in \mathbb{R}^J$ calculated from the item j's tag and the student's response. Here, J represents for the number of items. We estimate the the item characteristic difficulty parameter β^j_{item} as

$$\boldsymbol{\beta}^{(j)}_1 = GELU\left(\boldsymbol{W}^{(\beta_1)}\boldsymbol{q}_j + \boldsymbol{\tau}^{(\beta_1)}\right), \tag{9}$$

$$\boldsymbol{\beta}^{(j)}_m = GELU\left(\boldsymbol{W}^{(\beta_m)}\boldsymbol{\beta}^{(j)}_{m-1} + \boldsymbol{\tau}^{(\beta_m)}\right), \tag{10}$$

$$\beta^{(j)}_{item} = \boldsymbol{W}^{(\beta_{item})}\boldsymbol{\beta}^{(j)}_m + \boldsymbol{\tau}^{(\beta_{item})}. \tag{11}$$

Next, the skill difficulty β^j_{skill} is estimated using the embedding vector $\boldsymbol{s}_j \in \mathbb{R}^S$ calculated from the skill tag of item j and the student's response. Here, S represents the number of skills.

$$\boldsymbol{\gamma}^{(j)}_1 = GELU\left(\boldsymbol{W}^{(\gamma_1)}\boldsymbol{s}_j + \boldsymbol{\tau}^{(\gamma_1)}\right), \tag{12}$$

$$\boldsymbol{\gamma}^{(j)}_m = GELU\left(\boldsymbol{W}^{(\gamma_m)}\boldsymbol{\gamma}^{(j)}_{m-1} + \boldsymbol{\tau}^{(\gamma_m)}\right), \tag{13}$$

$$\beta^{(j)}_{skill} = \boldsymbol{W}^{(\beta_{skill})}\boldsymbol{\gamma}^{(j)}_m + \boldsymbol{\gamma}^{(\beta_{skill})}. \tag{14}$$

The each output of the last layer β^j_{item} and β^j_{skill} denote the j-th item characteristic difficulty parameter and the difficulty parameter of the required skills to solve the j-th item. Then, the item i's difficulty is calculated from two difficulty parameters β^j_{item} and β^j_{skill}.

$$\beta^{(j)} = \tanh\left(\beta^{(j)}_{item} + \beta^{(j)}_{skill}\right). \tag{15}$$

The proposed method predicts a student's response probability using the student ability $\theta^{(t)}$ and item difficulty $\beta^{(j)}$ as follows.

$$p_{jt} = sigmoid(\theta^{(t)} - \beta^{(j)}). \tag{16}$$

The proposed method updates the latent variable $\boldsymbol{M}^t_v$ according to the earlier method [27]. In addition, the loss function of the proposed method employs cross-entropy, which reflects classification errors [18].

4 Experiment

5 Prediction accuracy

This section presents a comparison of the prediction accuracies for student performance of the proposed methods with those of earlier methods (Yeung-DI [24],

Table 1. Summary of Benchmark Datasets

Dataset	No. students	No. skills	No. Items	Rate Correct	Learning length
ASSISTments2009	4151	111	26684	63.6%	52.1
ASSISTments2017	1709	102	3162	39.0%	551.0
Statics2011	333	1223	N/A	79.8%	180.9
Junyi	48925	705	N/A	82.78%	345
Eedi	80000	1200	27613	64.25%	177

AKT [7], and Tsutsumi-HN [19]). We used five benchmark datasets as ASSISTments2009, ASSISTments2017, Statics2011, Junyi, Eedi. For ASSISTments2009, ASSISTments2017, and Eedi with item and skill tags, we adopt both tags as input data. Also, For Statics2011, and Junyi with only skill tags, we employ the skill as input data. Table 1 presents the number of students (No. Students), the number of skills (No. Skills), the number of items (No. Items), the rate of correct responses (Rate Correct), and the average length of the items which students addressed (Learning length).

In this experiment, we evaluate the prediction accuracies of the methods based on standard five-fold cross-validation. For each fold, 20% students are used as the test set, 20% are used as the validation set, and 60% are used as the training set according to the earlier study [7]. The optimal number of layers and k_{last} of Skill Convolutional Network are decided to maximize AUC for the validation set as shown in Table 2. For all methods, we employ the tuning parameters according to the earlier studies [7, 24, 18]. If the predicted correct answer probability for the next item is 0.5 or more, then the student's response to the next item is predicted as correct. Otherwise, the student's response is predicted as incorrect. For this study, we leverage two metrics for prediction accuracy: AUC score and Accuracy score.

Tables 3 show the results, the model with the higher performance being given in bold. Results indicate that the proposed method provides the best average AUC and Accuracy scores. The proposed method outperforms Yeung-DI, AKT, and the Tsutsumi-HN for ASSISTments2019, ASSISTments2017, Eedi and Junyi. These results show that reflecting the past ability history data by TCN is effective at improving the prediction accuracy. In addition, these datasets are large-scale datasets including more than 1000 students. The sufficient number of students for model training is one of the reasons for the improved accuracy of the proposed method. By contrast, the proposed method tends to have lower prediction accuracies for statics2011 than AKT has. For Statics2011 and Junyi, the prediction accuracy of the proposed method is comparable to those of AKT. The reason for the limited improvement was suggested that the TCN did not work effectively because the student's ability might changed independently of the past ability at each time point.

Table 2. The optimal numbers of layers and k_{last} of the proposed method

Data set	layer	k_{last}
ASSISTments2009	5	3
ASSISTments2017	8	2
Statics2011	3	7
Junyi	8	2
Eedi	8	2

Table 3. Prediction accuracies of student performance

Dataset	metrics	Yeung-DI [24]	AKT[7]	Tsutsumi-HN[19]	Proposed
ASSISTments2009	AUC	82.09+/-0.28	82.20+/-0.25	81.98+/-0.54	**82.95+/-0.30**
	Accuracy	77.41+/-0.53	77.30+/-0.55	77.15+/-0.55	**77.60+/-0.56**
ASSISTments2017	AUC	73.56+/-0.27	74.54+/-0.21	75.13+/-0.20	**75.49+/-0.36**
	Accuracy	69.78+/-0.41	69.83+/-0.06	70.69+/-0.60	**70.85+/-0.50**
Statics2011	AUC	81.15+/-0.37	**82.15+/-0.35**	81.57+/-0.50	82.02+/-0.39
	Accuracy	80.01+/-0.92	**80.41+/-0.67**	80.11+/-0.92	80.40+/-0.80
Junyi	AUC	77.92+/-0.41	78.13+/-0.39	77.91+/-0.37	**78.14+/-0.43**
	Accuracy	86.79+/-0.15	86.79+/-0.17	86.65+/-0.15	**86.85+/-0.14**
Eedi	AUC	78.93+/-0.12	77.58+/-0.21	78.97+/-0.10	**79.14 +/-0.11**
	Accuracy	73.38+/-0.17	72.35+/-0.21	73.38+/-0.13	**73.55+/-0.13**
Average	AUC	78.73	78.92	79.11	**79.60**
	Accuracy	77.47	77.54	77.60	**77.81**

6 Parameter interpretation

6.1 Estimation accuracy of ability parameters

In this section, to evaluate the interpretability of the ability parameters of the proposed method according to the earlier study [18] . We use simulation data generated from Temporal IRT [23] to compare the parameter estimates with those of the earlier Deep-IRTs [24, 18] and the proposed method. Temporal IRT is a Hidden Markov IRT which models the student ability changes following Hidden Markov process with a parameter to forget past response data. It estimates the student i's ability θ_{it} at time t, the item j's discrimination parameter a_j and the item j's difficulty parameter b_j. The prior of θ_{it} is a normal distribution described as $\theta_{i0} \sim \mathcal{N}(0, 1),\ \theta_{it} \sim \mathcal{N}(\theta_{it-1}, \epsilon)$. Therein, ϵ represents the variance of θ_{it}. It controls the smoothness of a student's ability transition. Especially, it is noteworthy that ϵ reflects the degree of the dependence of the student's current ability on the past ability values. As ϵ becomes small (large), the current ability increases the degree of the dependence (independence) on the past abilities. Therefore, as ϵ increases, the fluctuation range of the true ability increases at each time point. In addition, the priors of the item parameters are $\log a \sim \mathcal{N}(0, 1)$, $b \sim \mathcal{N}(0, 1)$.

In this experiment, each dataset includes 2000 student responses to $\{50, 100, 200, 300\}$ items. First, we estimate the item parameters $\boldsymbol{a}$ and $\boldsymbol{b}$ using 1800 students' response data. Next, given the estimated the item parameter, we estimate the students' ability parameters θ_{it} at each time using the remaining 200 students'

Table 4. Correlation coefficients of the estimated abilities

	No. items	50	100	200	300	50	100	200	300	50	100	200	300
ϵ	Method	Pearson				Spearman				Kendall			
0.1	Yeung-DI	0.63	0.67	0.74	0.74	0.63	0.66	0.75	0.75	0.44	0.47	0.55	0.55
	Tsutsumi-HN	0.73	0.77	0.85	0.82	0.74	0.78	0.88	0.87	0.54	0.59	0.70	0.69
	Proposed	**0.86**	**0.90**	**0.91**	**0.89**	**0.87**	**0.91**	**0.94**	**0.94**	**0.68**	**0.74**	**0.79**	**0.79**
0.3	Yeung-DI	0.73	0.80	0.81	0.82	0.75	0.83	0.86	0.87	0.55	0.63	0.66	0.67
	Tsutsumi-HN	0.82	0.86	0.86	0.86	0.85	0.91	0.94	0.95	0.66	0.74	0.79	**0.80**
	Proposed	**0.84**	**0.91**	**0.90**	**0.91**	**0.88**	**0.93**	**0.95**	**0.95**	**0.67**	**0.77**	**0.80**	**0.80**
0.5	Yeung-DI	0.77	0.80	0.81	0.81	0.81	0.86	0.88	0.89	0.61	0.65	0.68	0.69
	Tsutsumi-HN	**0.85**	**0.84**	**0.83**	**0.82**	**0.90**	**0.93**	**0.94**	**0.95**	**0.71**	**0.76**	0.78	**0.80**
	Proposed	0.85	**0.84**	0.82	0.81	0.89	0.92	0.92	0.89	0.71	0.73	0.73	0.70
1.0	Yeung-DI	0.79	0.81	0.82	0.81	0.83	0.88	0.89	0.89	0.63	0.68	0.70	0.69
	Tsutsumi-HN	0.82	**0.80**	**0.81**	0.79	**0.89**	**0.92**	**0.94**	0.94	**0.70**	0.75	**0.79**	**0.79**
	Proposed	0.80	0.79	0.80	0.79	0.88	**0.92**	0.92	0.93	0.68	0.74	0.75	0.76

response data. In addition, we obtain results using $\epsilon = \{0.1, 0.3, 0.5, 1.0\}$ for each dataset.

We calculate the Pearson's correlation coefficients, the Spearman's rank correlation coefficients, and the Kendall rank correlation coefficients using a student's abilities θ_t at time $t \in \{1, 2, \cdots, T\}$, as estimated using the true model and the Deep-IRT methods. Next, we average these correlation coefficients of all students. The proposed method employs a 8 layers and $k_{last} = 2$ in TCN for all datasets.

Table 4 presents the average correlation coefficients of the methods for the respective conditions. Table 4 shows that the proposed method has a higher correlation than the earlier Deep-IRTs for the small variances of the ability parameters ($\epsilon = \{0.1, 0.3\}$). A small variance ϵ indicates that the ability strongly depends on the past ability history data. Therefore, TCN works effectively and improves the estimation accuracy by reflecting past ability values. On the other hand, a large variance ($\epsilon = \{0.5, 1.0\}$) means a weak relationship between the current and past ability values and then it leads to rapid ability fluctuates at each time point. In this case, Tsutsumi-HN estimating ability by only the most recent ability and response data, provides higher correlations. The results suggest that the proposed method is superior when the current ability depends on the past abilities, and Tsutsumi-HN is superior otherwise. It is noteworthy that there is no significant difference in the correlation coefficients between the ability estimates of the proposed method and those of Tsutsumi-HN. The proposed method provides comparable high estimation accuracies to those of Tsutsumi-HN.

6.2 Student ability transitions

In this section, we visualize the ability transitions estimated by the proposed method and verify the accuracy of the ability estimation. Visualization of ability

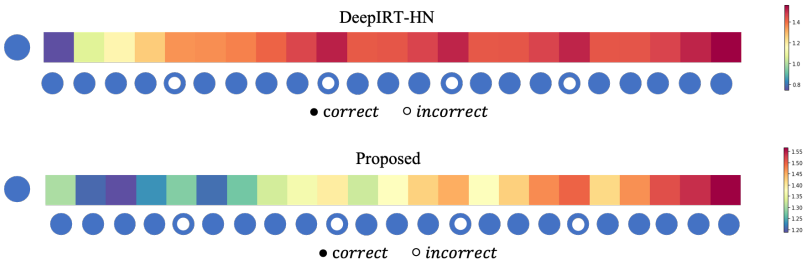

Fig. 5. The example of student's one-dimensional ability transitions

transition helps both students and teachers to identify students' strengths and weaknesses. We use ASSISTments2009 dataset according to the earlier studies [24, 19].

First, Fig.5 depicts an example of student's one-dimensional ability transitions estimated by Tsutsumi-HN and the proposed method. The vertical axis shows the student's ability value on the right side and the horizontal axis shows the item number. The student response is shown by filled circles "●" when the student answers the item correctly; it is shown by hollow circles "○" otherwise. DeepIRT-HN shows that the ability barely changes after it increases rapidly in items 1–5. Because DeepIRT-HN estimates an ability using only a most recent ability parameter, it might cause overfitting when a student continuously answers correctly (incorrectly) to items. As a result, the estimated ability converges to an extremely high (low) value. By contrast, the proposed method shows that the ability gradually increases as the student answers items correctly by estimating an ability reflecting past ability history data.

Second, Fig. 6 depicts an example of student multi-dimensional ability transitions of each skill estimated by the proposed method. We use the first 30 responses and the student attempted four skills: "equation solving more than two steps" (shown in grey), "equation solving two or few steps" (shown in green), "ordering factions" (shown in orange), and "finding percents" (shown in yellow). The proposed method estimates abilities considering relations among the skills. Therefore, when a student answers an item correctly (or incorrectly), the abilities of the other skills change with the ability of the corresponding skill. In addition, as shown in Fig.5, each ability gradually fluctuates reflecting the student's responses while a student is addressing items in the same skill. However, when the student answers an item in a different skill, the estimated ability value fluctuates significantly. Actually, it is unlikely that the ability of a particular skill changes rapidly. In future work, we improve the estimation accuracy and interpretability of multi-dimensional ability.

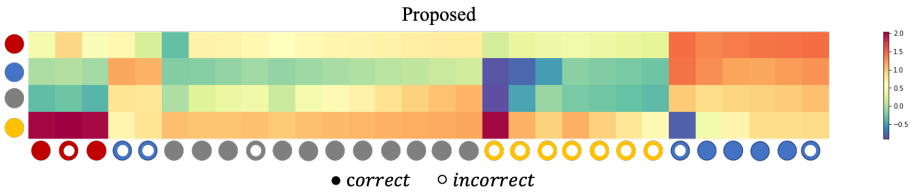

Fig. 6. The example of a student's multi-dimensional ability transition

7 Conclusion

This article proposed a new Deep-IRT with a temporal convolutional network for knowledge tracing. The proposed method stores the student's latent multi-dimensional abilities at each time point and estimates the latent ability reflecting the long-term ability history data comprehensively. To demonstrate the performance of the proposed method, we have conducted experiments using benchmark datasets and simulation data. To summarize the results, the proposed method improves the performance prediction accuracy of earlier Deep-IRT methods while maintaining the high parameter interpretability. Especially, when the ability fluctuates depending on the past abilities, the proposed method works effectively and outperforms the performance of the earlier methods. As future work, we will improve the estimation accuracy and interpretability of multi-dimensional ability.

References

1. Arkin, E., Yadikar, N., Xu, X., Aysa, A., Ubul, K.: A survey: object detection methods from cnn to transformer. Multimedia Tools and Applications **82** (10 2022). https://doi.org/10.1007/s11042-022-13801-3
2. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling (2018). https://doi.org/10.48550/ARXIV.1803.01271
3. Baker, F., Kim, S.: Item Response Theory: Parameter Estimation Techniques, Second Edition. Statistics: A Series of Textbooks and Monographs, Taylor & Francis (2004)
4. Chen, Z., Ma, M., Li, T., Wang, H., Li, C.: Long sequence time-series forecasting with deep learning: A survey. Information Fusion **97**, 101819 (2023). https://doi.org/https://doi.org/10.1016/j.inffus.2023.101819
5. Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR **abs/1412.3555** (2014), http://arxiv.org/abs/1412.3555

6. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User Model. User-Adapt. Interact. **4**(4), 253–278 (Dec 1995)
7. 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 (2020)
8. He, H., Huang, G., Yuan, Y.: Asymmetric valleys: Beyond sharp and flat local minima. In: Advances in Neural Information Processing Systems 32. pp. 2553–2564. Curran Associates, Inc. (2019)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. IEEE Computer Society, Los Alamitos, CA, USA (jun 2016). https://doi.org/10.1109/CVPR.2016.90
10. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. pp. 630–645. Springer International Publishing, Cham (2016)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
12. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
13. Morcos, A., Yu, H., Paganini, M., Tian, Y.: One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. In: Advances in Neural Information Processing Systems 32. pp. 4932–4942. Curran Associates, Inc. (2019)
14. Nagarajan, V., Kolter, J.Z.: Uniform convergence may be unable to explain generalization in deep learning. In: Advances in Neural Information Processing Systems 32. pp. 11615–11626. Curran Associates, Inc. (2019), http://papers.nips.cc/paper/9336-uniform-convergence-may-be-unable-to-explain-generalization-in-deep-learning.pdf
15. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. In: Arxiv (2016), https://arxiv.org/abs/1609.03499
16. Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. In: Proceedings of International Conference on Education Data Mining (2019)
17. Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., Sohl-Dickstein, J.: Deep knowledge tracing. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28. pp. 505–513. Curran Associates, Inc. (2015)
18. Tsutsumi, E., Guo, Y., Kinoshita, R., Ueno, M.: Deep knowledge tracing incorporating a hypernetwork with independent student and item networks. IEEE Transactions on Learning Technologies **17**, 951–965 (2024). https://doi.org/10.1109/TLT.2023.3346671
19. Tsutsumi, E., Guo, Y., Ueno, M.: Deepirt with a hypernetwork to optimize the degree of forgetting of past data. In: Proceedings of the 15th International Conference on Educational Data Mining (EDM) (2022)
20. Tsutsumi, E., Kinoshita, R., Ueno, M.: Deep-irt with independent student and item networks. In: Proceedings of the 14th International Conference on Educational Data Mining (EDM) (2021)
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In Advances in Neural Information Processing Systems pp. 5998–6008 (2017)

22. Weng, R., Coad, D.: Real-time bayesian parameter estimation for item response models. Bayesian Analysis **13** (12 2016). https://doi.org/10.1214/16-BA1043
23. Wilson, K.H., Karklin, Y., Han, B., Ekanadham, C.: Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. In: 9th International Conference on Educational Data Mining. vol. 1, pp. 539–544 (06 2016)
24. Yeung, C.: Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. In: Proceedings of the 12th International Conference on Educational Data Mining, EDM (2019)
25. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations (2016)
26. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: Individualized bayesian knowledge tracing models. In: Artificial Intelligence in Education. pp. 171–180. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
27. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory network for knowledge tracing. In: Proceedings of the 26th International Conference on World Wide Web. pp. 765–774. WWW '17, International World Wide Web Conferences Steering Committee (2017)